Journal of Network Intelligence ©2025 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **10**, Number **2**, May **2025**

Support Vector Machine Method Based on Improved Grey Wolf Optimization Algorithm

Xian-Ning Lin*, Shou-Zhen Peng
School of Information Technology, Guangdong Technology College
Zhaoqing 526100, P. R. China
linxianning1202@126.com, 31839930@qq.com

Zhen-Jie Jiang
School of Information Technology and Engineering, St. Paul University Philippines
Tuguegarao 3500, Philippines
1195617459@qq.com

*Corresponding author: Xian-Ning Lin

Received May 15, 2024; revised September 2, 2024; accepted December 16, 2024.

Abstract. *Traditional Least Squares Support Vector Machines (LSSVMs) are widely used within the realms of computational intelligence, specifically the sub-disciplines of pattern recognition and machine learning. However, the performance of LSSVM algorithms relies heavily on parameter selection and faces the challenge of high computational complexity when dealing with large-scale datasets. To address these issues, this work proposes a parameter optimisation strategy for LSSVM based on the Improved Grey Wolf Optimisation (IGWO) algorithm. Firstly, by introducing the Next-Alpha rank and dynamic rank adjustment mechanism, IGWO enhances the social structure of wolf packs and improves the diversity and convergence speed. Second, by combining the position update strategy with chaotic population initialisation and chaotic perturbation, IGWO further enhances the global search capability. In addition, IGWO-LSSVM effectively improves the search efficiency and adaptability to complex optimisation problems through adaptive neighbourhood search and adaptive grey wolf wandering mechanism. Finally, IGWO is applied to LSSVM parameter optimisation. Experimental results show that the classification accuracy and computational efficiency of IGWO-LSSVM on multiple datasets are better than existing methods, especially when dealing with high-dimensional datasets. The IGWO-LSSVM algorithm in this study provides a new parameter optimisation strategy in the field of machine learning, which has important theoretical and practical application value.*

Keywords: GWO; LSSVM; Parameter optimisation; Machine learning; Global search; Adaptive strategies

1. Introduction.

The research background of Support Vector Machine (SVM) is rooted in pattern recognition theory in computer science and statistics [1, 2, 3]. SVM was originally proposed by Vapnik and Chervonenkis in 1963, and further developed by Vapnik and his colleagues in 1990s. It was further developed in the 1990s by Vapnik and colleagues, especially the kernel trick proposed in 1995, which allows SVM to be applied to nonlinear problems. The basic principle of SVM is to discriminate between different classes of data by finding the optimal hyperplane, and the key to SVM lies in the notion of support vectors, which relies on the samples located on the decision boundary to determine the model [4, 5]. SVM has been paid attention to because of its efficient performance in small sample learning, nonlinear problems and high-dimensional data processing, especially in bioinformatics [6], image recognition [7], text classification [8] and financial market analysis [9] and other fields. With the rise of big data and machine learning technologies, the research and applications of SVM continue to expand, including the improvement of algorithms, the enhancement of computational efficiency, and the development of new kernel functions, among other directions.

Machine learning techniques play a crucial role in today's data analysis and pattern recognition fields. Among them, Least Squares Support Vector Machine (LSSVM), as an effective regression analysis method, has garnered significant interest because of its capabilities to minimise the error between the measured and predicted values when solving regression problems [10]. LSSVM is an improved version of the standard SVM, which significantly reduces the algorithmic complexity. Compared with the traditional SVM, LSSVM is not only computationally faster, but also has better generalisation capabilities, making it a very powerful tool in the field of machine learning.

However, although LSSVM performs well in many applications, its performance is still affected by the choice of parameters. The choice of regularisation and kernel function parameters has a direct impact on the error and generalisation ability of the model. To further improve the performance of LSSVM, researchers have begun to explore the use of optimisation algorithms to automatically tune these parameters [11]. Various metaheuristic optimisation algorithms, due to their simplicity, robustness and fast convergence properties, have been applied in the optimisation of LSSVM parameters. The research objective of this paper is to propose a new parameter optimisation strategy by improving the grey wolf optimisation algorithm in order to improve the performance of Least Squares Support Vector Machines (LSSVM) in machine learning tasks. Meanwhile, this paper aims to establish the validity of the newly devised algorithm on different datasets and compare it with existing methods to demonstrate its advantages in terms of classification accuracy and computational efficiency.

1.1. **Related work.** The current research status of SVM is reflected in several aspects: (1) the improvement of SVM algorithms has been a hot research topic, including the choice of kernel function, parameter selection and computational efficiency; (2) SVM has been effectively utilized across a spectrum of practical issues, such as image recognition, text categorisation, bioinformatics, and financial risk assessment. Despite the success of SVM in many fields, the computational complexity and sensitivity to kernel functions and parameters when facing large-scale datasets are still challenges that need to be addressed in current research.

LSSVM, as a variant of SVM, focuses on solving regression problems by minimising a quadratic loss function, and its application in regression analysis and pattern recognition is becoming more and more widespread. Currently, research on LSSVM focuses on improving the generalisation ability, optimising the computational process, and developing more efficient strategies for model selection and parameter tuning. In the medical field, Comak et al. [12] introduced the application of LSSVM in initial patient diagnosis, where a fuzzy weighted preprocessing technique was employed to enhance the precision of the machine learning framework. This highlights the potential of Least Squares Support Vector Machines for accurate and early diagnosis in healthcare. Yu et al. [13] proposed an LSSVM based classification algorithm for improving the accuracy of image recognition. Despite the success of this method in image classification tasks, the processing efficiency for large-scale image datasets still needs to be improved. Danenas and Garsva [14] analysed the application of LSSVM in financial risk assessment. The results show that LSSVM can provide accurate risk prediction, but the performance in real-time data processing needs to be further optimised. Kang et al. [15] discuss a wind speed prediction model that integrates the ensemble empirical mode decomposition (EEMD) and LSSVM. The EEMD-LSSVM model has a higher precision in forecasting outcomes of wind power generation, showing that the integration of the LSSVM with other environmental forecasting techniques is effective. In addressing the challenges associated with dispersed data sources and large datasets, Youssef [16] introduced the Global-Local Least Squares Support Vector Machine (GLocal-LSSVM) algorithm, which provides computational efficiency while maintaining a high classification performance, making it a valuable tool for real-world applications across a wide variety of domains.

The combination of various optimisation algorithms with LSSVM has been a recurring theme in related research. Meta-heuristic optimisation algorithms have shown excellent performance on LSSVM parameter optimisation problems, providing new research directions and application potentials in the field of machine learning. For example, Gorjaei et al. [17] discussed a PSO-LSSVM-based wellhead fluid rate prediction method, which showed good self-learning ability and high prediction accuracy even with small sample size. Gong et al. [18] introduced a technique for forecasting the intervals of landslide movement combining a dual output LSSVM (DO-LSSVM), demonstrating the utility of LSSVM in geotechnical engineering. Li et al. [19] proposed a Grey Wolf Optimization algorithm (GWO)-based SVM for unbalanced datasets, which exhibits better classification accuracy and robustness in dealing with unbalanced datasets compared with other classical algorithms. Li et al. [20] proposed a motor magnetic chain prediction model based on GWO-LSSVM, which modelling features high accuracy. Luo et al. [21] used semi-supervised LSSVM (SLSSVM) algorithm to predict deepwater oil exploration. Abdillah and Setiadi [22] proposed a power system stability monitoring method based on multi-output LSSVM (M-LSSVM).

1.2. Motivation and contribution.

The high computational complexity of LSSVM when facing large-scale datasets and the problem of kernel function and parameter sensitivity limit its efficiency in practical applications. Although the GWO-LSSVM algorithm improves the performance of LSSVM by automatically adjusting these parameters through GWO, the performance of the original GWO algorithm is still limited in terms of optimisation accuracy and efficiency. To address the aforementioned issues, this study introduces an improved Grey Wolf Optimization algorithm and applies it to LSSVM parameter tuning. The main innovations and contributions of this work include:

(1) An improved grey wolf optimisation algorithm (IGWO) is proposed, which enhances the versatility and robustness by introducing Next-Alpha rank and a dynamic rank adjustment mechanism. This improvement allows the algorithm to prevent entrapment in suboptimal solutions more effectively and hastens convergence. In addition, IGWO combines chaotic population initialisation, a position update strategy for chaotic perturbations and an adaptive search strategy to further improve the global search capability and stability.

(2) A new parameter optimisation strategy is proposed by applying IGWO to the parameter optimisation of LSSVM. The strategy aims to improve the performance of LSSVM in machine learning tasks by automatically adjusting the regularisation parameters and kernel function parameters of LSSVM. Experimental results show that IGWO-LSSVM outperforms existing methods in terms of classification accuracy and computational efficiency on multiple datasets, especially when dealing with high-dimensional datasets.

2. Related technical studies.

2.1. **Principle of LSSVM.** LSSVM is a regression analysis method, which is a variant of the standard SVM that solves the regression problem by minimising a quadratic loss function [23]. The fundamental concept underlying LSSVM involves the identification of a hyperplane within the feature space, with the objective of reducing the discrepancy between actual and projected outcomes to a minimum. LSSVM is an improvement of standard SVM, and its main purpose is to transform the optimization problem of SVM into a linear matrix solution problem, thus greatly reducing the complexity [24]. LSSVM is a very effective machine learning algorithm because of its faster computation speed and better generalisation ability compared with the standard SVM.

The target of LSSVM is to find a decision function $f(x)$ which is a function of the input vector x and can be expressed as.

$$f(x) = \langle w, \phi(x) \rangle + b \tag{1}$$

where $\phi(x)$ is the transformation from the original data space to the expanded feature space, $\langle \cdot, \cdot \rangle$ denotes the dot product between vectors, w is the weight vector, and b is the bias term.

The LSSVM finds the optimal w and b by minimising the following objective function:

$$J(w, b, \epsilon, \gamma) = \frac{1}{2}\|w\|^2 + \frac{1}{N}\sum_{i=1}^{N} \epsilon_i \tag{2}$$

where N is the total count of samples utilized, ϵ_i is the error term for the i-th sample, and γ is the regularisation parameter.

The LSSVM uses a quadratic loss function which measures the difference between the predicted value $f(x_i)$ and the measured value y_i.

$$\epsilon_i = \frac{1}{2}\left(y_i - f(x_i)\right)^2 \tag{3}$$

To ensure that the decision function $f(x)$ finds the optimal hyperplane, the LSSVM introduces the following constraints [25]:

$$\|\phi(x_i)\| \leq 1, \quad \forall i = 1, \ldots, N \tag{4}$$

This means that the mapping of all training samples in the feature space must be restricted to a single unit ball.

To solve the above optimisation problem, the Lagrange multiplier is usually used. The Lagrange function is first constructed:

$$L(w, b, \alpha, \epsilon) = \frac{1}{2}\|w\|^2 + \frac{1}{N}\sum_{i=1}^{N} \epsilon_i - \sum_{i=1}^{N} \alpha_i \epsilon_i \tag{5}$$

where α_i is the Lagrange multiplier.

By solving for the minimum value of L, we can derive the most effective resolution for w and b. The original problem can be transformed into a dyadic problem by applying the KKT (Karush-Kuhn-Tucker) condition to the Lagrangian function:

$$\begin{aligned} &\max_{\alpha} \frac{1}{N}\sum_{i=1}^{N} \alpha_i y_i - \frac{1}{2N}\sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \left\langle \phi\left(x_i\right), \phi\left(x_j\right) \right\rangle \\ &\text{s.t. } \sum_{i=1}^{N} \alpha_i = 0, \quad \alpha_i \geq 0, \quad \forall i = 1, \ldots, N \end{aligned} \tag{6}$$

Ultimately, the solution of the LSSVM can be expressed as:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i \left\langle \phi(x_i), \phi(x) \right\rangle + b \tag{7}$$

Note that only part of α_i is non-zero and these corresponding sample points are called support vectors [26].

2.2. Principles of GWO algorithm. GWO is a meta-heuristic optimization algorithm that simulates the hunting behaviour of grey wolves and was proposed by Mirjalili et al. in 2014. GWO is widely used in global optimization problems due to its simplicity, robustness and fast convergence properties. The GWO algorithm simulates the social behaviours and the leadership hierarchy of grey wolves' hunting, and mainly includes the following four steps [27, 28].

(1) Leadership level: In GWO algorithm, grey wolves are divided into four categories, alpha (α), beta (β), delta (δ) and omega (ω). Among them, α is the supreme ruler, responsible for decision-making and leading the whole pack; β and δ are the wolves supporting α and responsible for supervising the other wolves; and ω is the ordinary wolf.

(2) Hunting: Grey wolves determine the location of prey by sensing the location of prey. In the GWO algorithm, α, β and δ wolves infer the location of the prey through experience and guide ω wolves to move in the direction that is most promising for prey capture.

(3) Surrounding the prey: grey wolves approach the prey gradually by shrinking the encirclement. In the GWO algorithm, this process is simulated by mathematical models, and the wolves will continuously shrink the distance to the prey.

(4) Attacking prey: When the prey is close to or trapped, the grey wolf will eventually launch a fierce attack. In the GWO algorithm, ω wolves will keep approaching the target position according to the position of α, β and δ wolves.

These four steps constitute the core process of GWO algorithm. The algorithm can effectively solve various optimisation problems, has the advantages of fast convergence speed and strong robustness, and has been widely used in engineering applications. In GWO, the social structure of wolf pack is used to simulate the optimisation process. The wolf pack consists of grey wolves, including three leader wolves (α, β, δ), which are ranked based on fitness, with α being the most dominant leader. The initial positions of the wolves are randomly generated and each wolf represents a potential solution to the problem.

In each iteration, the location of the leader wolf is adjusted based on the following method:

$$D_\alpha = C \cdot X_p - X_\alpha \tag{8}$$

$$X_\alpha(t+1) = X_p - A \cdot D_\alpha \tag{9}$$

$$X_\beta(t+1) = X_p - A \cdot D_\beta \tag{10}$$

$$X_\delta(t+1) = X_p - A \cdot D_\delta \tag{11}$$

where D_α is the distance between the prey and the α leader wolf, X_p is the current position of the prey, X_α is the current position of the α leader wolf, and A and C are vectors of coefficients determined by random numbers.

To increase the randomness and diversity, GWO introduces random numbers to adjust the search direction and step size of the wolves.

$$A = 2a \cdot r_1 - a \tag{12}$$

$$C = 2 \cdot r_2 \tag{13}$$

The acceleration factor a is calculated as shown below [29]:

$$a = 2 - 2 \cdot \frac{t}{T} \tag{14}$$

where t is the number of current iterations and T is the total number of iterations.

After each iteration, the fitness of all wolves is evaluated and the wolves are sorted according to the fitness to determine the new lead wolf. The GWO algorithm stops when a preset number of iterations is reached or when the adaptation degree reaches a good enough result. The schematic principle of the GWO algorithm is shown in Figure 1.

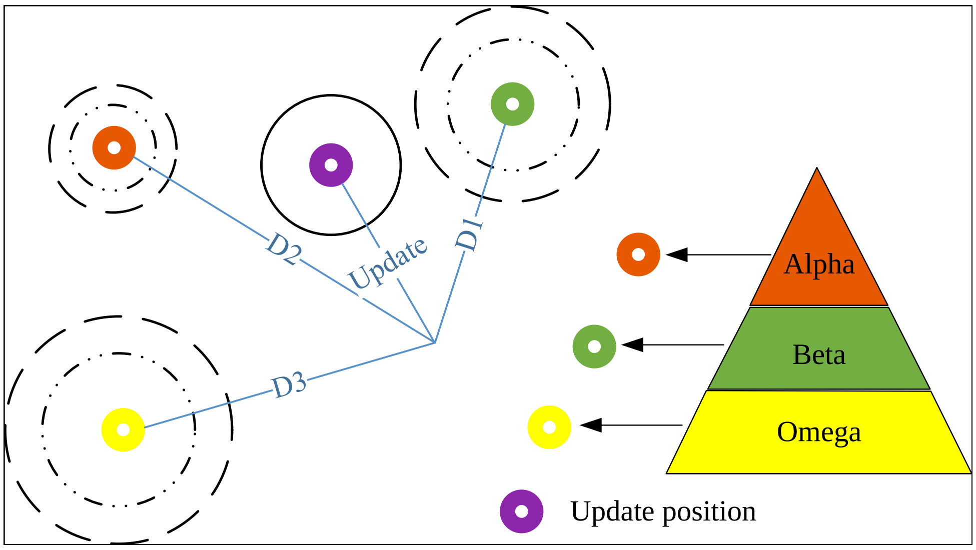

Figure 1. Schematic representation of the principle of the GWO algorithm

3. Improvement of the grey wolf optimisation algorithm.

3.1. **Improved grey wolf hierarchy.** The primitive grey GWO by fixing the rank updating mechanism, which causes the search efficiency and convergence speed to be affected. In addition, the hierarchy updating mechanism is too simple, resulting in the algorithm not being able to make full use of the population information and easily falling into local optimal solutions. With the aim of increasing the diversity and convergence of GWO, this paper improves the original grey wolf hierarchy.

Firstly, the Next-Alpha (N-α) rank is introduced. Next-Alpha (N-α) is a new rank added on top of the original four ranks. Next-Alpha wolf is the candidate of the current Alpha wolf, which has similar leadership ability and decision-making level as the Alpha wolf. Next-Alpha wolf will pay close attention to the behaviour of the Alpha wolf and learn the decision-making method of the Alpha wolf, so as to prepare for taking over the position of Alpha wolf in the future. Next-Alpha Wolf will pay close attention to Alpha Wolf's behaviour and learn Alpha Wolf's decision-making methods, so as to prepare itself to take over the position of Alpha Wolf in the future.

Second, a dynamic rank adjustment mechanism is used. During the iteration of the GWO algorithm, the rank of each wolf is not fixed, but dynamically adjusted according to the change of its adaptation value. The specific approach is as follows:

(1) At initialisation, the wolf with the highest fitness value in the population is designated as Alpha, the wolf with the second highest fitness value is designated as Next-Alpha, and the remaining wolves are classified as Beta, Delta, and Omega, in that order, according to their fitness values.

(2) In each iteration, the fitness value of each wolf is re-evaluated. If the fitness value of an Omega wolf exceeds the fitness value of the current Delta wolf, that Omega wolf will replace the current Delta wolf as the new Delta wolf. Similarly, if a Beta wolf's fitness value exceeds the current Alpha wolf's fitness value, that Beta wolf replaces the current Alpha wolf and becomes the new Alpha wolf; the Next-Alpha wolf becomes the new Alpha wolf.

(3) In order to ensure the convergence, the fitness value of the Next-Alpha wolf must be higher than the fitness value of the current Alpha wolf by a certain threshold in order to take the place of the Alpha wolf.

The diversity and robustness of the grey wolf optimisation algorithm can be enhanced by the introduction of the Next-Alpha rank and the mechanism of dynamically adjusting the rank to avoid falling into local optimal solutions. Meanwhile, the introduction of Next-Alpha wolf can also promote the convergence speed and elevate the quality of the concluding result.

3.2. **Chaotic population initialisation.** In this paper, chaos theory is introduced to enhance the population initialisation process of GWO to improve the global search capability. Chaotic mapping is a mathematical tool that can generate pseudo-random sequences, which can help the algorithm to jump out of the local optimal solution and enhance the global search capability. The commonly used chaotic mappings are Logistic mapping and Tent mapping. In this paper, Tent mapping is used and its mathematical expression is as follows:

$$X_{n+1} = \begin{cases} r \cdot X_n + (1-r) \cdot (1 - X_n), & \text{if } X_n < 0.5 \\ r \cdot (1 - X_n) + (1-r) \cdot X_n, & \text{otherwise} \end{cases} \tag{15}$$

where X_n is the value of the current iteration and r is a control parameter, usually taking a value between $(0, 1)$.

After setting the control parameter r and the offset parameter b of the chaotic mapping, for each wolf i position X_i, the chaotic mapping is used to generate the initial population position $X_i^{(0)}$.

$$X_i^{(0)} = TentMap(X_i, r) \tag{16}$$

where $TentMap$ is the Tent map.

3.3. **Grey wolf position update with introduction of chaotic perturbation.** In this paper, a chaotic perturbation is introduced to enhance the ability of the GWO algorithm to jump out of local optimal solutions. The chaotic perturbation is achieved by adding a chaotic term to the grey wolf's position update formula, which is also generated by the Tent chaotic mapping.

Assuming that X_i denotes the position of the i-th grey wolf, X_{best} denotes the best position in the current population, and the chaos term C is generated by the Tent mapping, the position update method is as follows:

$$X_i(t+1) = X_i(t) + A \cdot (X_{best} - X_i(t)) + C \tag{17}$$

where t denotes the current number of iterations; A denotes a positive control parameter used to adjust the step size of the grey wolf moving towards the optimal position; C denotes a chaotic perturbation term; used to increase the randomness; and X_{best} denotes the best position in the current population.

$$C = c_{\max} - (c_{\max} - c_{\min})\, e^{(-\lambda)} \lambda, \tag{18}$$

where $c_{\min}$ is the minimum of the chaotic perturbation term; $c_{\max}$ is the maximum of the chaotic perturbation term; and λ is the tuning parameter.

3.4. **Adaptive neighbourhood search.** In traditional GWO algorithms, wolves search for prey through territory search, a process that can be analogous to the search process of an optimisation problem. In IGWO, we introduce the adaptive neighbourhood search mechanism to improve the local search accuracy.

An adaptive neighbourhood is a search range that is dynamically adjusted at each iteration based on the fitness and position of the grey wolf. This neighbourhood can be dynamically expanded or contracted to fit the current search state as required by the search process. First, initialise the wolf positions $X = \{x_1, x_2, \ldots, x_n\}$ and the corresponding fitness $F = \{f_1, f_2, \ldots, f_n\}$.

Then, the neighbourhood range is calculated. For each grey wolf i, the neighbourhood range d_i is dynamically computed based on its fitness f_i and historical best position x_i^{best}.

$$d_i = d_i^{\max} - \frac{d_i^{\max} - d_i^{\min}}{2} \tanh\left(\frac{f_i - f_{\text{worst}}}{f_{\text{best}} - f_{\text{worst}}}\right) \tag{19}$$

where d_i is the dynamic range of the neighbourhood of the i-th grey wolf; $d_i^{\max}$ and $d_i^{\min}$ are the maximum and minimum ranges of the neighbourhood of the i-th grey wolf; f_{best} and f_{worst} are the best and worst fitnesses in the current wolf pack.

A neighbourhood search is performed within d_i to update the position of the grey wolf x_i. The search step for the grey wolf can be expressed as follows:

$$x_i^{t+1} = x_i^{\text{best}} + \alpha d_i \cdot \left(x_i^t - x_i^{\text{best}}\right) \tag{20}$$

where α is a random number in the range [0,1] and t denotes the current iteration number.

Calculate the fitness of the new position f_i^{new}. If f_i^{new} is better than f_i^{best}, update x_i^{best}, otherwise continue to compute the range of the computed neighbourhood until the stopping condition is met.

3.5. **Adaptive grey wolf wandering.** Adaptive grey wolf wandering is a method for dynamically adjusting a grey wolf's search strategy, which allows the grey wolf to change its search behaviour based on the current search state and historical information. The adaptive wandering proposed in this paper aims to resolve the challenge of premature convergence that may occur in traditional grey wolf optimization algorithms by dynamically adjusting the search strategy to enhance the algorithm's ability to search for the globally optimal solution.

(1) Dynamic adjustment of wandering step length.

The wandering step size S_i is dynamically adjusted according to the current fitness f_i and the historical best fitness f_i^{best} of the grey wolf.

$$S_i = S_i^{\text{init}} \cdot e^{-q \cdot (f_i - f_i^{\text{best}})} \tag{21}$$

where S_i^{init} is the initial wandering step size, and q is a positive moderator controlling the effect of adaptation differences on step size.

(2) Dynamic adjustment of travelling direction.

Direction of wandering D_i Dynamically adjusted according to the distance of the grey wolf from the optimal position of the pack.

$$D_i = D_i^{\text{init}} \cdot \tanh\left(\frac{\|x_i - x_{\text{best}}\|}{d_{\max}}\right) \tag{22}$$

where D_i^{init} is the initial wandering direction, x_{best} is the best position in the pack, and $d_{\max}$ is the maximum distance in the pack.

(3) Update grey wolf locations.

Update the position of the grey wolf in relation to the length and direction of the wandering step.

$$x_i^{t+1} = x_i^t + S_i \cdot D_i \tag{23}$$

The adaptive grey wolf wandering mechanism allows the algorithm to dynamically adjust its search strategy based on real-time information during the search process, which not only improves the algorithm's search efficiency, but also enhances its ability to adapt to complex optimisation problems.

4. **Optimisation of LSSVM parameters based on IGWO.**

4.1. **Definition of parametric optimisation problem.**

Parameter selection plays a crucial role in determining the effectiveness of LSSVM, including the regularisation parameters and the kernel function parameters. The goal of using IGWO for LSSVM parameter optimisation in this paper is to find a set of parameters that minimise the model's error on the training data while maintaining good generalisation.

Let θ denote the parameter vector of the LSSVM, including the regularisation parameter C and the kernel function parameter σ. The parameter optimisation problem can be defined as follow:

$$\theta^* = \arg\min_{\theta}[E(\theta)] \tag{24}$$

where $E(\theta)$ is the error function of LSSVM under the parameter θ, which is usually a combination of the training error and the regularity term.

4.2. **Steps in parameter optimisation.**

Step 1: Generate an initial grey wolf population $X = \{x_1, x_2, \ldots, x_n\}$, with each grey wolf representing a set of parameters of the LSSVM. Encode the parameters of the LSSVM θ as the position vector x_i of the grey wolves.

Step 2: For each grey wolf, train the model using LSSVM and calculate its error on the validation set as the fitness. The fitness function can be defined as the inverse of the error of the LSSVM, with maximising the error as the optimisation objective.

$$F(\theta) = \frac{1}{E(\theta)} \tag{25}$$

Step 3: Based on the fitness, the population is updated using IGWO's strategies including chaotic perturbation, adaptive neighbourhood search and adaptive grey wolf wandering.

Step 4: Select the best adapted grey wolf from the updated population and use its parameter θ as the optimization parameter of LSSVM.

Step 5: Repeat Step 2-4 until the iteration stop condition is satisfied.

5. Experimental results and comparisons.

5.1. Experimental environment and experimental data.

The computer used for the experiments was Windows 10 (64-bit), with Matlab R2020b software, CPU model 13700KF, 16 G RAM, and graphics card GeForce RTX3070. The subsequent settings were applied to the parameters: the population size was 200, the maximum number of iterations was 100, and the tuning parameter of the chaotic perturbation term λ is 0.1, $c_{\min}$ is 0.1, $c_{\max}$ is 10, the control parameter r is 0.2, the neighbourhood search random number α is 0.5, and the wandering positive adjustment factor q is 0.2.

The data used in the experimental section are four widely used machine learning datasets, Bitcoin-Heist, Covertype, Census-Income, and GitHub MUSAE, which are shown in Table 1. The Bitcoin-Heist dataset is usually related to Bitcoin transactions and user behaviour analysis, and may contain information such as transaction amounts, timestamps, user addresses, etc. It is used to study cryptocurrency circulation patterns or anomaly detection. that are used to study cryptocurrency circulation patterns or anomaly detection. The Covertype dataset contains data on the type of forest cover and is often used for classification problems such as predicting the type of forest cover in a particular area, and it contains data on both geographic and biological features. The Census-Income dataset is based on income data from the U.S. Census and is often used for supervised learning tasks such as forecasting if a person's yearly earnings exceeds 50,000, this dataset contains demographic information and economic indicators. GitHub MUSAE is a dataset consisting of GitHub projects for software engineering research that may include project metadata, code commit records, issue tracking, and user interaction data, and is suitable for project health, defect prediction, and contributor behaviour analysis.

Table 1. Experimental datasets information

Data set	**Data length**	**Number of features**	**Number of categories**
BitcoinHeist	2916697	10	2
Covertype	581012	54	2
Census-Income	299285	40	2
GitHub MUSAE	37700	4906	2

5.2. Feasibility Analysis of IGWO..

In order to verify the feasibility of the IGWO algorithm, this paper performs simulation comparison experiments for IGWO, GWO, PSO (Particle Swarm Optimisation) and GA (Genetic Algorithm) using five classical benchmark test functions. The convergence of the test functions is used to evaluate the parameter optimisation ability of the IGWO algorithm. The test functions include continuous single-peak function and nonlinear multi-peak function, in which the single-peak functions f_1 and f_2 are used to test the accuracy and convergence speed, and the multi-peak functions f_3, f_4, and f_5 are utilized to evaluate the algorithm's capacity for searching and its proficiency in escaping local optima, and the test functions are shown in Table 2.

The four optimisation algorithms are run independently for 50 times on five test functions, and the optimal value, mean and standard deviations on each function are calculated, where the mean reflects the maximum accuracys and the standard deviation reflects the stabilitys. Table 3 shows the results of the four optimisation algorithms on the test functions.

Table 2. Test function

Function name	Function expression	Search scope
Sphere	$f_1(x)=\sum_{i=1}^{D}x_i^2$	$[-5.12, 5.12]$
Rosenbrock	$f_2(x)=\sum_{i=1}^{D-1}\left[100\left(x_{i+1}-x_i^2\right)^2+(x_i-1)^2\right]$	$[-30, 30]$
Rastrigin	$f_3(x)=\sum_{i=1}^{D}\left(x_i^2-10\cos(2\pi x_i)+10\right)$	$[-5.12, 5.12]$
Griewank	$f_4(x)=\frac{1}{4000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	$[-600, 600]$
Ackley	$f_5(x)=-20\exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{D}x_i^2}\right)-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi x_i\right)+20+e$	$[-2, 32]$

Table 3. Optimisation results analysis of test function

Function	Arithmetic	Optimum value	Average value	Standard deviation
f_1	GA	5.99E-04	2.17E-02	3.78E-02
	PSO	1.64E-56	4.51E-45	1.40E-44
	GWO	1.03E-61	2.15E-54	5.39E-52
	IGWO	4.30E-67	7.66E-63	1.75E-60
f_2	GA	1.36E+01	4.76E+02	5.58E-01
	PSO	2.57E+00	2.46E+00	3.16E-01
	GWO	1.76E+00	1.12E+00	2.43E-02
	IGWO	0.00E+00	1.64E-23	1.60E-22
f_3	GA	3.31E+00	8.89E+00	2.63E+00
	PSO	2.02E+00	5.52E+00	3.88E+00
	GWO	2.65E-01	3.75E-01	8.30E-02
	IGWO	7.54E-11	2.56E-09	1.31E-08
f_4	GA	1.50E-01	2.74E-01	1.91E-01
	PSO	2.27E-02	1.76E-01	4.27E-02
	GWO	6.80E-03	9.84E-03	3.27E-04
	IGWO	1.14E-09	5.96E-07	4.88E-07
f_5	GA	2.73E-15	6.14E-01	5.50E-01
	PSO	4.86E-21	3.16E-21	0.00E+00
	GWO	2.32E-28	5.49E-24	0.00E+00
	IGWO	2.85E-37	2.85E-37	0.00E+00

For the f_1 function, IGWO shows the best optimal and average values, indicating that it has good global search capability and stability on this function. For the f_2 function, IGWO also shows the best performance with an optimal value of 0, which is the global minimum for the tested function, indicating that IGWO has a very high accuracy on this function. For the f_3 function, IGWO outperforms the other algorithms in terms of both optimal and mean values, with a very small standard deviation, showing its excellent performance and stability. For the f_4 function, the optimal and average values of IGWO are also better than other algorithms with the smallest standard deviation, which further proves its stability and superiority. For the f_5 function, IGWO's optimal and average values are the same as PSO's both being the global minimum of the tested functions with a standard deviation of 0, showing IGWO's extremely high performance on this function.

It can be concluded that IGWO shows excellent performance on all the tested functions, especially on the f_2 and f_5 functions, IGWO reaches the global minimum, which shows its strong global search capability and stability. In addition, the mean and standard deviation of IGWO on all the tested functions are better than or at least comparable to other algorithms, which further proves the feasibility and effectiveness of IGWO on the parameter search problem. Therefore, IGWO can be considered as a powerful optimisation algorithm, especially for complex optimisation problems.

5.3. **Comparative analysis of classification accuracy.** In order to evaluate the classification performance of the IGWO-LSSVM, comparative experiments are conducted with GWO-LSSVM [20], SLSSVM [21] and M-LSSVM [22] on the Bitcoin-Heist, Covertype, Census-Income and GitHub MUSAE datasets. The classification accuracy F-measure of each algorithm is compared in the experiments respectively and the experimental results are shown in Figure 2.

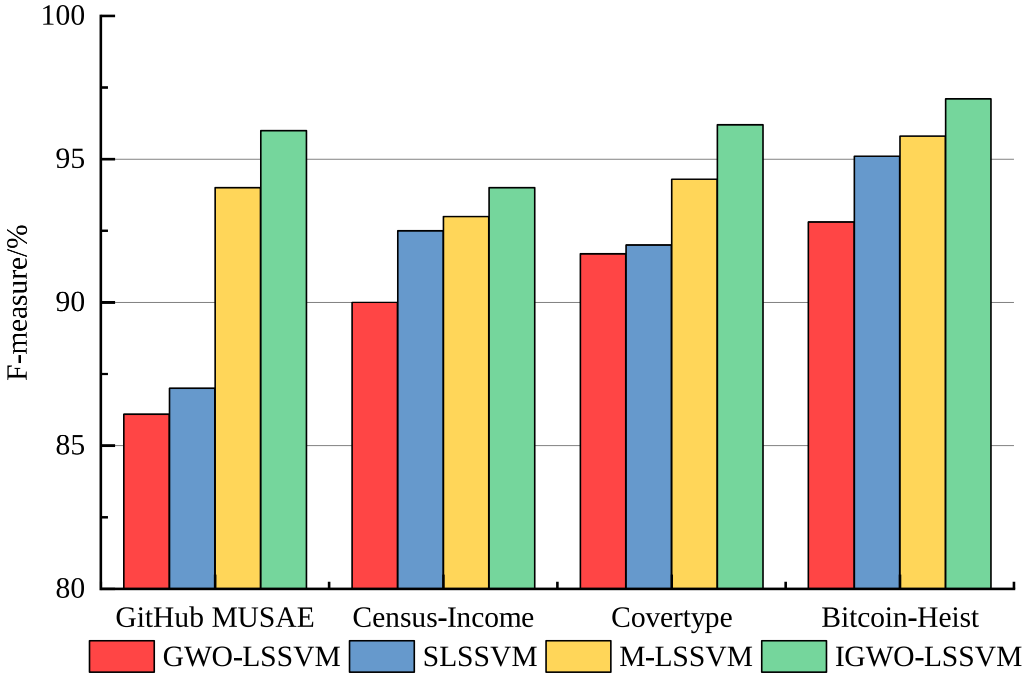

Figure 2. F-measure of four algorithms on different datasets

It can be seen that the classification accuracy of IGWO-LSSVM is always higher than that of the other three on each dataset. On various datasets with different features, IGWO-LSSVM has better robustness. On the Bitcoin-Heist dataset, the accuracy of IGWO-LSSVM is 8.7%, 7.5%, and 2.6% higher than that of GWO-LSSVM, SLSSVM, and M-LSSVM, respectively; on the Covertype dataset, the accuracy of IGWO-LSSVM is 5% higher than that of GWO-LSSVM, SLSSVM, and M-LSSVM, respectively. LSSVM by 5.1%, 4.9%, and 2.3%, respectively; on the Census-Income dataset, the accuracy of IGWO-LSSVM is 4.5%, 2.8%, and 2.4% higher than that of GWO-LSSVM, SLSSVM, and M-LSSVM, respectively; on the GitHub MUSAE dataset, the IGWO-LSSVM's accuracy is 8.5%, 7.4%, and 2.9% higher than GWO-LSSVM, SLSSVM, and M-LSSVM, respectively. From the experimental results, it can be seen that IGWO-LSSVM does not have a significant increase in accuracy when dealing with the lower dimensionality BitcoinHeist, Covertype, and Census-Income datasets, whereas it is much more accurate than the other three when dealing with the higher dimensionality GitHub MUSAE dataset. This is due to the fact that IGWO-LSSVM introduces chaotic perturbations to enhance the ability to jump out of the local optimal solution, and also uses IGWO for parameter optimisation. Thus, the classification accuracy is effectively improved. Therefore, from the above experimental results, it can be seen that the accuracy of IGWO-LSSVM is higher than that of GWO-LSSVM, SLSSVM and M-LSSVM when facing most datasets. When dealing with high-dimensional data, the advantage of IGWO-LSSVM is even more obvious.

5.4. **Comparative analysis of classification efficiency.** The running times of the four algorithms on different datasets are shown in Figure 3. It can be seen that compared with GWO-LSSVM, the running time of IGWO-LSSVM is reduced by 37%, and the computational efficiency is obviously improved. This is because the introduction of Next-Alpha wolf can also promote the convergence speed. However, compared to SLSSVM and M-LSSVM, the running time of IGWO-LSSVM is still high, which is an issue that this study plans to continue to address.

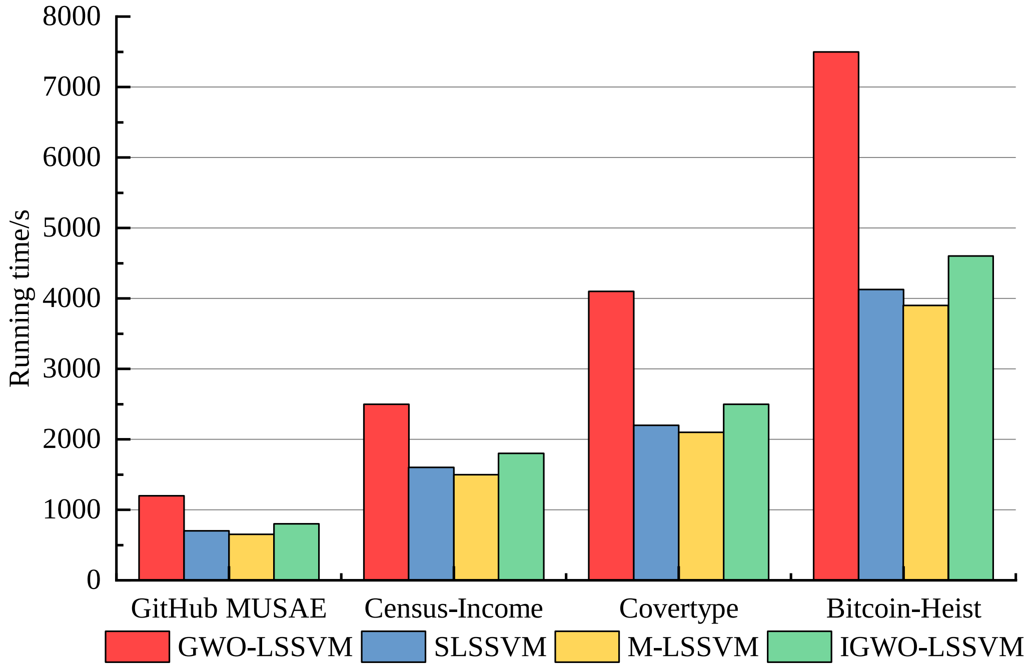

Figure 3. Running time of the four algorithms on different datasets

6. **Conclusions.**

In this study, an IGWO-based LSSVM parameter optimisation strategy is proposed. Experimental results on four widely used machine learning datasets show that IGWO-LSSVM outperforms GWO-LSSVM, SLSSVM, and M-LSSVM by 8.7%, 7.5%, and 2.6%, respectively, in terms of classification accuracy, 5.1%, 4.9%, and 2.3%, respectively, on the Covertype dataset, 4.5%, 2.8%, and 2.4%, respectively, on the Census-Income dataset by 4.5%, 2.8%, and 2.4%, and on the GitHub MUSAE dataset by 8.5%, 7.4%, and 2.9%, respectively. In terms of computational efficiency, IGWO-LSSVM reduces the running time by 37% compared to GWO-LSSVM, although there is still room for improvement compared to SLSSVM and M-LSSVM. These results demonstrate the advantages of IGWO-LSSVM in terms of parameter optimisation, global search capability, and computational efficiency, especially when dealing with high-dimensional datasets. Parameter tuning and the introduction of chaotic perturbations in the IGWO-LSSVM algorithm may increase the complexity of the model, which may have an impact on fast decision-making in real-time applications. Future work needs to focus on the optimisation to improve its scalability on larger datasets and to reduce the consumption of computational resources so as to better accommodate the demands of real-time or near real-time machine learning tasks.

Acknowledgment. This work is supported by the Department of Education of Guangdong Province “Exploration and Practice of Software Engineering Major Construction in Private Universities under the Background of New Engineering” (No. 2021-644).

REFERENCES

[1] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, 2020.

[2] M. Tanveer, T. Rajani, R. Rastogi, Y.-H. Shao, and M. Ganaie, "Comprehensive review on twin support vector machines," *Annals of Operations Research*, pp. 1–46, 2022.

[3] A. Kurani, P. Doshi, A. Vakharia, and M. Shah, "A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting," *Annals of Data Science*, vol. 10, no. 1, pp. 183–208, 2023.

[4] M. Shao, X. Wang, Z. Bu, X. Chen, and Y. Wang, "Prediction of energy consumption in hotel buildings via support vector machines," *Sustainable Cities and Society*, vol. 57, 102128, 2020.

[5] C. Avcı, M. Budak, N. Yağmur, and F. Balçık, "Comparison between random forest and support vector machine algorithms for LULC classification," *International Journal of Engineering and Geosciences*, vol. 8, no. 1, pp. 1–10, 2023.

[6] O. Okwuashi, and C. E. Ndehedehe, "Deep support vector machine for hyperspectral image classification," *Pattern Recognition*, vol. 103, 107298, 2020.

[7] F. Jiang, Y. Lu, Y. Chen, D. Cai, and G. Li, "Image recognition of four rice leaf diseases based on deep learning and support vector machine," *Computers and Electronics in Agriculture*, vol. 179, 105824, 2020.

[8] E. Y. Boateng, J. Otoo, and D. A. Abaye, "Basic tenets of classification algorithms K-nearest-neighbor, support vector machine, random forest and neural network: a review," *Journal of Data Analysis and Information Processing*, vol. 8, no. 4, pp. 341–357, 2020.

[9] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. K. Behera, "Deep feature based rice leaf disease identification using support vector machine," *Computers and Electronics in Agriculture*, vol. 175, 105527, 2020.

[10] F. Zeng, M. Nait Amar, A. S. Mohammed, M. R. Motahari, and M. Hasanipanah, "Improving the performance of LSSVM model in predicting the safety factor for circular failure slope through optimization algorithms," *Engineering with Computers*, vol. 15, pp. 1–12, 2021.

[11] J. Wang, W. Shao, and J. Kim, "Combining MF-DFA and LSSVM for retina images classification," *Biomedical Signal Processing and Control*, vol. 60, 101943, 2020.

[12] E. Comak, K. Polat, S. Güneş, and A. Arslan, "A new medical decision making system: least square support vector machine (LSSVM) with fuzzy weighting pre-processing," *Expert Systems with Applications*, vol. 32, no. 2, pp. 409–414, 2007.

[13] X. Yu, K. Liu, D. Wu, and Y. He, "Raisin quality classification using least squares support vector machine (LSSVM) based on combined color and texture features," *Food and Bioprocess Technology*, vol. 5, pp. 1552–1563, 2012.

[14] P. Danenas, and G. Garsva, "Selection of support vector machines based classifiers for credit risk domain," *Expert Systems with Applications*, vol. 42, no. 6, pp. 3194–3204, 2015.

[15] A. Kang, Q. Tan, X. Yuan, X. Lei, and Y. Yuan, "Short-term wind speed prediction using EEMD-LSSVM model," *Advances in Meteorology*, vol. 2017, pp. 14–26, 2017.

[16] A. Youssef Ali Amer, "Global-local least-squares support vector machine (GLocal-LS-SVM)," *PLOS One*, vol. 18, no. 4, e0285131, 2023.

[17] R. G. Gorjaei, R. Songolzadeh, M. Torkaman, M. Safari, and G. Zargar, "A novel PSO-LSSVM model for predicting liquid rate of two phase flow through wellhead chokes," *Journal of Natural Gas Science and Engineering*, vol. 24, pp. 228–237, 2015.

[18] W. Gong, S. Tian, L. Wang, Z. Li, H. Tang, T. Li, and L. Zhang, "Interval prediction of landslide displacement with dual-output least squares support vector machine and particle swarm optimization algorithms," *Acta Geotechnica*, vol. 17, no. 9, pp. 4013–4031, 2022.

[19] B. Li, C. Luo, and Z. Wang, "Application of GWO-SVM algorithm in arc detection of pantograph," *IEEE Access*, vol. 8, pp. 173865–173873, 2020.

[20] K. Li, G. Cheng, X. Sun, and Z. Yang, "A nonlinear flux linkage model for bearingless induction motor based on GWO-LSSVM," *IEEE Access*, vol. 7, pp. 36558–36567, 2019.

[21] W.-P. Luo, H.-Q. Li, and N. Shi, "Semi-supervised least squares support vector machine algorithm: application to offshore oil reservoir," *Applied Geophysics*, vol. 13, pp. 406–415, 2016.

[22] M. Abdillah, and H. Setiadi, "Advanced wide-area monitoring system design for electrical power system," *International Review on Modelling and Simulations*, vol. 13, no. 6, pp. 362–372, 2020.

[23] T.-Y. Wu, H. Li, S. Kumari, and C.-M. Chen, “A Spectral Convolutional Neural Network Model Based on Adaptive Fick’s Law for Hyperspectral Image Classification,” *Computers, Materials & Continua*, vol. 79, no. 1, pp. 19–46, 2024.

[24] T.-Y. Wu, A. Shao, and J.-S. Pan, “CTOA: Toward a Chaotic-Based Tumbleweed Optimization Algorithm,” *Mathematics*, vol. 11, no. 10, p. 2339, 2023.

[25] T.-Y. Wu, H. Li, and S.-C. Chu, “CPPE: An Improved Phasmatodea Population Evolution Algorithm with Chaotic Maps,” *Mathematics*, vol. 11, no. 9, p. 1977, 2023.

[26] R. Mall, and J. A. Suykens, “Very sparse LSSVM reductions for large-scale data,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 1086–1097, 2015.

[27] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[28] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, “Grey wolf optimizer: a review of recent variants and applications,” *Neural Computing and Applications*, vol. 30, pp. 413–435, 2018.

[29] M. H. Nadimi-Shahraki, S. Taghian, and S. Mirjalili, “An improved grey wolf optimizer for solving engineering problems,” *Expert Systems with Applications*, vol. 166, pp. 113917–113932, 2021.