



An Evolutionary-Based Approach for Low-Complexity Intrusion Detection in Wireless Sensor Networks

Ting Zhang¹ · Dezhi Han¹ · Mario D. Marino² · Lin Wang³ · Kuan-Ching Li⁴ 

Accepted: 1 August 2021 / Published online: 8 August 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

As security threats are increasingly diversified, a critical problem in Wireless Sensor Network environments (WSNs) is detecting anomalies. WSNs are affected by several limitations, such as limited energy, insufficient memory, weak computation power, and short communication range. Hence, it is necessary to improve the detection accuracy and convergence speed of intrusion detection in such environments. In this article, we propose an intrusion detection model based on Time-Varying Parameter Improved Particle Swarm Optimization (TVP-IPSO) with Principal Component Analysis (PCA) and Support Vector Machine (SVM). The PCA is applied aimed at the data dimension reduction by compressing the data to reduce energy consumption, and an intrusion detection algorithm based on SVM is considered to ensure high detection accuracy. To optimize the SVM algorithm and identify its optimal parameters, the TVP-IPSO is used to improve the intrusion detection algorithm's detection precision and convergence speed. Experimental results show that the detection accuracy of TVP-IPSO-SVM is higher than GA-SVM and IPSO-SVM, demonstrating that the proposed research has better adaptability, higher detection accuracy, and faster convergence speed when compared to other works presented.

Keywords Low-complexity intrusion detection · Particle swarm optimization · Principal component analysis · Support vector machines · Time-varying parameter · WSNs

1 Introduction

Wireless Sensor Networks (WSNs) are the bridge between the physical world and information technology. WSNs enable us to study physical world environmental phenomena via an abundant number of sensors, which collect information converted into digital format to be processed and transmitted via network and subsequently stored and analyzed into fog nodes.

Due to the rapid development and maturity of wireless communication facilities, sensor technology, embedded application, and microelectronic technology, WSNs integrated into these technologies gain popularity and have significantly impacted various fields of

✉ Kuan-Ching Li
kuancli@pu.edu.tw

Extended author information available on the last page of the article

modern society. Furthermore, owing to their characteristics (i.e., small nodes, low cost, and low energy consumption [1]), WSNs are widespread in many contexts, such as environmental pollution monitoring, smart grid, biomedical health management, and behavioral habit detection.

WSN is a kind of network with limited energy consumption and weak storage capacity, in which nodes are often placed in "no man's land". Considering factors such as the "openness" of the network, the fixed routing mechanism of the data, and the limitation of resources in terms of time and space, some critical and sensitive data could be subject to several attacks, resulting in serious security problems [2–4, 28].

Besides, as WSNs are closely related to scientific experimentations, human production activities, and daily activities, WSN has become an essential asset to consider when dealing with the entire system's security from the perspective of network security technology and system applications. Hence, the problem of network security should also keep pace with research.

A WSN is established based on the public Internet and wireless networks. When establishing these traditional computer networks, only consider connectivity' convenience, but ignore the security threats. Like the protocol itself, the device itself all has some loopholes [5, 27].

For this reason, several techniques have been developed, such as firewall technology, anti-virus technology, encryption technology, authentication technology, and other security methods [6–9, 21, 22, 26, 32]. Nevertheless, many of these technologies are based on passive defense methods, and most of them can only detect security problems that have occurred. The difference from the abovementioned technology is Intrusion detection technology, which has a specific active defense function [10].

By focusing on the security problems that affect WSNs, we propose a WSN intrusion detection model, referred as TVP-IPSO-SVM: Time-Varying Parameter Improved Particle Swarm Optimization with Principal Component Analysis (PCA) and Support Vector Machine (SVM). TVP-IPSO-SVM combines *Principal Component Analysis (PCA)* with *Time-Varying Parameter Improved Particle Swarm Optimization (TVP-IPSO)* and *Support Vector Machine (SVM)*, that analyzes and compares sensor nodes' data and determines whether there is an intrusion, according to the intrusion behavior characteristics. First, it maps the high-dimensional features in the input space to the new low-dimensional feature space through PCA, reducing the data dimension to optimize the amount of transmitted data. Next, it uses the SVM classifier to evaluate the attack. Aimed at reducing the training time and improve the classification performance of SVM, an improved PSO with Time-Varying parameters is introduced by combining the advantages of *Local Particle Swarm Optimization (LPSO)* and *Global Particle Swarm Optimization (GPSO)* to optimize the parameters of SVM.

In TVP-IPSO, the particles learn mainly from the global and local optimal particles and the optimal particles in the population. It has a robust and comprehensive optimization ability that overcomes the drawbacks of the PSO algorithm, such as the rapid fall into local extremum points.

On top of that, the proposed model reduces the amount of data transmission between WSN nodes while reducing energy consumption and improving the detection rate. Overall, the main contributions include:

1. Application of the PCA to reduce the size of the intrusion detection data, exploiting the performance of the SVM classifier in processing imbalanced datasets,

2. The use of time-varying inertial weights and learning factors introduce an improvement to the standard particle swarm optimization algorithm. The TVP-IPSO can search for the optimal value faster and avoid the search falling into an optimal local state,
3. Discovery of the SVM parameters through the proposed TVP-IPSO to provide a helpful intrusion detection model,
4. Design of a layered network structure as the WSNs model. The TVP-IPSO-SVM detection mechanism is loaded onto all nodes of WSNs that significantly saves the energy consumption caused by data transmission,
5. The performance of the proposed model is evaluated through different experimentations utilizing the KDD Cup99 public dataset and compared the feasibility and effectiveness of the model through two indicators: detection rate and false alarm rate concerning standard SVM, GA-SVM [11], and IPSO-SVM [13].

The remainder of this article is structured as follows. First, Sect. 2 provides some necessary preliminaries and related work, while in Sect. 3, the PCA, SVM, and the proposed model for intrusion detection in WSNs are presented, besides explaining the logical functioning of such a model. Next, the main aspects of the proposed TVP-IPSO-SVM algorithm are introduced in Sect. 4, the performance of the proposed scheme is analyzed, and evaluations of the experimental results achieved are discussed in Sect. 5. Finally, the concluding remarks and future research directions are given in Sect. 6.

2 Background and Related Work

Nowadays, the aims of WSN security research are twofold. The former is the network attack defense mechanism, and the latter is the network intrusion detection mechanism.

A Network attack defense mechanism is the first line of defense of network security that mainly adopts critical management, secure routing, and authentication to ensure network security. However, with the development of attack technology, it is easier for attackers to break through the first line of defense to cause attacks and damage to the network. Therefore, it is necessary to have a corresponding mechanism to detect the occurrence of attack behavior. As the second line of defense for network security, intrusion detection can detect abnormal data after the intrusion from the standard dataset and locate the intruder [29].

With the growth and extensive application of Machine Learning (ML), an increasing number of researchers have applied ML in intrusion detection systems, achieving excellent results. Several research works [2–5, 30, 31, 33] have successfully chosen to adopt a modified neural network algorithm to implement an intrusion detection system. However, the neural network algorithm calculates the input error function repeatedly and transports the error from one layer to the previous layer. Therefore, in the process of behavior establishment, each link is weighted to learn, resulting in the slow convergence speed of the algorithm. We stress that the energy consumption of WSNs with a large number of nodes is undoubtedly fatal.

SVM is a machine learning technique based on the statistical learning theory proposed by Vapnik [6], which can be helpful in the intrusion detection process when there is insufficient prior knowledge. From the literature, network intrusion data have some characteristics that SVM can exploit, such as small sample size, linear inseparability, and high dimension. Mukkamala et al. implemented an SVM technology for intrusion detection and carried out experimental verification using the neural network method [7]. The achieved

result showed that SVM is better than Decision Tree (DT), K-Nearest Neighbors (KNN), Neural Network (NN), and other algorithms.

From other related works [6, 7, 23, 24] dealing with SVM-based intrusion detection, one major issue is that the dimension of the input space is a critical factor that degrades its performance.

WSNs are applied in complex and harsh environments nowadays, so most of the sensor nodes are randomly arranged to employ aircraft dissemination, making it possible for different sensor nodes in the network to collect the same data information. If all the data collected by sensor nodes are transmitted, the life cycle of the network is severely affected. Furthermore, the incorrect data in the network also affect the administrator's decision. In particular, since most of the energy in WSNs is consumed for the process of data communication rather than node computing, "data disaster" will most likely lead to a large amount of transmission energy consumption.

Furthermore, when dealing with large-scale node deployment and real-time data perception in WSNs, the increase of data complexity affects the accuracy of the intrusion detection process. Indeed, network connection data have a high characteristic dimension; data features have significant redundancy and uncertainty that significantly affect the validity of intrusion detection results [8]. Thus, the problem of intrusion detection in WSNs is still a deeply-felt research area since it deals with a massive amount of network traffic data, a high-dimensional training dataset that continuously changes the environment, and real-time detection needs.

Sun et al. proposed the PCA method for feature dimensionality reduction to reduce data storage and calculate "data disaster" [8]. At the same time, Jing et al. addressed the problem of unbalanced training samples [9]. Ma et al. proposed the *PCA-ELM (Extreme Learning Machine, ELM)* method [10], in which they used the PCA to reduce the dimension of data and the ELM algorithm to create the classifier. This method improves the learning ability of small samples, reduces the learning time, and reduces the false detection rate and missed detection rate. Another hybrid method to improve the detection accuracy is proposed by Kuang et al., who combined multi-layer SVM with *Kernel Principal Component Analysis (KPCA)* and *Genetic Algorithm (GA)* for intrusion detection [11], by using KPCA to decrease the dimensionality of the feature set and reduce the training time. However, the selection of SVM parameters, including penalty factor C , the nuclear function type, and the nuclear function parameters, significantly influences classification accuracy [12].

At present, researchers have proposed different algorithms for optimizing parameters in the context of Swarm Intelligence (SI) [2, 4, 11, 12], which has had a significant impact on the field of computational science over the past decade, inspired by the natural evolution of organisms and is especially useful for solving optimization problems. For example, Kalaivani et al. proposed an effective classifier based on Artificial Bee Colony (ABC) for intrusion detection [4]. Neha et al. introduced Salp Swarm Optimization (SSO) to optimize the hyperparameters of classifiers [2]. Likewise, some researchers have applied the *Genetic Algorithm (GA)* [11] and standard *Particle Swarm Optimization (PSO)* [12] to optimize SVM parameters.

Among other research directions, the PSO algorithm has been widely adopted in objective function optimization and neural network, as it is one of the most popular and practical techniques for optimizing SVM parameters. Nevertheless, the PSO algorithm [12] shows disadvantages, such as easy traps in local optimization, low accuracy of optimization calculation, and slow convergence in the later period. Due to these shortcomings, to solve the parameters setting of SVM and provide the most appropriate feature subset, researchers have devoted much work to particle swarm optimization methods. Related, Aburomman

et al. proposed an efficient way to optimize SVM parameters by using PSO to improve the accuracy of intrusion detection [12]. Also, S. Liu et al. proposed an intrusion detection model based on SVM and *IPSO* (*Improved Particle Swarm Optimization, IPSO*) [13], which has higher detection accuracy and faster convergence speed. Furthermore, H. Liu et al. put forward a WSNs intrusion detection algorithm based on *CS-CPSO* (*Complete Sine-Mapping Chaotic Particle Swarm Optimization, CS-CPSO*) and SVM [14] to improve the detection accuracy and convergence speed of intrusion detection algorithms in WSNs. However, the methods proposed by Aburomman et al. [12] and S. Liu et al. [13] mainly focus on the phase mixing of various algorithms, and the convergence speed is slow in the later stages of evolution. H. Liu et al. [14] introduced the chaos model to realize the PSO algorithm simply, but its local search ability is weak, making it easier to fall into the local optimum. The proposed TVP-IPSO algorithm combines the advantages of the LPSO and GPSO algorithms to improve the ability to jump out of the optimal local solution, effectively avoid computational blindness, and improve the convergence speed of the algorithm.

This investigation proposes a new intrusion detection model based on WSNs, which mainly focuses on two aspects. First, the PCA is adopted to reduce the characteristic dimension of data. Next, the TVP-IPSO algorithm is proposed to achieve the optimal SVM kernel parameters to establish the intrusion detection model of WSNs. This proposed model reduces the computation time and improves the accuracy of intrusion detection. The following section introduces the proposed TVP-IPSO algorithm.

3 The Proposed Intrusion Detection Model

The proposed intrusion detection model for WSNs is based on PCA, SVM, and TVP-IPSO algorithms. PCA is a dimension reduction technique that is the most widely used in data analysis and compression [11]. The basic idea is to construct a series of linear combinations of primitive variables to form several comprehensive geometric indicators, to remove the correlation of data, and make low-dimensional data maintain the variance information of original high-dimensional data to the greatest extent [15].

The learning ability and generalization ability of SVM depend on the choice of its parameters. In this paper, the *Radial Basis Function (RBF)* is adopted, so there are two parameters to be optimized, namely, the kernel function parameter σ^2 and the penalty parameter C , where the size of C is related to the tolerable error. At the same time, the proportion of σ^2 is associated with the blank input range of learning samples.

This section mainly introduces the framework and implementation mechanism of the proposed model, including the underlying algorithms of PCA and SVM in detail. The TVP-IPSO algorithm for optimizing SVM parameters is described in the next section.

3.1 WSNs Intrusion Detection Model Framework

Intrusion detection is essentially a classification problem, where the challenge is to distinguish between abnormal data and normal data. The intrusion data has high-dimensional features and contains many noise attributes. Therefore, PCA is adopted to reduce data dimension, and the SVM classifier is applied to intrusion detection.

Two stages characterize this model. In the first stage, a series of irrelevant and redundant data is deleted using PCA. In contrast, in the second stage, the data are classified according to the training dataset and the test dataset. Next, we use the TVP-IPSO algorithm to

select the optimal parameters of SVM by particle iteration and position update. Then SVM is applied with the optimal parameters to model the intrusion detection training dataset and the established WSNs intrusion detection model to detect the test dataset. Lastly, the acquisition of intrusion detection ends. The framework of the proposed intrusion detection model is depicted in Fig. 1.

3.2 Intrusion Detection Implementation Mechanism

Based on the analysis of the WSN network structure, in this paper, we have chosen to utilize a layered network structure for the representation of the proposed WSN network model, as shown in Fig. 2, which structure consists of three main parts: the common node, the sink node, and the cluster-head node [14]. The cluster-head node handles the information collected by different member nodes in the cluster and then transfers the processed data to the sink node, thus reducing the data sent to the sink node and significantly saving the energy consumption caused by data transmission.

Before performing the WSNs intrusion detection, the TVP-IPSO-SVM algorithm needs to learn from offline experimental data to create the detection mechanism. After that, the TVP-IPSO-SVM detection mechanism can be placed in each node of the WSNs. Moreover, in the network intrusion detection phase, sink nodes, club-head nodes, and common nodes cooperate for detection. First, the sink node sends broadcast messages to the cluster-head node and activates the TVP-IPSO-SVM misuse detection mechanism in the cluster head node when it is threatened. Then, the detection mechanism for the

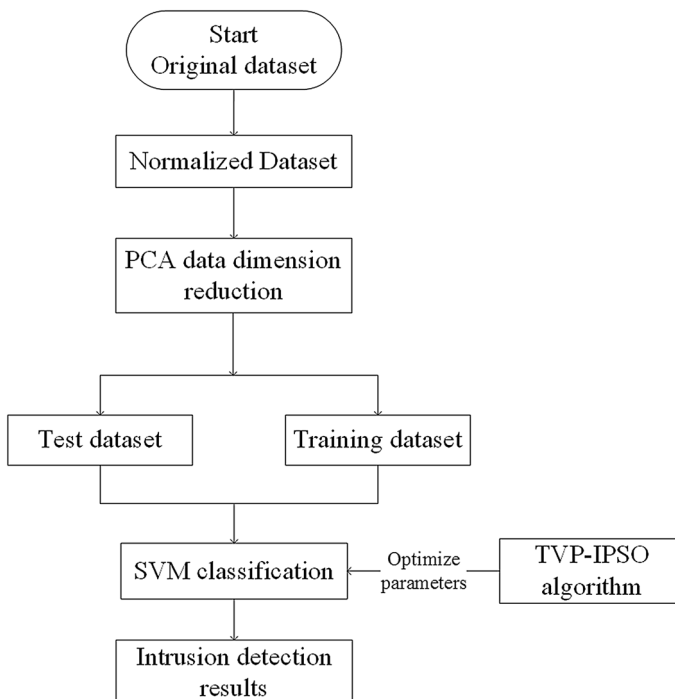
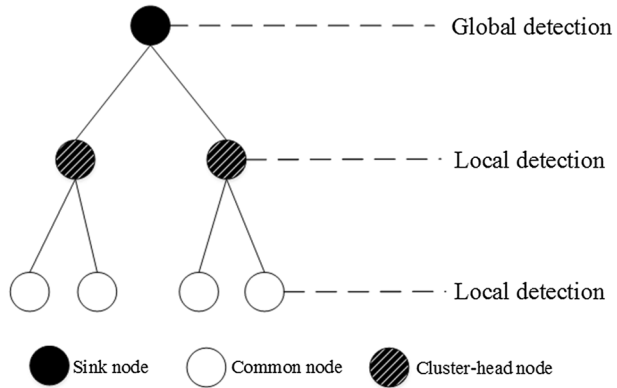


Fig. 1 The proposed intrusion detection model framework

Fig. 2 The layered intrusion detection structure for the WSN network model



cluster-head node is activated. Once such a mechanism detects an anomaly, the cluster-head node sends the data of the exception node to the sink node.

On the other hand, if it cannot detect the anomaly, the cluster-head node sends broadcast messages to the common node in the cluster. Then, it activates the TVP-IPSO-SVM anomaly detection mechanism of the common node. Finally, the common node performs the detection of anomalous data. More precisely, if an exception is detected, the common node sends the anomalous data to the sink node through the cluster-head node. The sink node makes the final judgment according to the data information it has mastered.

3.3 Principal Component Analysis

The core concept of PCA is to find the standardized linear combination of some original variables with the most considerable variance and transform a large number of related variables into fewer unrelated variables. The linear combination of the most considerable variance in the original variables is the first principal component of the transformation.

The second principal component is a linear combination of the first main component and the original variable orthogonal to the second largest variance, and so on. In many datasets, the principal components that contribute the most variance to the original dataset are initially the main components. Therefore, the remaining data ignored as the minimum loss of dimension reduction variance. The following transformation process is given as follows. Let X_1, X_2, \dots, X_n be the training samples, in which a set of m -dimensional vectors represent each value. Therefore, we use a matrix to represent the dataset:

$$X_{m \times n} = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix} = [x_1, x_2, \dots, x_n]. \tag{1}$$

The average observation value μ is defined as:

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i. \tag{2}$$

The deviation ϕ_i can be obtained from the average value:

$$\phi_i = x_i - \mu. \quad (3)$$

The covariance matrix of the sample in the datasets can be constructed by:

$$H = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T = \frac{1}{n} \sum_{i=1}^n \phi_i \cdot \phi_i^T. \quad (4)$$

When PCA is applied, *Singular Value Decomposition (SVD)* [15] usually calculates the eigenvector and eigenvalue corresponding to the sample covariance matrix H .

Let $(\lambda_1, \mu_1), (\lambda_2, \mu_2), \dots, (\lambda_m, \mu_m)$ be the m pairs of eigenvalues and eigenvectors of the covariance matrix H . We select the k eigenvectors with the highest eigenvalues. The formula to determine the dimension of subspace k is:

$$\frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^m \lambda_i} \geq \beta, \quad (5)$$

where β is the ratio of the variables of the subspace to all the variables of the original space. Next, matrix A with size $m \times k$ is generated, in which columns contain k eigenvectors. According to (6), the data represented by principal component data are projected into the k -dimensional subspace.

$$y_i = A^T(x_i - \mu) = A^T \phi_i. \quad (6)$$

Principal component analysis has the characteristics of easy calculation and robust explanation. PCA algorithm measures information by the value of the variance of data. The higher the variation, the more data it contains; otherwise, the less information it contains. Therefore, PCA is transforming the coordinate projection of high-dimensional data to the direction with the most significant variance of data and forming a new coordinate system to represent the coordinate transformation of data.

3.4 Support Vector Machine

SVM can solve the problem of constructing high dimensional data model effectively. Its advantages include strong generalization ability, high fitting accuracy, and an insensitive dimension. The working principle of SVM is as follows. Firstly, it transforms the input variables of the lower dimensional space into a higher dimensional characteristic space by the appropriate nonlinear transformation. Then it constructs an optimal classification hyperplane, which separates the two types of input data correctly as much as possible. Therefore, the following constrained binary classification problem can be outlined.

Assuming that the training sample set is $\{x_i, y_i\}, x_i \in R_n$, where x_i is the i th training sample in the input space, and y_i represents the category to the example belonging to a group, the search of the optimal classification hyperplane under the condition of linear inseparability is equivalent to solve the optimization problem in (7), where (8) satisfies the requirements of (7):

$$\min \frac{1}{2} w \cdot w + C \sum_{i=1}^n \delta_i \quad (7)$$

$$s.t. y_i(w \cdot x_i + b) \geq 1 - \delta_i, \delta_i \geq 0, \quad i = 1, 2, \dots, n \tag{8}$$

where w is a vector and b is the threshold, C is the penalty parameter used to control the penalty degree of right and wrong sample distribution, δ_i is the relaxation variable used to measure the distance between delta y_i and the hyperplane.

By constructing the Lagrangian function, the dual problem can be given as below:

$$m \in \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j (\varphi(x_i) \cdot \varphi(x_j)) - \sum_{i=1}^n \alpha_i \tag{9}$$

$$s.t. \sum_{i=1}^n \alpha_i y_i = 0, C \geq \alpha_i \geq 0, \quad i = 1, 2, \dots, n. \tag{10}$$

where α_i represents the i th training example of Lagrange multiplier coefficient regression, obtained by solving the double optimization problem in support vector learning.

If the sample set is of nonlinear relation, $\varphi(x)$ is firstly used to map the original sample set to the higher dimensional feature space for linear classification through nonlinear mapping. Nonlinear mapping using the kernel function is as follows $K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$. At the optimal solution of (9), the form of the optimization function is as follows:

$$m \in \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^n \alpha_i. \tag{11}$$

SVM seeks the maximum geometric space of the hyperplane in the kernel space and realizes the nonlinear separation of feature space. Based on the relationship between the size of the training dataset and feature space, the use of RBF produces better results when the dimension of feature space is less than the training dataset. Therefore, we select RBF as the kernel function of the SVM, as follows:

$$K(x_i, x_j) = \exp\left\{ \frac{-x_i - x_j}{\sigma^2} \right\}, \tag{12}$$

where σ^2 is the radial parameter of basis kernel width. Finally, the decision function is:

$$f(x) = \text{sgn}\left(\sum_{i=1}^n \alpha_i y_i K(x_i, x) + b \right), \tag{13}$$

where the threshold value is $b = y_j - \sum_{i=1}^n \alpha_i y_i K(x_i, x_j)$

In essence, intrusion detection aims to distinguish between normal activities and attack activities through the detector. Therefore, intrusion detection is a typical classification problem. As one of the most representative classification algorithms, the SVM algorithm has the following characteristics:

1. SVM uses a structural risk minimization principle to solve classification problems. Such problems can be high dimensions, small samples, and nonlinearity. Meanwhile, this classification algorithm can reinforce the generalization ability of the learning machine. After obtaining minor errors from training samples, it can still ensure small errors for test sets,

2. The analysis of the SVM classification function shows that the kernel function parameter σ and penalty parameter C affect the classification performance of SVM. Therefore, it is imperative to study the selection of penalty parameters and kernel function parameters. Furthermore, an improved particle swarm optimization (TVP-IPSO) algorithm optimizes the SVM to find the suitable values of C and σ .

3.5 TVP-IPSO-SVM Algorithm

For the intrusion detection of WSNs with high-dimensional data samples, SVM has excellent advantages. However, many experiments and analyses on the basic concepts of SVM in Sect. 3 show that the performance of intrusion detection models based on SVM is directly related to its parameters, i.e., kernel function parameter σ and penalty factor C . Therefore, if the parameters are not selected properly, the accuracy of intrusion detection will be lower.

To find the appropriate values of σ and C , the optimization of SVM parameters by PSO can optimize the classification performance of SVM to some extent. However, the standard PSO algorithm has precocious convergence during the optimization process, and the local searchability is weak. Although it converges quickly, it has insufficient iterative precision and slow evolution speed in the later phases. Nevertheless, the particle swarm shows convergence, which is easy to sink into the optimal local solution [12].

To solve the above problem, we propose the TVP-IPSO algorithm, which introduces Time-Varying parameters based on the basic PSO algorithm. In TVP-IPSO, particles learn from global and local optimal particles and learn from the optimal particles in the population, so they have strong global optimization ability.

The TVP-IPSO-SVM is described in detail next.

3.6 PSO Algorithm

A particle swarm optimization algorithm consists of a random search algorithm developed by simulating the foraging behavior of birds, which is based on group cooperation [12]. The algorithm starts with a set of random particles (stray solutions) and then iteratively finds the optimal solution. PSO algorithm uses the speed-position search model, and each particle corresponds to an inserted solution in the solution space. The fitness function determines the fitness of the solution, defined according to the optimization objective. The specific description is as follows [30].

Firstly, we initialize a group of random particles with the size of n , so population $Z = (Z_1, Z_2, \dots, Z_n)$. Here, the position of the i th random particle in the d -dimensional solution space is expressed as $Z_i = (z_{i1}, z_{i2}, \dots, z_{iD})$, and the velocity is expressed as $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$. Then the optimal solution is generated through iterative search. At each iteration, the random particle updates its position and velocity by dynamically tracking two extremes.

The first extremum is the optimal solution (i.e., the optimal position of an individual random particle) generated by random particle search from the initial iteration to the current iteration, which is called *individual extremum* $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$. The next extreme value is the optimal solution that is the optimal global location of all particles found by the entire particle population, which is called the *absolute global value* $P_g = (p_{g1}, p_{g2}, \dots, p_{gD})$.

Before finding these two optimal values, particle i updates its position and velocity in d -dimension according to (14) and (15) in the l th iteration. Then, the particles continue to iterate to the set maximum number of iterations, or the fitness function value of the optimal position of the current group is equal to the preset minimum value:

$$v_{id}^{l+1} = w * v_{id}^l + c_1 * rd_1^l (p_{id}^l - z_{id}^l) + rd_2^l (p_{gd}^l - z_{gd}^l) \quad (14)$$

$$z_{id}^{l+1} = z_{id}^l + v_{id}^{l+1}, \quad (15)$$

where w is the inertia weight, which affects the overall optimization ability; c_1 and c_2 are acceleration constants (learning factors); rd_1^l and rd_2^l are random numbers on the interval $[0, 1]$. Again, l represents the current iterations times; z_{id}^l is the position of particle i in the d -dimensional space of the l th repetition; $v_{id} \in [v_{min}, v_{max}]$ is the particle velocity that determines the update direction and size of the next generation.

The PSO algorithm is a random, parallel optimization algorithm, which advantages are that it does not need the objective function to have a specific property, and it can make the objective function converge to a final point in a short time. Besides, the PSO algorithm has low process complexity and is easy to implement [16]. However, the PSO algorithm also has the following shortcomings:

1. By analyzing the evolutionary (14) and (15), the particle's speed is almost close to 0, because it always chases the current global best and best search. This problem causes particles to fall into local minima and unable to escape quickly.
2. To expand the search scope, we need to increase the number of particles in the particle swarm or reduce the chasing of particles to the global search of the whole particle swarm. Unfortunately, those operations lead the algorithm only to allow for global particle search, but whether the final result is an optimal global value cannot be guaranteed. This limitation reduces the global optimal search capability of the particles, and the algorithm is not easy to converge [17, 33].

3.7 TVP-IPSO Algorithm

An exciting feature of evolutionary algorithms is that they have strong global searchability in the early stages of algorithm iteration and a local search ability with higher accuracy in the later stages [18]. In other words, the improvement of the optimization algorithm depends on the optimization ability and optimization speed.

Nevertheless, it relies on the cooperation and competition between groups for the standard PSO, and the particles themselves have no mutation mechanism. Therefore, once a local extreme value constrains a single particle, it is challenging to jump out of the local extreme value. The particle needs to use other particles to jump out of the local value [19].

The inertia weight is a critical factor in balancing algorithm categories and local searchability. Learning factors can also effectively solve the coordination problem between their own experience and social experience. These two parameters control the importance of group learning and particle self-learning. Therefore, necessary improvements are made to learning factors c_1 , c_2 , and the inertia weight w of standard PSO to improve the algorithm performance in the iterative process and ensure that it can find the optimal global solution.

Two versions of particle swarm optimization that are based on different topological structures are available, namely, local particle swarm optimization and global particle

swarm optimization [12]. LPSO has a slow convergence rate, but it is easy to out of the local optimal. On the other hand, GPSO converges quickly but sometimes falls into local optimum. In this paper, an improved algorithm for particle swarm optimization with time-varying parameters (TVP-IPSO) is proposed based on the advantages of both LPSO and GPSO.

The use of acceleration factor and time-varying inertia weight leads us to the dynamic balance algorithm’s global and local search capabilities. In the early stage, particles are encouraged to search extensively in the search space, avoiding premature convergence of the algorithm due to aggregation in the optimal local value. In the later stage, particles are encouraged to move to the optimal amount to accelerate the convergence.

In this article, we propose a time-varying parameter particle swarm algorithm to dynamically adjust the learning factors c_1 and c_2 , and the inertia weight w . The algorithm adjusts the performance by the cooperation of learning factors and inertia weight. The recursive formula of the TVP-IPSO algorithm is as follows:

$$v_{id}^{l+1} = w^l * v_{id}^l + c_1^l * rd_1^l \left[(1 - \alpha^k)(p_{id}^l - z_{id}^l) + \alpha^l (p_{1,d}^l - z_{id}^l) \right] + c_2 * rd_2^l (p_{gd}^l - z_{gd}^l), \tag{16}$$

$$z_{id}^{l+1} = z_{id}^l + v_{id}^{l+1}, \tag{17}$$

$$w^l = w_i + \frac{l(w_f - w_i)}{l_{max}}, \tag{18}$$

$$c_1^l = c_{1i} + \frac{l(c_{1f} - c_{1i})}{l_{max}}, \tag{19}$$

$$c_2^l = c_{2i} + \frac{l(c_{2f} - c_{2i})}{l_{max}}, \tag{20}$$

where α is a constant between [0, 1]; w is the inertia weight; c_1 and c_2 are acceleration constants; rd_1^l and rd_2^l are random numbers on the interval [0, 1]; v_{id}^{l+1} is the velocity of the $l+1$ th iteration of particle i in the d dimension; $p_{1,d}^l$ is the position of the optimal particle in the population of the l generation; l_{max} is the total number of algorithm iterations; $w_i, w_f, c_{1i}, c_{1f}, c_{2i}, c_{2f}$ are constants.

According to (18), increasing the total number of iterations l_{max} , the inertia weight w decreases non-linearly. The inertia weight w weighs the local search and global search of particle swarm, and its value directly affects the optimization ability of the particle swarm. When the weight becomes large, the global optimization ability of the particle swarm also increases.

In contrast, the weak local optimization capabilities help prevent the population from entering the optimal local solution. Therefore, this paper introduces dynamic inertial weight, which makes the PSO algorithm find a balance between global and local, significantly improving the algorithm’s performance and promoting the extensive application of PSO.

Wang et al. show that the learning factors can significantly impact the algorithm’s performance over time [18]. According to (19) and (20), and through the study on improving learning factors c_1 and c_2 , the global and local search ability of the algorithm’s dynamic

balance is benefited. In the early stage, particles are encouraged to search extensively in the search space, avoiding premature convergence of the algorithm due to aggregation in the optimal local value. Though, particles are encouraged to move to the optimal amount to accelerate the convergence in the later stage.

For obtaining better SVM parameters, time-varying parameters are introduced based on the standard PSO algorithm. As a result, TVP-IPSO can have excellent global searchability and ensure local search accuracy, further enhancing the convergence speed of the algorithm.

3.8 Fitness Function

Four performance indicators (or metrics) of intrusion detection systems are widely known and applied: true positive (*TP*), true negative (*TN*), false positive (*FP*), and false-negative (*FN*). The *TP* denotes the correct prediction of normal behavior, *FP* indicates that abnormal data was not detected correctly, *FN* means that normal behavior is considered abnormal. Finally, *TN* denotes the correct detection of anomalies [11].

The critical point is to use TVP-IPSO to find the optimal parameters of the SVM and improve intrusion detection capability. Therefore, the intrusion detection accuracy used on the SVM training set as the fitness function is defined as:

$$fitness(i) = \frac{TP + TN}{TP + FP + TN + FN}. \quad (21)$$

3.9 Process of TVP-IPSO-SVM

Given that the energy available for nodes is limited in WSNs, TVP-IPSO is adopted to optimize SVM parameters and reduce both training time and energy consumption. Firstly, we use kernel parameter σ and the penalty factor C of SVM to set the position Z of particle swarm, that is, $Z = (C, \sigma)$. The location of the optimal solution is the optimal value of the two parameters of the SVM.

Following the abovementioned analysis, Fig. 3 shows the primary process of optimizing SVM parameters by TVP-IPSO in the proposed model.

Step 1 Initialize population according to kernel parameter σ and the penalty factor C of the SVM. The particle position is $z_{id} = (C_{id}, \sigma_{id})$;

Step 2 Initialize TVP-IPSO algorithm parameters, including maximum iteration number k_{max} , speed v_{id} , then set learning factors c_1 and c_2 , and select random numbers rd_1^l and rd_2^l on $[0,1]$;

Step 3 Set the extreme individual value of the particle P_i as the current position, and take the global absolute value P_g as the best particle position in the initial population;

Step 4 Calculate the fitness value of particle $fitness(i)$ according to (21);

Step 5 Adjust, according to $fitness(i)$, the historical and global optimal values of the particle, referred to as P_i and P_g , respectively, to find the extreme global value and extreme point of the particle, then initialize the extreme individual value and extreme individual point;

Step 6 Update the parameters according to (18)-(20) and the position and speed of particles according to (16)-(17). Recalculate the corresponding $fitness(i)$ and modify the iteration coefficient as $l = l + 1$;

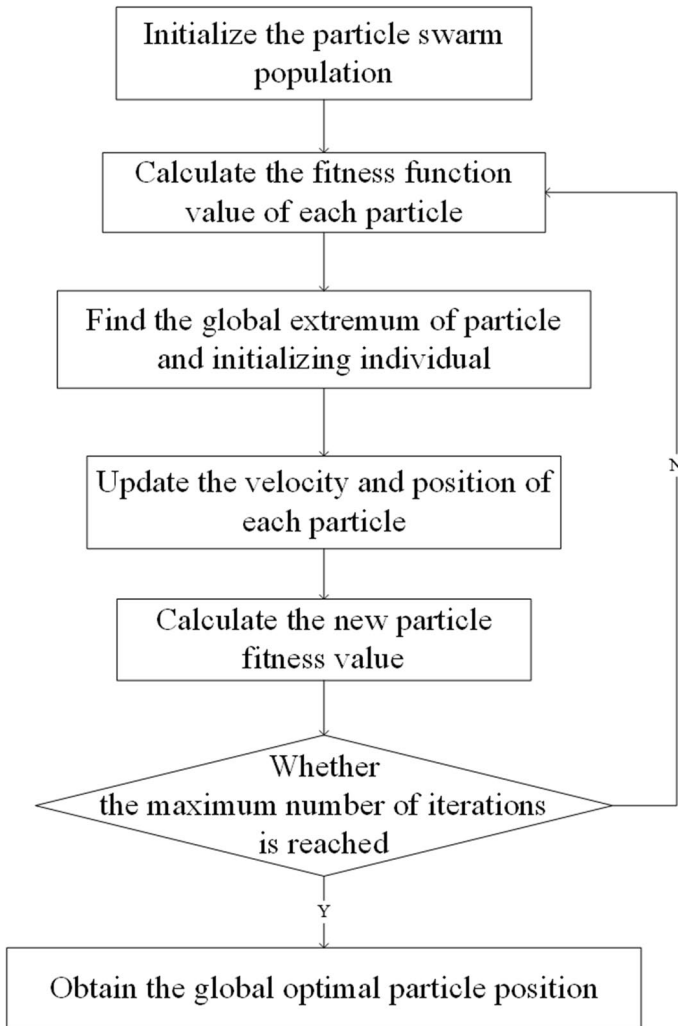


Fig. 3 The process of optimizing the parameters of SVM by TVP-IPSO

Step 7 Determine whether l_{max} has reached the maximum number of iterations. If l_{max} is reached, end the process; otherwise, return to Step 4;

Step 8 Decode the optimal Z , then obtain the optimal parameter combination (C, σ) .

3.10 Complexity Analysis

This part analyzes the complexity of the algorithm proposed in this paper. From the above description of the traditional PSO algorithm and TVP-IPSO algorithm, it can be observed that the difference in the complexity of the two algorithms is mainly based on the number of particles and the running time required to replace the particles in each generation.

More precisely, for the traditional PSO, the number of particles in each iteration is the same. Assume the number of particles in the i th iteration is N_i , where $i = 1, 2, \dots, m$. Here, m represents the maximum number of generations, so $N_1 = N_2 = \dots = N_m = N$. The operation time required for each iteration of each particle is T_T , so we can conclude that the total running time needed for the traditional PSO algorithm for optimization is $N \times m \times T_T$.

On the other hand, for what concerns TVP-IPSO, the number of particles gradually decreases with each iteration, then $N_1 \geq N_2 \dots \geq N_m$. Again, assuming that the calculation time required for each iteration of each particle is T_D , the total running time needed for the TVP-IPSO algorithm after optimization is $\sum_i^m N_i \times T_D$, which shows to be better than the traditional PSO algorithm.

Moreover, the space complexity of the proposed algorithm is mainly related to the SVM in this paper. In detail, the SVM runtime space complexity is $O(k \times d)$, where k is the number of support vectors, and d is the dimension of data. It is important to note that this is because, before the execution of the proposed TVP-IPSO-SVM algorithm, we use PCA for data dimensionality reduction, which significantly reduced the space complexity of the algorithm.

4 Results and Evaluation

This section first compares the proposed scheme with some of the most representative works proposed in state-of-the-art. Following next, taking the detection rate and false detection rate as the key points, we emphasize the main advantages of the model proposed in this paper applied to WSNs.

4.1 Methodology

In this experimentation, the following goals are aimed:

1. Determine the influence of parameters obtained by different algorithms on SVM classification performance;
2. Verify the superiority of TVP-IPSO-SVM in terms of intrusion detection;
3. Verify the advantage of TVP-IPSO-SVM in WSNs compared with other algorithms.

In the experimental evaluation of the proposed TVP-IPSO-SVM scheme, we compared it with related works and evaluated it on multiple datasets. Finally, the analysis of the experimental results proves that the proposed scheme is better than other schemes in terms of safety and energy consumption. In particular, especially in detecting DoS attack data, its classification accuracy reaches more than 98%. At the same time, in terms of verifying the convergence speed and training time, our proposal has better feasibility in WSNs intrusion detection due to its faster speed.

4.2 Algorithm Comparison for Selecting the Classifier Parameters

In this part, we further detail the methodology we have previously described. In this methodology, we have compared some schemes mentioned in related work with our scheme in the stage of selecting the classifier parameters. For example, reference [2, 4, 5, 13] and [11]

respectively used *Salp Swarm Optimization (SSO)*, *Artificial Bee Colony (ABC)*, *Particle Swarm Optimization (PSO)*, and *Genetic Algorithm (GA)* to optimize classifier parameters and create a new intrusion detection model. More precisely, we have compared the following indicators for the above algorithms:

1. The relation between solution time and dataset size: of course, the solution time of an algorithm increases as the dataset size increases. As regards the algorithms taken into consideration, the relationship defined above can be *Linear* or *Exponential*.
2. Algorithm convergence: any algorithm can only iterate a finite number of times. Algorithm convergence means that a stable solution can be obtained after a limited number of iterations. This index reflects the performance of the algorithm to a certain extent. This indicator can take *Good* and *Poor* values.
3. Search space range: In some cases, the optimization algorithm still has poor results. Whether the optimization problem can get good results, the range of search space is also an important indicator. The larger the search space, the higher the probability of obtaining a high-quality solution. For the algorithms described above, this indicator can take the *Small*, *Medium*, and *Wide* magnitudes.
4. The capacity of algorithm memory: In the algorithms taken into consideration, particles have a "memory capacity", so that the next generation solution can inherit more information from the previous generation and thus find the optimal solution in a shorter time. This indicator can take the *Ordinary*, *Strong*, and *None* values.
5. High-dimensional data processing capacity: Due to the large amount of high-dimensional data in WSNs, we compare the algorithm's ability to process high-dimensional data. For the algorithms described above, this indicator can take the *Strong*, *Weak*, and *General* magnitudes.

We use the above five indicators to summarize the performance of each algorithm. The GA does not follow the optimal solution in the update position operation of the optimization process. The previous knowledge will be destroyed during the population iteration process, the reason why the GA has no memory and poor convergence. Therefore, we use the GA as the benchmark for this comparison. The performance of the above algorithms is shown in Table 1.

Clearly, in the case of a large number of datasets, the GA algorithm requires more time, and the accuracy is the worst. Although the ABC algorithm has a good algorithm convergence, the optimization effect on high-dimensional problems is still not noticeable. The most promising algorithms mentioned above are SSO and PSO.

Table 1 Algorithm performance comparison

Schemes	Relationship between solution time and dataset size	Algorithm convergence	Search space	Memory capacity	Capacity of processing high-dimensional data
[2] SSO	Linear	Good	Small	Strong	General
[4] ABC	Linear	Good	Medium	Ordinary	Weak
[5, 13] PSO	Linear	Good	Wide	Strong	Strong
[11] GA	Exponential	Poor	Small	None	Weak
TVP-IPSO	Linear	Good	Wide	Strong	Strong

Table 2 The computational complexity

Algorithm	The computational complexity
SSO	$O(t(d * n + Cof * n))$
PSO	$O(tdn)$

Table 3 Dataset category distribution and description

Type	Normal	Abnormal (8000)		
		DoS	Probe	R2L
Number	4000	4000	2000	2000

However, the range of SSO particle search space is not extensive, and the particle needs to optimize the objective function of multiple local extreme values in the optimization process. Thus, there is a possibility of evolutionary stagnation. As a result, the SSO algorithm brings higher energy consumption than PSO under the exact condition of WSNs.

Importantly, it should be noted that our energy assumptions and comparisons are indirectly measured via time: Energy = Power * time, where Energy is the energy spent by the algorithm, Power is the total system (processor, memory and other components) power, and time is the time to execute the particular algorithm. The higher complexity the algorithm presents, the higher time it requires to complete. However, we have not included a power meter to measure the power used by the algorithms in the experimentation since it does change the natural environment where intrusion detection is aimed at.

Table 2 shows the computational complexity of the SSO and PSO, where *t* represents the number of iterations, *d* is the number of variables (dimensions), *n* represents the number of solutions, and *Cof* is the cost of the objective function.

As shown in Table 2, the computational complexity of SSO is slightly higher than that of PSO. Finally, by analyzing the information given above, we choose to optimize the basic PSO algorithm to process the classifier’s parameters. This action is desirable. In the last section of this paper, we have proved theoretically that the computational complexity of our proposed TVP-IPSO performs better than the traditional PSO algorithm.

4.3 Experimental Description

To assess the performance of the proposed model, we selected the KDD Cup99 dataset from the MIT Lincoln laboratory for simulation. In detail, such a dataset is mainly composed of both abnormal and normal data. Again, the abnormal data can be further divided into the following four categories: unauthorized access from a remote machine (Remote to Local, *R2L*), denial of service (*DoS*), *Probe*, and unauthorized access to local supervisor privileges (User to Root, *U2R*) [11]. However, due to the small size of U2R, for the experimental phase in this paper, we only focus on data concerning *DoS*, *Probe*, and *R2L* categories.

Considering that SVM is a valid statistical method for small samples, we use a 50% sample as a training dataset and the rest as a testing dataset. Before performing the experiments, we changed the symbol field of the data points to numeric values and converted them into the normalized format. Table 3 shows the dataset category distribution.

The server used for the experimentations is configured with one Intel(R) Core i5-6500 @ 3.20 GHz CPU and 8G memory, running Microsoft Windows10 operating system and MATLAB R2018a environment. We use the detection rate and false alarm rate [25] to measure the most effective intrusion detection model.

$$\text{Detection rate: } DR = \frac{TP}{TP + FN}.$$

$$\text{False alarm rate: } FAR = \frac{FP}{TN + FP}.$$

There are some constraints in these experiments. In the TVP-IPSO algorithm, the population size is 30, the evolution times are 50, and the acceleration constants c_1 and c_2 are equal to 1.5 and 1.6. We used the Libsvm in MATLAB as a supplementary tool. The time of cross-validation is set to be $\nu=4$. The penalty factor range is between $[2^{-5}, 2^5]$, and RBF is used as the kernel function.

4.4 Experimental Results and Discussion

To evaluate the effectiveness of the TVP-IPSO+SVM in terms of intrusion detection performance, we compared it with SVM that does not use any Intelligent Evolutionary Algorithms, GA+SVM [11], and IPSO+SVM [13]. First of all, from Table 4, it can be observed that for different optimization algorithms, we obtain different best penalty factor C and kernel parameter σ .

In detail, from Table 4, we can observe that the proposed TVP-IPSO+SVM algorithm performs better than other detection algorithms in terms of detection rate, accuracy, and false alarm rate. More precisely, when TVP-IPSO is used for classification training of SVM, the maximum classification accuracy is 97.3%. Again, the detection rate of the four algorithms is relatively high because the SVM algorithm has good generalization ability while avoiding the minimum local convergence. However, the detection rate of our proposal is better than the other three methods, reaching 98.2%. Furthermore, to reduce the false alarm rate.

According to the experimental results shown in Table 4, different types of attacks are not considered separately. Besides, for further analyzing the detection performance of TVP-IPSO+SVM against unknown attacks, we carried out the following experiments:

Table 4 Comparison of different algorithms

Method	Parameters		Train accuracy (%)	Test accuracy (%)	DR (%)	FAR (%)
	C	σ				
SVM	2	32	97.8667 (5872/6000)	96.9333 (5816/6000)	97.1	5.4
[11] GA+SVM	74.71	86.28	97.9167 (5875/6000)	96.9667 (5818/6000)	97.9	5.3
[13] IPSO+SVM	1.255	50	97.8667 (5872/6000)	96.95 (5817/6000)	98.08	4.3
TVP-IPSO+SVM	1.511	49.11	98.35 (5901/6000)	97.3 (5838/6000)	98.2	4.3

Table 5 Comparison of the detection rates of various categories

Category method	DR (%)			FAR (%)		
	Probe	DoS	R2L	Probe	DoS	R2L
SVM	96	97	96.9	2.1	1.15	1.05
[11] GA+SVM	96	97.25	96.8	1.95	1.10	1.05
[13] IPSO+SVM	96	97.25	97.2	1.95	1.10	1.05
TVP-IPSO+SVM	96	98.05	97.5	1.75	1.05	1.0

Table 6 The average training time of three different algorithms

Algorithm	Training time (s)
[11] GA+SVM	864.343
[13] IPSO+SVM	841.734
TVP-IPSO+SVM	811.137

1. Mix different types of attacks and regenerate a new test dataset.
2. Compare the detection rate and false alarm rate of various attacks with different methods.

The experimental results of the simulation experiments are given in Table 5.

As shown in Table 5, the TVP-IPSO+SVM is better than IPSO+SVM, GA+SVM, and SVM in terms of accuracy, DR, and FAR, for different types of attacks. In this research, the algorithm is adapted to integrate the local search ability and global search ability of the TVP-IPSO algorithm and the SVM detection ability, so that the proposed method can identify intruders faster and improve the detection rate. Moreover, in terms of FAR, taking Probe as an example, the FAR of the TVP-IPSO+SVM algorithm is 0.2% and 0.35% lower than the IPSO+SVM algorithm [13] and traditional SVM, respectively. Again, since a Time-Varying Parameter method is adopted in the weight updating of the PSO algorithm, the global search efficiency and local search accuracy are improved, and the FAR is reduced.

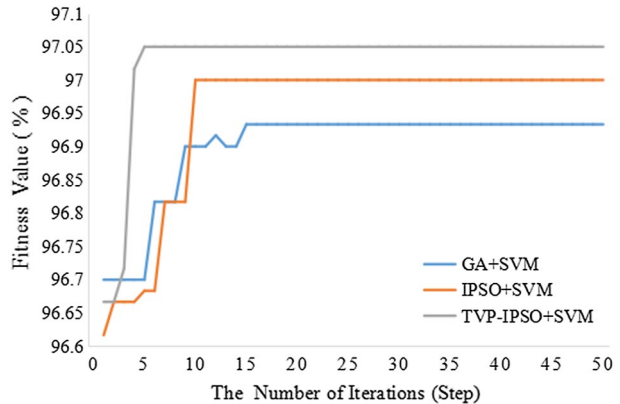
Under the same training samples and iteration conditions, we can evaluate the algorithm’s computational complexity by comparing the training time of the proposed algorithm, GA+SVM, and IPSO+SVM [13]. The average training time of those three algorithms, obtained from the simulation, is shown in Table 6.

In Table 6, we see that the training time of the TVP-IPSO+SVM algorithm is 53.206 s shorter than GA+SVM, and 30.597 s less than IPSO+SVM. Therefore, in terms of training time, our proposal results to be better than the other two. We remark that this is of tremendous importance in the context of WSNs, where node energy consumption is a critical factor to consider.

In terms of algorithm performance improvement, we compared the convergence and fitness changes of TVP-IPSO+SVM, IPSO+SVM, and GA+SVM during the iterative process. The optimization curve is shown in Fig. 4.

Figure 4 further shows the advantages of the TVP-IPSO+SVM algorithm over other algorithms. For example, we can observe that TVP+IPSO has a convergent step of 6 for SVM parameter optimization, while IPSO and GA have concurrent levels of 12 and 17, respectively. Furthermore, among them, the fitness value obtained by TVP-IPSO+SVM

Fig. 4 Curves of convergence and fitness changes of the three algorithms in the iterative process



optimization reaches 97.05%, while that of IPSO and GA is 97% and 96.93%, respectively. Therefore, the TVP-IPSO model proposed is superior to IPSO and GA models in both convergence speed and detection accuracy.

5 Conclusion and Future Work

Since PSO tends to fall into the local extremum, in this paper, we propose a method to optimize PSO by using Time-Varying inertia weight and acceleration factor, thereby improving the global and local searchability of the algorithm. In the TVP-IPSO algorithm, particles learn from global and local optimal particles and the optimal particles in the population. Therefore, particles have a robust global optimization ability, enabling them to optimize SVM classifier parameters better.

This research combines the TVP-IPSO algorithm and SVM to carry out the intrusion detection model for WSNs. Such a model improves the detection accuracy of known attacks and increases the detection rate of unknown attacks. PCA is used for data dimension reduction processing, while RBF is adopted as an SVM kernel function. Furthermore, appropriate parameters are selected for SVM to avoid SVM classifiers with excessive or low fitting due to the improper determination of parameters and to reduce training time and improve the classification performance of SVM.

Experimental results show that compared with the IPSO-SVM in [13] and traditional GA-SVM, the proposed TVP-IPSO-SVM improves the detection rate, accelerates the convergence rate while reducing the false alarm rate and detection time.

As future work, we intend to focus on the nodes of WSNs, to quickly find and process an attacked node when an attack is detected, improving the ability of sensor nodes to deal with problems independently. When there are errors or attacks, ensure the strength of the essential services and the sustainable application of the network.

Acknowledgements This work is supported by the National Natural Science Foundation of China (Nos. 61672338 and 61873160).

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal

relationships that could have appeared to influence the work reported in this paper.

References

1. Sudip, M., Krishna, P. V., & Abraham, K. I. (2011). A simple learning automata-based solution for intrusion detection in wireless sensor networks. *Wireless Communications & Mobile Computing*, *11*(3), 426–441.
2. Neha, N., Raman, M. R. G., Somu, N., Senthilnathan, R., & Shankar, V. S. (2020). An improved feed-forward neural network using Salp swarm optimization technique for the design of intrusion detection system for computer network. *Advances in Intelligent Systems and Computing*, *999*, 867–875.
3. Benmessahel, I., Xie, K., Chellal, M., & Semong, T. (2019). A new evolutionary neural networks based on intrusion detection systems using locust swarm optimization. *Evolutionary Intelligence*, *12*, 131–146.
4. Kalaivani, S., Vikram, A., & Gopinath, G. (2019). An effective swarm optimization based intrusion detection classifier system for cloud computing. In *2019 5th international conference on advanced computing & communication systems (ICACCS), Coimbatore, India, 2019* (pp. 185–188).
5. Kala, T. S., & Christy, A. (2019). An intrusion detection system using opposition based particle swarm optimization algorithm and PNN. In *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon), Faridabad, India, 2019* (pp. 184–188).
6. Vapnik, V., & Cortes, C. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273.
7. Mukkamala, S., Janoski, G., & Sung, A. (2002). Intrusion detection using neural networks and support vector machines. In *Proceedings of the 2002 international joint conference on neural networks (IJCNN), Honolulu, HI, USA, 2002* (Vol. 2, pp. 1702–1707).
8. Sun, Z. W., Liang, G. W., Bai, Y., & Ji, Z. C. (2013). A hierarchical intrusion detection model in wireless sensor networks. *Information and Control*, *42*(6), 670–676.
9. Jing, X., Wang, H. X., Nie, K., & Luo, Z. W. (2012). Feature selection algorithm based on IMGA and MKSVM to intrusion detection. *Computer Science*, *07*, 102–105.
10. Ma, S. H., & Hu, B. (2015). Network intrusion detection based on features selecting and samples selecting. *Computer Systems & Applications*, *24*(9), 426–441.
11. Kuang, F., Xu, W., & Zhang, S. (2014). A novel hybrid KPCA and SVM with GA model for intrusion detection. *Applied Soft Computing*, *18*, 178–184.
12. Aburomman, A. A., & Reaz, M. B. I. (2016). A novel SVM-KNN-PSO ensemble method for intrusion detection system. *Applied Soft Computing*, *38*, 360–372.
13. Liu, S., Wang, L., Qin, J., & Guo, Y. (2018). Zuo H (2018) An intrusion detection model based on IPSO-SVM algorithm in wireless sensor network. *Journal of Internet Technology*, *19*(7), 2124–2133.
14. Liu, H., Li, L., & Hu, J. (2017). WSNs intrusion detection algorithm based on CS-CPSO and SVM fusion. *Transducer and Microsystem Technologies*, *36*(9), 110–112.
15. Bahsi, H., Nömm, S., & La Torre, F. B. (2018). Dimensionality reduction for machine learning based IoT botnet detection. In *Proceedings of the 15th international conference on control, automation, robotics and vision, (ICARCV), Singapore, 2018* (pp. 1857–1862).
16. Cláudio, P., Filho, A. P., & Schirru, R. (2017). A novel method to improve dose assessment due to severe NPP accidents based on field measurements and particle swarm optimization. *Annals of Nuclear Energy*, *110*, 148–159.
17. Hu, Z., Zou, D., Kong, Z., & Shen, X. (2018). A particle swarm optimization algorithm with time varying parameters. In *2018 Chinese control and decision conference (CCDC), Shenyang, Jun. 2018* (pp. 4555–4561).
18. Wang, D. F., & Meng, L. (2016). Performance analysis and parameter selection of PSO algorithm. *Journal of Automation*, *42*(10), 1552–1561.
19. Li, C. W., Yang, L., Fournier, P. V., Wu, M. T., Hong, T. P., Wang, S. L., & Zhan, J. (2016). Mining high-utility itemsets based on particle swarm optimization. *Engineering Applications of Artificial Intelligence*, *55*, 320–330.
20. Eduardo, D. L. H., Emiro, D. L. H., Ortiz, A., Ortega, J., & Prieto, B. (2015). PCA filtering and probabilistic SOM for network intrusion detection. *Neuro computing*, *164*, 71–81.
21. Bi, K., Han, D., et al. (2020). K maximum probability attack paths generation algorithm for target nodes in networked systems. *International Journal of Information Security*. <https://doi.org/10.1007/s10207-020-00517-4>

22. Han, D., Pan, N., & Li, K. (2020). A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection. *IEEE Transactions on Dependable and Secure Computing*. <https://doi.org/10.1109/TDSC.2020.2977646>
23. Zhang, W., Han, D., Li, K. C., & Massetto, F. I. (2020). Wireless sensor network intrusion detection system based on MK-ELM. *Soft Computing*, 24, 12361–12374.
24. Xu, J., Han, D., Li, K. C., & Jiang, H. (2020). A K-means algorithm based on characteristics of density applied to network intrusion detection. *Computer Science and Information Systems*, 17(2), 665–687.
25. Nandhakumar, S., & Malmurugan, N. (2016). ETIDS: An effective trust based intrusion detection system for wireless sensor networks. *Journal of Computational & Theoretical Nanoscience*, 13(3), 1791–1797.
26. He, J., Han, D., & Li, K.-C. (2020). On one-time cookies protocol based on one-time password. *Soft Computing*, 24(8), 5657–5670.
27. Fan, Y., Zhao, G., Li, K. C., Zhang, B., Tan, G., Sun, X., & Xia, F. (2020). SNPL: One scheme of securing nodes in IoT perception layer. *Sensors*, 20(4), 1090.
28. Xiao, T., Han, D., He, J., Li, K., & de Mello, R. F. (2021). Multi-Keyword ranked search based on mapping set matching in cloud ciphertext storage system. *Connection Science*, 33(1), 95–112.
29. Taheri, R., Shojafar, M., Alazab, M., et al. (2020). FED-IIoT: A robust federated malware detection architecture in industrial IoT. *IEEE Transactions on Industrial Informatics*, 2020, 1–11.
30. Shojafar, M., Taheri, R., Pooranian, Z., Javidan, R., Miri, A., & Jararweh, Y. (2019) Automatic clustering of attacks in intrusion detection systems. In *The 16th ACS/IEEE international conference on computer systems and applications*. IEEE.
31. Liang, W., Xiao, L., Zhang, K., Tang, M., He, D., & Li, K. C. (2021). Data fusion approach for collaborative anomaly intrusion detection in Blockchain-based systems. *IEEE Internet of Things Journal*. <https://doi.org/10.1109/JIOT.2021.3053842>
32. Liang, W., Huang, W., Long, J., et al. (2020). Deep reinforcement learning for resource protection and real-time detection in IoT environment. *IEEE Internet of Things Journal*, 7(7), 6392–6401. <https://doi.org/10.1109/JIOT.2020.2974281>
33. Xu, J., Han, D., et al. (2020). A K-means algorithm based on characteristics of density applied to network intrusion detection. *Computer Science and Information Systems*, 17(2), 665–687.
34. Du, Z., Han, D., et al. (2019). Improving the performance of feature selection and data clustering with novel global search and elite-guided artificial bee colony algorithm. *The Journal of Supercomputing*, 75(8), 5189–5226.
35. Liang, W., Xiao, L., Zhang, K., et al. (2021). Data fusion approach for collaborative anomaly intrusion detection in Blockchain-based systems. *IEEE Internet of Things Journal*. <https://doi.org/10.1109/JIOT.2021.3053842>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Ting Zhang is currently pursuing the M.S. degree with Shanghai Maritime University. Her research interests include wireless communication and security.



Dezhi Han received the Ph.D. degree from the Huazhong University of Science and Technology. He is currently a Professor of computer science and engineering with Shanghai Maritime University. His research interests include visual question answering, network security, cloud computing, mobile networking, wireless communication, and cloud security.



Mario Donato Marino is currently a Senior Lecturer at Leeds Beckett University. He received his B.E. in Electrical Engineering with emphasis in Computer Engineering, while MSc. and Ph.D. in Computer Engineering all at the University of Sao Paulo. He has received a best paper award on an international top conference and has co-authored several international articles in journals, conferences, and workshops which include computer architecture, microprocessor evaluation, systems, high-performance computing, distributed computing, parallel computing, and performance evaluation. He serves/has served a number of committees in international conferences and workshops and has been a reviewer in top journals in the area of computer architecture and high performance systems. Mario is also currently an Assistant Editor at the IEEE Access Journal as well as at the International Journal of Embedded Systems (IJES, Inderscience). He is a member of the Institute of Electrical and Electronics Engineers (IEEE) and a member of the Association of Computer Machinery (ACM).



Lin Wang is currently an Associate Professor at the School of Physics and Optoelectronic Engineering at Xidian University. From 2015 to 2016, she was a visiting scholar with Remote Sensing Signal and Image Processing Laboratory (RSSIPL), University of Maryland, Baltimore County, USA. Her research interests include hyperspectral image processing, automatic target recognition, and real-time image processing.



Kuan-Ching Li is currently a Distinguished Professor at Providence University, Taiwan. He is a recipient of awards and funding support from several agencies and industrial companies, as he also received distinguished chair professorships from universities in China and other countries. He has been actively involved in many major conferences and workshops in program/general/steering conference chairman positions and a member of the program committee. He has organized numerous conferences related to high-performance computing and computational science and engineering. Besides publication in refereed journals and top conference papers, he is co-author/co-editor of several technical professional books published by CRC Press/Taylor and Francis, Springer, and McGraw-Hill. Dr. Li's research interests include GPU/manycore computing, Big Data, and cloud. He is a senior member of the IEEE and a fellow of the IET.

Authors and Affiliations

Ting Zhang¹ · Dezhi Han¹ · Mario D. Marino² · Lin Wang³ · Kuan-Ching Li⁴ 

Ting Zhang
tingzhang33@foxmail.com

Dezhi Han
dzhan@shmtu.edu.cn

Mario D. Marino
m.d.marino@leedsbeckett.ac.uk

Lin Wang
lwang@mail.xidian.edu.cn

¹ College of Information Engineering, Shanghai Maritime University, Shanghai, China

² School of Built Environment, Engineering and Computing, Leeds Beckett University, Leeds, UK

³ School of Physics and Optoelectronic Engineering, Xidian University, Xi'an, China

⁴ Department of Computer Science and Information Engineering (CSIE), Providence University, Taichung, Taiwan