A novel optimized SVM algorithm based on PSO with saturation and mixed time-delays for classification of oil pipeline leak detection

Chuang Wang\textsuperscript{a,b}, Yong Zhang\textsuperscript{b,c}, Jinbo Song\textsuperscript{d}, Qingqiang Liu\textsuperscript{d} and Hongli Dong\textsuperscript{a,b}

\textsuperscript{a}Institute of Complex Systems and Advanced Control, Northeast Petroleum University, Daqing, People’s Republic of China; \textsuperscript{b}Heilongjiang Provincial Key Laboratory of Networking and Intelligent Control, Northeast Petroleum University, Daqing, People’s Republic of China; \textsuperscript{c}Institute of Electronic Sciences, Northeast Petroleum University, Daqing, People’s Republic of China; \textsuperscript{d}School of Electrical Engineering & Information, Northeast Petroleum University, Daqing, People’s Republic of China

\textbf{ABSTRACT}

In this paper, a novel particle swarm optimization (PSO) algorithm is proposed in order to improve the accuracy of the traditional support vector machine (SVM) approaches with applications in analyzing data of oil pipeline leak detection. In the proposed saturated and mixed-delayed particle swarm optimization (SMDPSO) algorithm, the evolutionary state is determined by evaluating the evolutionary factor in each iteration, based on which the velocity updating model switches from one to another. With the purpose of reducing the possibility of getting trapped in the local optima and also expanding the search space, time-varying time-delays and distributed time-delays are introduced in the velocity updating model to respectively reflect the history of previous personal and global optimum particles. The introduction of saturation constraint ensures that the particles will converge in case that the velocity of the particles is too large. Eight well-known benchmark functions are employed to evaluate the proposed SMDPSO algorithm which is shown via extensive comparisons to outperform some currently popular PSO algorithms. To further illustrate the application potential, the developed framework SMDPSO-based SVM algorithm is exploited in the problem of oil pipeline leak detection. Experiment results demonstrate that the SMDPSO-based SVM method is superior over other well-known classification algorithms.

\textbf{ARTICLE HISTORY}

Received 28 December 2018
Accepted 19 January 2019

\textbf{KEYWORDS}

Oil pipeline leak detection; classification; particle swarm optimization; support vector machine; saturation; mixed time-delays

1. Introduction

Oil, as an important strategic resource, is needed as energy and raw materials in daily life, industrial production and aerospace industry. With the rapid development of the economy, the demand for oil is increasing, and the problem of transportation has gradually attracted people’s attention. Pipelines have the characteristics of high pressure, flammable and explosive medium, and environmental sensitivity. During pipeline transportation, accidents of pipeline leakage are frequent due to corrosion of materials, natural disasters, third-party damage and other reasons. In the event of leakage of oil pipelines, it not only causes economic losses, but also brings environmental pollution and major accidents such as destruction, casualties, etc. The safe operation of oil pipelines affects the production order, economic development, social stability and national energy security supply directly.

Therefore, it is necessary to inspect the oil pipeline regularly. After decades of development of pipeline leak detection technology, at present, there are many methods for pipeline leakage detection and location, which have been extensively applied. According to the principle of pipeline leak detection technology, it can be classified into hardware-based methods and software-based methods; direct detection methods and indirect detection methods; internal detection methods and external detection methods. Among these methods, hardware-based methods and software-based methods are widely used in the pipeline leakage detection. Hardware-based methods include distributed optical fibre detection method, magnetic flux leakage detection method, gas infrared imaging detection method, radioactive tracing detection method, characteristic impedance detection method and self-organizing wireless sensor detection. Software-based methods include sound wave detection, wavelet analysis, negative pressure wave detection, flow balance detection, generalized correlation analysis, support vector machine detection, fuzzy-neural network detection, statistical detection and real-time model detection.
In actual pipeline operating conditions, there are fewer pipeline leakage samples. Due to the advantages of small sample learning and strong generalization ability, support vector machines (SVM) have been applied to pipeline leakage detection by many scholars. SVM has been used in Qu, Feng, Zeng, Zhuge, and Jin (2010) to classify and identify characteristics of distributed optical fibre sensor vibration signal based on extraction, so as to judge whether there is leakage along the pipeline. In Wang, Zhang, and Liang (2010), independent quantitative analysis has been applied to noise reduction of pressure detection signal on the basis of SVM. The improved SVM is trained by collecting samples of 100 different working conditions and 24 leakage working conditions. Compared with 77.4% of the BP neural network, its classification accuracy is 90.3%. In Sun, Xiao, Wen, and Zhang (2016), according to the leakage of different apertures, the effective numerical entropy (RMS entropy) of the signal is used as the eigenvector to train the SVM. The experiment is carried out in a test pipeline of 62 metres. When the leakage aperture is 5 mm, the accuracy of the test results reaches 95%.

So far, SVM, as a popular supervised learning algorithm, has been successfully applied to solve data classification and regression problems due to its excellent performance and small computation cost. It should be mentioned that although the SVM detection method has the ability to classify the pipeline leakage signal, how to choose the appropriate SVM kernel function and penalty parameters is very important for identifying pipeline leakage signal effectively. Currently, a series of optimization algorithms is used to optimize the parameters of SVM (e.g. genetic algorithm (Huerta, Duval, & Hao, 2006), particle swarm optimization algorithm (Huang & Dun, 2008), artificial bee colony algorithm (Alshamlan, Badr, & Alohal, 2016)). It avoids the empirical allocation of penalty parameters to SVM, and the choice of kernel function improves the accuracy of pipeline leak condition identification. Therefore, it is a natural idea to optimize SVM parameters using the evolutionary computation (EC) algorithm to maximize the effectiveness of SVM. In order to find a suitable EC algorithm, the PSO algorithm has been proposed by Kennedy and Eberhart (1995) as an appropriate candidate algorithm. PSO is a population-based stochastic optimization algorithm, which simulates the behaviour of birds, fish and other organisms.

The PSO algorithm simulates the group intelligent optimization algorithm for the foraging flight of the bird population in the biological world. The PSO algorithm is simple and easy to implement. However, the PSO algorithm has some congenital inadequacies, such as easy premature convergence, trap local optima and weak target. In the last few decades, various modified PSO algorithms have been proposed to improve the search ability of PSO algorithms and reduce the possibility of falling into local optima (Song, Wang, & Zou, 2017; Tang, Wang, & Fang, 2011; Zeng, Wang, Zhang, & Alsaadi, 2016; Zhan, Zhang, Li, & Chung, 2009). In Zhan et al. (2009), an adaptive PSO (APSO) algorithm of system parameter based on evolutionary factors has been proposed for automatic control of parameters such as inertial weight and acceleration coefficient. A switching PSO (SPSO) algorithm has been developed in Tang et al. (2011), in which the velocity model is updated according to the Markov chain, thereby exploring the search space more thoroughly than the APSO algorithm. In addition, a switching delayed PSO (SDPSO) algorithm has been presented in Zeng, Wang, et al. (2016) to further enhance search capability by utilizing delay information (including previous personal best and global optimal particles). At the same time, a switching local evolutionary PSO algorithm has been introduced in Zeng, Zhang, Chen, Chen, and Liu (2016) to plan path for intelligent robot. More recently, a multimodal delay PSO (MDPSO) algorithm has been introduced in Song et al. (2017), which adds multi-modal time-delays to the velocity update model to reduce the possibility of falling into the local optimum and expand the search space. PSO has better global optimum performance, but the local search ability is poor. At the same time, the problem of convergence is not considered.

It should be noted that the PSO algorithm introduces saturation and time-delays into the model of velocity update, which achieves better results. Saturation can control the particle velocity, too high velocity will make the system not converge, using \( \zeta \) instead of velocity of 0 can ensure that the system is dynamic stable and has better robustness. On the other hand, time-delayed information is fully utilized to expand the search space so as to reduce the possibility of falling into local optimization. By introducing time-varying delays, the optimal position of the particle can be fully utilized. Distributed time-delays is the signal propagation delay distributed through a certain number of parallel paths at a given time. So far, systems with distributed time-delays have been well studied, such as complex networks, sensor networks and networked control systems. The introduction of distributed time-delays can make full use of the local optima of all particles to reach the effect of jumping out of the local optima and enlarging the search space. A seemingly natural idea is proposed to introduce the saturation term, time-varying time-delays and distributed time-delays into the velocity update model to make better use of the historically optimal particles. Therefore, the main purpose of this paper is to study the saturated and mix-delayed particle swarm optimization (SMDPSO) algorithm and
apply it to the classification of pipeline leakage detection data.

PSO algorithm has been proved to be a strong competitor of other optimization algorithms in SVM optimization. For instance, PSO optimized support vector machine has been used in Subasi (2013) to classify EMG signals for the diagnosis of neuromuscular diseases. A hybrid PSO/SVM and GA/SVM method has been mentioned in Alba, Garcia-Nieto, Jourdan, and Talbi (2007) to be used in gene selection in cancer classification. An application of the PSOCSVM model has been developed in Ranaee, Ebrahimzadeh, and Ghaderi (2010) for recognition of control chart patterns. A hybrid PSOCSVM method has been proposed in Selakov, Cvijetinović, Milović, Mellon, and Bekut (2014) to predict the short-term load in Burbank city when the temperature changes greatly. A hybrid PSOCSVM-based method has been introduced in Nieto, Garcia-Gonzalo, and Lasheras (2015) to predict the remaining service life of aircraft engines and evaluate their reliability. Very recently, a new switching-delayed-PSO-based optimized SVM algorithm in Zengetal.(2018) has been proposed to analyze the data to realize oil pipeline leakage detection. With proper classification, pipeline leaks can be discovered accurately and quickly, which greatly reduces economic losses.

2. PSO algorithms

The PSO algorithm which is proposed by Kennedy to imitate the behavior of fish schooling or birds flocking is an evolutionary algorithm, and has been applied to deal with a large number of optimization problems.

In PSO algorithm, each particle of the swarm is regarded as a feasible candidate of the problem. Consider a swarm with all the particle in a D-dimensional target search space. In the kth iteration, the velocity and position of the ith particle are represented by two vectors, namely the velocity vector \( v_i(k) = (v_{i1}(k), v_{i2}(k), \ldots, v_{iD}(k)) \) and the position vector \( x_i(k) = (x_{i1}(k), x_{i2}(k), \ldots, x_{iD}(k)) \), respectively. In the iterative process, in order to get global optimum, the velocity of each particle is updated according to its own optimal position (pbest) represented by \( p_i = (p_{i1}, p_{i2}, \ldots, p_{iD}) \), and the global position (gbest) represented by \( g = (g_1, g_2, \ldots, g_D) \). The velocity and the position of the ith particle at the \( (k+1) \)th iteration are updated as follows:

\[
\begin{align*}
  v_i(k+1) &= \omega v_i(k) + c_1 r_1 (p_i(k) - x_i(k)) \\
             &\quad + c_2 r_2 (g(k) - x_i(k)), \\
  x_i(k+1) &= x_i(k) + v_i(k+1),
\end{align*}
\]

where \( k \) represents the current iteration number; \( \omega \) represents the inertia weight; \( c_1 \) and \( c_2 \) are the acceleration coefficients called as cognitive and social parameters, \( r_1 \) and \( r_2 \) are two random numbers which are uniformly distributed on the interval \([0, 1]\).

In the past decade, a host of well-known improved algorithms have been proposed to promote the searching performance of the traditional PSO algorithm. The larger inertia weight is good for the particle to wake up the global exploration, and the smaller weight is beneficial for the particle to trend to the local exploration. Therefore, linearly decreasing inertia weights (PSO-LDIW) have been proposed by Eberhart in Shi and Eberhart (1999), where \( \omega \) is given as follows:

\[
\omega = (\omega_1 - \omega_2) \times \frac{\text{iter}_{\text{max}} - \text{iter}}{\text{iter}_{\text{max}}} + \omega_2,
\]

where \( \omega_1 \) and \( \omega_2 \) represent the initial and final value of the inertia weight, respectively; the current number of
iteration is represented by \( \text{iter} \) and the maximum iteration number is represented by \( \text{iter}_{\text{max}} \). Moreover, a PSO with time-varying acceleration coefficients (PSO-TVAC) has been proposed in Ratnaweera, Halgamuge, and Watson (2004), and two acceleration coefficients \( c_1 \) and \( c_2 \) are linearly decreased and increased as follows:

\[
\begin{align*}
    c_1 &= (c_{1i} - c_{1f}) \times \frac{\text{iter}_{\text{max}} - \text{iter}}{\text{iter}_{\text{max}}} + c_{1f}, \\
    c_2 &= (c_{2i} - c_{2f}) \times \frac{\text{iter}_{\text{max}} - \text{iter}}{\text{iter}_{\text{max}}} + c_{2f},
\end{align*}
\]

where \( c_{1i} \) and \( c_{2i} \) denote the initial values of the acceleration coefficients, \( c_{1f} \) and \( c_{2f} \) represent the final value of the cognitive acceleration coefficient \( c_1 \), and the social acceleration coefficient \( c_2 \), respectively. It should be touched that the parameters \( c_{1i} = 2.5, c_{1f} = 0.5, c_{2i} = 0.5 \) and \( c_{2f} = 2.5 \) are determined based on experience.In addition, in order to the performance of searching, a PSO with the constriction factor (PSO-CK) has been developed in Clerc and Kennedy (2002), where the recommended inertia weight, as well as acceleration coefficients, are 0.729 and 1.49, respectively. Note that all of above variants of PSO algorithms have mainly focused on adjusting parameters of the PSO algorithms, but the parameters can not be adjusted adaptively. Furthermore, an adaptive PSO (APSO) has been put forward in Zhan et al. (2009) to overcome the above defect, in which an evolutionary factor is applied to identify four evolutionary states of the whole particle swarm, including the exploration state, the exploitation state, the convergence state and the jumping-out state. Thus the inertia weight and acceleration coefficients are controlled according to the four states defined by the evolutionary factor. And then, an SPSO has been proposed to optimize the APSO. In SPSO, Markov chain has been introduced to predict the next state from the current state. Recently, the time-delays have been added into the velocity updating model of the SDPSO in Zeng, Zhang, Liu, Liang, and Alsaadi (2017). More recently, The MDPSO algorithm has been proposed in Song et al. (2017) where have introduced multimodal delays and new terms into the model updating velocity of the traditional PSO algorithm according to the evaluated evolution state.

### 3. A novel SMDPSO algorithm

In this section, a novel SMDPSO algorithm is proposed to further improve the searching ability of the traditional PSO algorithm. The main novelty of the proposed SMDPSO lies in the introduction of the mixed time-delays and the saturation in the velocity updating model. At the same time, the nonlinear inertia weight is addicted to increases the searching ability of PSO algorithm. On the other hand, the artificial fish swarm algorithm in Wang, Zhang, Wang, and Liang (2015) is introduced to defined four states, foraging behaviour, clustering behaviour, following and stochastic behaviour. SMDPSO algorithm overcomes the shortcomings of the basic PSO algorithm that it is difficult to jump out of the local optima and to guarantee convergence. (1) make better use of accumulated history about the population evolution with better accuracy; (2) pursue stronger capability of avoiding local optima trapping problems; (3) maintain an appropriate balance between convergence and diversity; (4) the addition of saturation guarantees the convergence of PSO algorithm and (5) the addition of nonlinear inertia weights enhances the spatial search capability of the SMDPSO algorithm.

In the proposed SMDPSO algorithm, the velocity and position equations are updated according to the evolutionary state depending on the evolutionary factor and Markov chain. The searching characteristics of the PSO algorithm are revealed through the four evolutionary behaviours, foraging behaviour, clustering behaviour, following and stochastic behaviour denoted by \( \xi(k) \) = 1, \( \delta(k) \) = 2, \( \theta(k) \) = 3 and \( \eta(k) \) = 4, respectively.

As mentioned in Zhan et al. (2009), the evolutionary factor is calculated based on the distance between the particles. The mean distance between the \( i \)-th particle and other particles denoted by \( d_i \) is given as follows:

\[
d_i = \frac{1}{S} \sum_{j=1}^{S} \sqrt{\sum_{k=1}^{D} \left(x_{ik} - x_{jk}\right)^2},
\]

where \( S \) denotes the swarm size and \( D \) represents the dimension of the particle. The evolutionary factor denoted by \( \delta \) is shown as follows:

\[
\delta = \frac{d_g - d_{\text{min}}}{d_{\text{max}} - d_{\text{min}}},
\]

where \( d_g \) represents the global best particle among \( d_i \), \( d_{\text{max}} \), \( d_{\text{min}} \) represent the maximum and minimum of \( d_i \) in the swarm, respectively. In this paper, the equal division strategy is employed to classify the four evolutionary states represented by \( \xi(k) \) as follows:

\[
\xi(k) = \begin{cases} 
1, & 0 \leq \delta < 0.25, \\
2, & 0.25 \leq \delta < 0.5, \\
3, & 0.5 \leq \delta < 0.75, \\
4, & 0.75 \leq \delta \leq 1,
\end{cases}
\]

where \( \xi(k) = 1, 2, 3, 4 \) represent the foraging behaviour, clustering behaviour, following and stochastic behaviour, respectively. In this paper, \( \Delta \) is defined as \( \Delta = \text{sign}(v_i(k)) \min\{v_i, v_{\text{max}}(k), \|v_i(k)\|\} \). The velocity of saturation function
is defined as follows:

\[
\sigma(v_i(k)) = \begin{cases} 
\zeta, & v_i(k) = 0, \\
\Delta, & v_i(k) \neq 0,
\end{cases}
\]  

where \( \sigma(\cdot) \) is a saturation function. \( \Delta = \text{sign}(v_i(k)) \min \{v_{i,\text{max}}(k), \|v_i(k)\|\} \)

if \( v_{i,\text{max}}(k) > \|v_i(k)\| \)

\( \sigma(v_i(k)) = v_i(k) \)

else if \( v_i(k) < 0 \)

\( \sigma(v_i(k)) = -v_{i,\text{max}}(k) \)

else

\( \sigma(v_i(k)) = v_{i,\text{max}}(k) \)

where \( v_{i,\text{max}}(k) \) is the \( i \)th element in saturation level \( v_{\text{max}}(k) \).

Remark 1 In literature [22], the switching time-delay is introduced into the velocity update model, which effectively reduces the possibility of falling into the local optima. In this paper, when introducing the time-delay, the saturation constraint is added to the velocity, which limits the velocity constraint to a certain range and effectively ensures the convergence of the system.

Sigmoid function is a common s-shaped function in biology, also known as s-shaped growth curve. In information science, sigmoid function is often used as the threshold function of neural network to map variables to between 0 and 1 due to its singularities and inverse singularities. the sigmoid function in the neural network is introduced as a nonlinear inertia weight \( \omega \) as follows:

\[
\omega = \frac{1}{1 + \exp(-\delta)},
\]

where \( \omega \in [0.5, 0.731] \).

In this paper, a novel velocity updating strategy is demonstrated for four behaviour as follows:

1. Foraging behaviour is denoted by \( \varepsilon(k) = 1 \), the velocity and the position of the \( i \)th particle at the \((k + 1)\)th iteration are updated as follows:

\[
v_i(k + 1) = \omega \sigma(v_i(k)) + c_1 r_1 (p_i(k) - x_i(k)) + c_2 r_2 (p_g(k) - x_i(k)),
\]

\[x_i(k + 1) = x_i(k) + v_i(k + 1),
\]

where \( \omega \) is the inertia weight defined in Equation (9), acceleration coefficients \( c_1 \) and \( c_2 \) are updated according to Equations (3) and (4), the particles are trying to fly into the globally optimal region as soon as possible. Therefore, the velocity updating model in the traditional PSO algorithm is employed.

2. Clustering behaviour is denoted by \( \varepsilon(k) = 2 \), the velocity and the position of the \( i \)th particle at the

\((k + 1)\)th iteration are updated as follows:

\[
v_i(k + 1) = \omega \sigma(v_i(k)) + c_1 r_1 (p_i(k) - x_i(k)) + c_2 r_2 (p_g(k) - x_i(k)),
\]

\[x_i(k + 1) = x_i(k) + v_i(k + 1),
\]

where \( \varepsilon_1(k) \) represents time-varying time-delays and satisfies \( d_m \leq \tau_1(k) \leq d_M \), where \( d_m \) and \( d_M \) are normal numbers, which are the upper and lower limits of the time-varying time-delays, respectively. This behaviour enables the particles to refer to the optimal position when they have experienced in determining the direction of the next step, reducing the blindness of particles movement, and the particles in the group are willing to remain in the area around the local optimal particles. Therefore, Time-varying time-delays are introduced in local search to reduce falling into local optima.

3. Following behaviour is denoted by \( \varepsilon(k) = 3 \), the velocity and the position of the \( i \)th particle at the \((k + 1)\)th iteration are updated as follows:

\[
v_i(k + 1) = \omega \sigma(v_i(k)) + c_1 r_1 (p_i(k) - x_i(k)) + c_2 r_2 \sum_{\tau_2 = 1}^{N} (p_g(k - \tau_2) - x_i(k)),
\]

\[x_i(k + 1) = x_i(k) + v_i(k + 1),
\]

where \( N \) represents the upper bound of the distributed time-delays, \( \tau_2(k) \) represents distributed time-delays to meet \( p_g(k - \tau_2) = 0, \forall (k - \tau_2) \in Z^- \). This behaviour allows the particles to refer to the optimal position of the particles when determining the next step. The communication and sharing between individuals in the particle swarm search process is strengthened, and the blindness of particle search is further reduced. It is important to search for the best location as much as possible. Therefore, Distributed time delay is introduced to explore the entire search space in a more thorough way.

4. Stochastic behaviour is denoted by \( \varepsilon(k) = 4 \), the velocity and the position of the \( i \)th particle at the \((k + 1)\)th iteration are updated as follows:

\[
v_i(k + 1) = \omega \sigma(v_i(k)) + c_1 r_1 (p_i(k) - x_i(k)) + c_2 r_2 \sum_{\tau_2 = 1}^{N} (p_g(k - \tau_2) - x_i(k)),
\]

\[x_i(k + 1) = x_i(k) + v_i(k + 1),
\]

the particles that are stuck in local optima are hoping to jump out of local optima. Therefore, stochastic behaviour is introduced to provide energy for these particles to escape from the region.

The flowchart of the novel SMDPSO algorithm is given in Figure 1.
4. Simulation experiments

4.1. Selection of Benchmark functions

In this section, the performances of the novel SMDPSO algorithm are proven through a series of simulation experiments. The benchmark functions from (14) to (21) are typical test functions, and they are difficult to get the optimal solution. It is important to note that the details of the benchmark function are shown in Table 1, including the function number, function name, threshold, search space for each dimension, dimension and a minimum value of the benchmark function.

Note that all the benchmark functions are high-dimensional problems. The Sphere function $f_1(x)$ is generally exploited to evaluate the convergence rate of the algorithm. The Rosenbrock function $f_2(x)$, which is viewed as a mono-modal optimization problem is often used to verify not only the local search but also the global search ability. The Schwefel 1.2 function $f_3(x)$ and the Schwefel 2.21 function $f_6(x)$ are single-peak functions that are easily trapped in local optima. The Griewank function $f_4(x)$ is a typical nonlinear multimodal function. The extensive search space is usually considered as a complex multimodal problem which is difficult to deal with by the optimization algorithm. The Penalized 1 function $f_5(x)$ and the Schwefel 2.21 function $f_6(x)$ are irregular functions, which is difficult to converge to the optimal value. The Step function $f_7(x)$ is a common benchmark function which is widely used to test the convergence of optimization algorithms. The Penalized 1 function $f_5(x)$ and The Penalized 2 function $f_8(x)$ are multi-peak functions, it is difficult to find the optimal value.

$$
\text{Sphere} : f_1(x) = \sum_{i=1}^{D} x_i^2. \quad (14)
$$

$$
\text{Rosenbrock} : f_2(x) = \sum_{i=1}^{D} 100((x_{i+1} - x_i)^2 + (x_i - 1)^2). \quad (15)
$$

$$
\text{Schwefel 1.2} : f_3(x) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2. \quad (16)
$$

$$
\text{Griewank} : f_4(x) = \frac{1}{4000} \sum_{i=1}^{D} x_i^2 + 1 - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right). \quad (17)
$$

$$
\text{Penalized 1} : f_5(x) = \frac{\pi}{D} (10 \sin^2(\pi y(1)) + \sum_{i=1}^{D} (y(i) - 1)^2 (1 + \sin^2(10\pi y(i + 1))) + (y(D) - 1)^2) + \sum_{i=1}^{D} \mu(x(i), 10, 100, 4). \quad (18)
$$

$$
\text{Schwefel 2.21} : f_6(x) = \max|x_i|, 1 \leq i \leq D. \quad (19)
$$

$$
\text{Step} : f_7(x) = \sum_{i=1}^{D} (|x_i + 0.5|)^2. \quad (20)
$$
Penalized 2: 
\[ f_8(x) = 0.1 \left( \sin^2(3 \pi x(1)) + \sum_{i=1}^{D-1} (x(i) - 1)^2 \right. \]
\[ \times \left( 1 + \sin^2(3 \pi x(1 + i)) \right) \]
\[ + (x(D) - 1)^2 + (x(D) - 1)^2 \]
\[ \times (1 + \sin^2(2 \pi x(D))) \]
\[ + \sum_{i=1}^{D} \mu(x(i), 5, 100, 4). \] (21)

4.2. Experiment results of the SMDPSO algorithm

In this section, a host of simulation experiments are employed to illustrate the performance of the introduced SMDPSO algorithm. The parameter settings of the simulation experiments are shown as follows: \( S = 20 \) for the particle swarm, \( D = 20 \) for the dimension of each particle, \( K = 20,000 \) for the maximum iteration of each experiment, and \( T = 50 \) for the repeated times of each experiment. For the SMDPSO algorithm, the inertia weight \( w \) is decreased from 0.731 to 0.5. The acceleration coefficients \( c_1 \) and \( c_2 \) are belonging to \([0.5, 2.5]\). The setting of the time-delay \( \tau \) is determined based on the simulation results. The performance of the SMDPSO algorithm in the 20-dimensional search space with settings of the upper bound of the distributed time-delay \( N = 100 \). The upper and lower limits of the saturation constraint are 20 and \(-20\), respectively.

The superiority of the proposed SMDPSO algorithm is demonstrated over six popular PSO algorithms, including the PSO-LDIW (Shi & Eberhart, 1999), PSO-TVAC (26), PSO-CK (4), SPSO (Tang et al., 2011), SDPSO Zeng, Wang, et al. (2016) and MDPSO (Song et al., 2017). The performance tests for the proposed SMDPSO algorithms are shown in Figure 2 to Figure 9. The vertical coordinate represents the logarithmic formation of the mean fitness value of all the tested PSO algorithms, and the horizontal coordinate denotes the number of iteration for Figure 2 to Figure 9. Additionally, detailed information of the optimization performance is listed in Table 2, where the mean, minimum and the standard deviation of the fitness value with respect to each benchmark function is presented to demonstrate the algorithms performance as well as the successful convergence ratio.

The results show that the proposed SMDPSO algorithm is superior to other PSO algorithms in the mean, minimum and standard deviation of fitness values of functions (14)–(21). It is worth saying that the mean fitness of SMDPSO algorithm is smaller than others, which shows the superiority of SMDPSO algorithm in achieving global optimum. In particular, the SMDPSO algorithm shows great superiority over other PSO algorithms, for functions (15) and functions (17). In terms of the optimal mean fitness, although the SMDPSO algorithm is not the smallest for other functions, the SMDPSO algorithm shows better competitive performance than the PSO-LDIW algorithm, PSO-TVAC algorithm, PSO-CK algorithm and SPSO algorithm.

Another important indicator in verifying the convergence performance of the optimization algorithm is the successful convergence rate. Table 2 shows the global optimal convergence rate of all benchmark functions, from which we can see that the convergence rate is not always 100%. It is indicated that not possible for some functions to achieve global optimal convergence. It is worth mentioning that the SMDPSO algorithm is superior to other PSO algorithms in the convergence rate. Because the Griewank function has a large number of local minimum values, it is difficult to detect the global minimum value, so its successful convergence rate can not reach 100%. We can see that all the testing PSO algorithms have low successful convergence ratio for function (17) which are 12%, 14%, 28%, 26%, 22%, 16%, 40%, respectively. Nevertheless, the improved SMDPSO algorithm still has the best convergence among all algorithms, which indicates that it has better performance than other PSO algorithms.

The plots of the convergence rate for testing algorithms are depicted in Figure 2 to Figure 9. Obviously the convergence rate of the SMDPSO algorithm is faster than that of other PSO algorithms for the function (14), which means that the saturation term in SMDPSO guarantees the convergence of the algorithm. In addition,
it can be seen that SMDPSO algorithm has lower mean fitness value and higher successful convergence rate, and can reach the global optimal robustness for all benchmark functions. Among the benchmark functions of single-mode and multi-mode optimization, the proposed SMDPSO algorithm is superior to six well-known PSO algorithms, indicating that SMDPSO algorithm can get rid of local optimization. For instance, the SMDPSO algorithm can solve the optimization problem with satisfactory convergence speed and convergence accuracy.

5. The SMDPSO-based SVM algorithm

5.1. The SVM algorithm

Support vector machine (SVM) algorithm that was proposed by Cortes and Vapnik in 1995, shows many unique advantages in solving small samples, non-linear and high-dimensional pattern recognition. Based on the principle of Structural risk minimization (SRM), SVM can achieve the best learning ability according to the limited samples and can make regression prediction effectively.

Note that all the samples are set \( D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\} \) in the regression problem, the original sample space is mapped into the feature space of high dimensions by introducing a non-linear reflection spline \( x \) to input \( n \) dimension and output 1 dimension of sample set \( D \). The decision function is supposed to be as follows:

\[
f(x) = \omega \varphi(x) + b,
\]

where \( \omega \) is the weight vector, \( \varphi(x) \) is the eigenvector after mapping \( x \), and \( b \) is the threshold.

According to the principle of structural risk minimization, the relaxation variability \( \xi_i \) of the algorithm is introduced, and SVM can be formalized into the quadratic programming problem described as follows:

\[
\begin{align*}
\min & \quad \frac{1}{2} ||\omega||^2 + C \sum_{i=1}^{m} \xi_i^2, \\
\text{s.t.} & \quad y_i(\omega \varphi(x) + b) = 1 - \xi_i, \\
& \quad \xi_i \geq 0, i = 1, 2, \ldots, m.
\end{align*}
\]

where \( C (C > 0) \) is the penalty parameter. The penalty factor \( C \) affects the loss value of the objective function, that is, the error value. More specifically, the larger the \( C \), the greater the error, especially when the loss function is given. When the error is large, the SVM is prone to overfitting. Otherwise, when \( C \) is too small, SVM may have the problem of inadequate fitting.
The construction of Lagrange’s function is available as follows:

\[ L(\omega, b, \alpha, \xi, \mu) = \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^{m} \xi_i^2 - \sum_{i=1}^{m} \alpha_i[y_i(\omega \varphi(x) + b) - 1 + \xi_i], \]

(24)

where \( \mu_i \geq 0, \alpha_i \geq 0. \)

The following equation constraints can be obtained from the KKT condition as follows:

\[ \omega = \sum_{i=1}^{m} \alpha_i y_i \varphi(x_i), \]

\[ \sum_{i=1}^{m} \alpha_i y_i = 0, \]

\[ \alpha_i = C \xi_i, \]

\[ y_i(\omega \varphi(x) + b) - 1 + \xi_i = 0. \]

(25)

According to Hilbert–Schmidt principle, the kernel function satisfying Mercer condition is introduced as follows:

\[ k(x_i, x_j) = \varphi(x_i)^T \varphi(x_j). \]

(26)

The kernel method is introduced and the form of feature map is considered, then the final decision function is obtained as follows:

\[ f(x) = \text{sgn}\left( \sum_{i=1}^{m} \alpha_i y_i k(x_i, x_j) + b \right). \]

(27)

where \( \alpha_i \) is the optimal Lagrangian coefficient, \( b \) is the optimal value of \( b \), \( \text{sgn} \) denotes a symbolic function. The commonly used kernel functions (Linear kernel function, polynomial kernel function, gaussian kernel function, Laplace kernel function and sigmod kernel function) are as shown:

\[ k(x_i, x_j) = \varphi(x_i)^T \varphi(x_j). \]

(28)

\[ k(x_i, x_j) = ( \varphi(x_i)^T \varphi(x_j) )^d. \]

(29)

\[ k(x_i, x_j) = \exp\left( - \frac{\|x_i - x_j\|^2}{2\sigma^2} \right). \]

(30)

\[ k(x_i, x_j) = \exp\left( - \frac{\|x_i - x_j\|}{\sigma} \right). \]

(31)

\[ k(x_i, x_j) = \tanh(\beta x_i^T x_j + \theta). \]

(32)

For the complex classification problem, due to different sources of data, it may be composed of heterogeneous data sets, which leads to the complexity of data distribution. If only the SVM model of a single kernel function is selected, it is difficult to achieve satisfactory classification performance.

For the kernel functions described in Equations (28)–(32), they can be divided into global and local kernel functions according to their effects on classification problems. The typical representative of global kernel functions is polynomial kernel functions. Its outstanding feature is that the generalization performance is strong and the number of samples can be well extracted. As one of the typical representatives of local kernel function, Gaussian kernel function has a good learning ability for samples within a certain distance, but its generalization ability is relatively poor. Therefore, considering the advantages and disadvantages of the two kernels, the above two kernels are combined. According to certain weights, the final kernel function is formed, which can improve the comprehensive performance of the model.

According to the above analysis, if the weight coefficient of Gaussian kernel \( k_G \) is \( \alpha \) and the polynomial kernel \( k_P \) is \( 1 - \alpha \) in the final mixed kernel function, the final mixed kernel function is obtained as follows:

\[ k = \alpha k_G + (1 - \alpha) k_P. \]

(33)

In this paper, the SVM needs to select four model parameters, the positive normalization constant \( C \), the width of Gaussian kernel \( \sigma \), polynomial times \( d \) and the weight of final mixed kernel function combination \( \alpha \). \( C \) determines the size of training error and the strength of generalization ability; \( \sigma \) and \( d \) reflect the distribution characteristics of training samples. These parameters affect the predictive ability and algorithm efficiency of the model directly. Usually, the optimal parameter combination is determined by the comparison of multiple parameter combinations such as cross validation or gradient descent. However, practical experience shows that these methods have many human factors, large blindness and low efficiency. PSO algorithm not only has the ability of fast solution, but also can carry out global search. In this paper, the optimal value of SVM model parameters is obtained through SMDPSO algorithm.

### 5.2. The SMDPSO-based SVM model

A hybrid classifier, namely, SMDPSO-based SVM model, is proposed to enhance the classification performance by utilizing the SMDPSO to optimize the positive normalization constant \( C \), the width of Gaussian kernel \( \sigma \), polynomial times \( d \) and the weight of final mixed kernel function combination \( \alpha \). The classification accuracy of SVM is selected as the fitness function of SMDPSO, which is formulated as follows:

\[ ACC = \frac{TP + TN}{TP + FP + TN + FN}. \]

(34)
Figure 3. Function performance test of Rosenbrock.

Figure 4. Function performance test of Schwefel 1.2.

Figure 5. Function performance test of Griewank.

Figure 6. Function performance test of Penalized 1.

where TP, FP, TN and FN indicate true positives, false positives, true negatives and false negatives, respectively. The flowchart of proposed SMDPSO-SVM algorithm is shown in Figure 10.

6. Results and analysis of the SMDPSO-based SVM algorithm

Pipelines are widely used to transport oil in the oil field. As a result of natural factors, human factors and other factors, it is easy to cause oil leakage in pipeline transportation. The loss and harm caused by the leakage of the oil pipeline make people have to pay attention to the detection of the leakage of the oil pipeline. Therefore, the research of oil pipeline leakage detection technology has become a hot topic. At present, there are mature methods for leak detection of oil pipeline, but with the development of industrial technology, the accuracy of oil pipeline leak detection is also increasing. Therefore, the detection technology of oil pipeline leakage is put forward higher request.
6.1. Data pre-processing

In this section, the classification performance of the SVM algorithm based on SMDPSO is evaluated from the aspect of accuracy, and the error curve is given. The data is provided by the oil and gas pipeline leakage detection simulation experiment platform of Northeast Petroleum University. A total of 470 groups were collected, of which 300 were used for training and 170 were used for testing. The schematic diagram of pipeline leakage simulation experiment system is shown in Figure 11.

The test bench is the HD-II type pipeline leak detection system. The pipe has a length of 160m, a pipe wall thickness of 1 cm, and a pipe diameter of DN 50 (nominal diameter). The pipeline is equipped with a leak valve every 10m. The experimental platform has a total of 15 leak points. Each leak point is connected by a 4-point ball valve, and a 4-point ball valve is used to simulate the leakage of the pipeline. The pipeline pressure is 0.5 MPa, the flow rate is 60 m³/h and the ambient temperature is 24.3°C. The pipe layout plan is shown in Figure 12.

The simulation experiment software uses the LABVIEW development environment, and data is collected using the NI-9215 acquisition card. The sampling frequency is 5KHz, and the piezoelectric acoustic wave sensor is used to transmit the data to the computer. The simulation experiment signals mainly include normal signals and leakage signals. Among them, the leakage signal is obtained by quickly switching the 4-point ball valve switch to simulate the pipeline leakage. Install a 10m high-pressure sonic attenuator at the leak point, and install a 4-point ball valve and a plug with a leakage aperture of 1mm at the end to reduce the impact of the vibration caused by the diverter switch valve on the signal’s own attenuation Figures 13–17.

6.2. Experiments results of the SMDPSO-based SVM algorithm

In this paper, the proposed SMDPSO-based SVM, together with back propagation neural network (BP-NN), stack sparse self-encoding (SSAE) (including three-layer network, four-layer network and five-layer network) is applied to the classification of long-distance pipeline
leakage data. The accuracy of various classification methods are 90.588% (GA-NN), 93.529% (SSAE3), 94.706% (SSAE4), 98.235% (SSAE5), 99.417% (SMDPSO-based SVM). By analyzing the accuracy and the images of error curve,
the proposed SMDPSO-based SVM demonstrates satisfactory and superiority over others.

7. Conclusion

In this paper, a new PSO algorithm is proposed for the sake of improving the accuracy of the traditional support vector machine approaches with applications in analysing data on oil pipeline leak detection. In the proposed saturated and mix-delayed particle swarm optimization algorithm, the introduction of time-delays reduces the possibility of falling into local optima, the introduction of saturated term ensures convergence and the addition of nonlinear inertial weight increases the diversity of particles, thus greatly enhancing the search ability of the algorithm. Eight famous benchmark functions are used to evaluate the proposed SMDPSO algorithm, and a large number of comparisons prove that the performance of SMDPSO algorithm is better than some currently popular PSO algorithms. In order to further illustrate the application potential of this algorithm, the SMDPSO algorithm is applied to oil pipeline leakage detection. Experimental results show that SMDPSO-based SVM is superior to other well-known classification algorithms. In order to further illustrate the application potential of this algorithm, the SMDPSO algorithm is applied to oil pipeline leakage detection. Experimental results show that SMDPSO-based SVM is superior to other well-known classification algorithms. Future work can be summarized into two aspects: (1) How to improve the search performance of PSO and apply it to higher dimensional and more complex space; (2) How to prove the convergence of PSO from theory.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported in part by the National Natural Science Foundation of China [grant number 61873058], the China Postdoctoral Science Foundation under Grant 2017M621242, the PetroChina Innovation Foundation under Grant 2018D-5007-0302, the Natural Science Foundation of Heilongjiang Province of China under Grant F2018005, the Open Fund of the Key Laboratory for Metallurgical Equipment and Control of Ministry of Education in Wuhan University of Science and Technology under Grant 2018A01, the Northeast Petroleum University Innovation Foundation For Postgraduate JYCX_CX06_2018, YJSCX2017-026NEPU and the Alexander von Humboldt Foundation of Germany. JYCX CX06 2018, YJSCX2017-026

References

Alba, E., Garcia-Nieto, J., Jourdan, L., & Talbi, E. (2007). Gene selection in cancer classification using PSO/SVM and GA/SVM hybrid algorithms. In IEEE congress on evolutionary computation (pp. 284–290). Singapore.

Alshamlan, H. M., Badr, G. H., & Alohalie, Y. A. (2016). Artificial bee colony and SVM method for microarray gene selection.
and multi class cancer classification. *International Journal of Machine Learning and Computing, 6*(3), 184–190.

Clerc, M., & Kennedy, J. (2002). The particle swarm: Explosion, stability, and convergence in a multi-dimensional complex space. *IEEE Transactions on Evolutionary Computation, 6*(1), 58–73.

Eberhart, R. C., & Kennedy, J. (1995, October). A new optimizer using particle swarm theory. In *Proceedings of the 6th International Symposium on Micro Machine and Human Science* (pp. 39–43). Nagoya, Japan.

Huang, C. L., & Dun, J. F. (2008). A distributed PSO-SVM hybrid system with feature selection and parameter optimization. *Applied Soft Computing, 8*(4), 1381–1391.

Huerta, E. B., Duval, B., & Hao, J. K. (2006). A hybrid GA/SVM approach for gene selection and classification of microarray data. In *Workshops on Applications of Evolutionary Computation* (pp. 34–44). Berlin: Springer.

Nieto, P. J. G., Garcia-Gonzalo, E., & Lasheras, F. S. (2015). Hybrid PSO-SVM-based method for forecasting of the remaining useful life for aircraft engines and evaluation of its reliability. *Reliability Engineering & System Safety, 138*, 219–231.

Qu, Z., Feng, H., Zeng, Z., Zhuge, J., & Jin, S. (2010). A SVM-based pipeline leakage detection and pre-warning system. *Measurement, 43*, 513–519.

Ranee, V., Ebrahimzadeh, A., & Ghaderi, R. (2010). Application of the PSO-SVM model for recognition of control chart patterns. *ISA Transactions, 49*(4), 577–586.

Ratnaweera, A., Halgamuge, S. K., & Watson, H. C. (2004). Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computation, 8*(3), 240–255.

Selakov, A., Cvijeticinović, D., Milović, L., Mellon, S., & Bekut, D. (2014). Hybrid PSO-SVM method for short-term load forecasting during periods with significant temperature variations in city of Burbank. *Applied Soft Computing, 16*, 80–88.

Shi, Y., & Eberhart, R. C. (1999, July). Empirical study of particle swarm optimization. In *Proceedings of the 1999 IEEE Congress on Evolutionary Computation* (pp. 1945–1950). Washington, DC, USA.