EAI Endorsed Transactions
on Energy Web
Research Article EAI.EU

Research on short-term photovoltaic power prediction based on LMD-IPSO-LSVM

Yanjuan Ma[1,*], Mai Jiang[1,2] and Limin Zhang[1]

[1] Criminal Investigation and Counter-Terrorism College, Criminal Investigation Police University of China, Shenyang 110854, China.

[2] Key Laboratory of Trace Inspection and Identification Technology, Ministry of Public Security, Shenyang, China

Abstract

With the rapid development of human society, resource shortages and environmental degradation have become increasingly pressing issues. To address these challenges, solar energy has garnered significant attention due to its high efficiency, safety, and pollution-free nature. This paper proposes a novel short-term photovoltaic power prediction framework based on an integrated LMD-IPSO-LSVM approach. The model's key innovation lies in its hierarchical decomposition-optimization architecture: First, Local Mean Decomposition (LMD) addresses the non-stationary and nonlinear characteristics of PV power data by decomposing original signals into physically meaningful Product Functions (PFs). Second, an Improved Particle Swarm Optimization (IPSO) algorithm featuring an adaptive inertia weight mechanism is developed to optimize LSVM hyperparameters for each PF component. This strategic integration enables the model to simultaneously capture complex temporal patterns while maintaining superior generalization capability. Experimental validation demonstrates that our IPSO achieves significantly faster convergence (46.3% improvement in convergence speed) and enhanced optimization precision compared to standard PSO, providing a solid foundation for accurate power forecasting. In order to evaluate the proposed methodology, comparative models including standalone LSVM and PSO-LSVM are also established and tested on the same dataset. Experimental results demonstrate that the proposed hybrid model called LMD-IPSO-LSVM achieves high prediction accuracy and better performance compared with other algorithms.

Keywords: Photovoltaic power prediction; Local Mean Decomposition; Least Squares Support Vector Machine; Particle Swarm Optimization

Received on 15 October, accepted on 14 November 2025, published on 09 February 2026

doi: 10.4108/ew.11823

1. Introduction

With the continuous advancement of society, electricity consumption has risen significantly. In light of both green environmental protection and economic development needs, the adoption of green energy has become a prevailing trend. Among various renewable sources, solar energy stands out for being inexhaustible and readily available. For example, employing solar energy systems enables the operation of police facilities and specialized equipment with a renewable power source, which makes photovoltaic (PV) power generation forecasting a focus of grid management. However, its inherent intermittency and instability pose challenges to grid security and operation. Therefore, improving the accuracy of PV power prediction is of significant importance for the safe and stable operation of power systems[1].

At present, numerous studies have been conducted on PV power forecasting, which can be broadly categorized into direct prediction methods[2] and indirect prediction methods[3]. Indirect prediction relies on physical models, such as all-sky imaging and satellite-based weather prediction. Under ideal weather conditions, physical models can yield satisfactory results; however, their application in

*Corresponding ahthor. Email: myj_2016@163.com

real-world scenarios tends to be complex. In contrast, direct prediction methods utilize historical data combined with artificial intelligence algorithms to build neural network models for forecasting. Due to their lower cost and higher practicality, these approaches are more widely adopted in the field. Common techniques include time series analysis[4], BP neural networks[5], support vector machines (SVM)[6], and so on.

Although the quality of historical data is a critical determinant of PV power forecasting accuracy, data preprocessing can effectively mitigate the impact of outliers. This, in turn, allows machine learning models to more precisely analyze influencing factors and improve prediction performance. In recent years, common outlier detection methods include the optimal variance algorithm[7], the interquartile range (IQR) [8], and Isolation Forest (iForest)[9]. While the optimal variance algorithm can identify abnormal operation data of wind turbines, its detection efficiency is relatively low. The IQR method involves statistical analysis of historical data and can effectively detect anomalies, but the process is computationally intensive. Among various anomaly detection algorithms, iForest stands out for its simplicity, high detection efficiency, and accuracy, making it widely applicable.

Data clustering and analysis methods include the grey correlation coefficient method, K-means algorithm[10], fuzzy C-means clustering[11], among others. The grey correlation method tends to be subjective and struggles to objectively reflect the influence of different weather factors on PV output. The K-means algorithm relies on random initial selection of K values, which may lead to inaccurate data partitioning. In contrast, fuzzy C-means clustering optimizes an objective function to automatically classify data, offering more adaptive performance[12].

Traditional PV prediction methods often focus on optimizing algorithms without adequately considering the varying importance of different samples during model training, leading to limited accuracy. Moreover, the typical time interval for PV power prediction is one hour, which falls short of meeting higher grid scheduling requirements. To address these issues, this paper proposes a weighted support vector machine model based on the fuzzy C-means learning algorithm, designed to fully account for the similarity between historical and forecast power data[13].

2. Basic theory

2.1. Local Mean Decomposition

Basic principles

For $x(t)$, the specific decomposition process is as follows:

1) Select each local extreme point n_i in the original signal $x(t)$, and use n_i and n_{i+1} to calculate the average value of adjacent local extreme points m_i.

$$m_i = \frac{n_i + n_{i+1}}{2} \tag{1}$$

The corresponding time of each extreme point is t_{ni} and t_{ni+1}. The obtained m_i is connected and extended between the corresponding time of each extreme point. The extended straight line is processed by the moving average method to obtain the local mean function $m_{11}(t)$

2) Calculate the local amplitude a_i according to the absolute value of the difference between adjacent extreme points n_i and n_{i+1}:

$$a_i = \frac{|n_i - n_{i+1}|}{2} \tag{2}$$

The moving average method is used to process the straight line a_i extending between the corresponding times t_{ni} and t_{ni+1} of each extreme point, and the local mean function $a_{11}(t)$ can be obtained.

3) Separate the local mean function $m_{11}(t)$ from the original signal $x(t)$, and demodulate the separated function with $a_{11}(t)$:

$$h_{11}(t) = x(t) - m_{11}(t) \tag{3}$$

$$s_{11}(t) = \frac{h_{11}(t)}{a_{11}(t)} \tag{4}$$

Since the decomposed pf component is the product of the envelope function and pure frequency modulation function, it is necessary to judge whether $s_{11}(t)$ is pure frequency modulation function. Through the above steps, the envelope function $a_{12}(t)$ of $s_{11}(t)$ is obtained and judged according to the following formula:

$$\begin{cases} -1 \le s_{1n}(t) \le 1 \\ a_{1(n+1)}(t) = 1 \end{cases} \tag{5}$$

If Eq. (5) is satisfied, it is a pure FM function. If not, repeat the above iteration for $s_{11}(t)$ until a pure FM signal $s_{1n}(t)$ satisfying the conditions is obtained:

$$\begin{cases} h_{11}(t)=x(t)-m_{11}(t) \\ h_{12}(t)=s_{11}(t)-m_{12}(t) \\ \vdots \\ h_{1n}(t)=s_{1(n-1)}(t)-m_{1n}(t) \end{cases} \quad (6)$$

4) Multiply all envelope functions in the iterative process to obtain envelope signal $a_1(t)$:

$$a_1(t)=a_{11}(t)a_{12}(t)\cdots a_{1n}(t)=\prod_{q=1}^{n}a_{1q}(t) \quad (7)$$

5) The first pf component of the original signal $x(t)$ can be obtained by multiplying $a_1(t)$ and $s_{1n}(t)$:

$$PF_1(t)=a_1(t)s_{1n}(t) \quad (8)$$

6) Separate $PF_1(t)$ from the original signal $x(t)$ to obtain a new signal $u_1(t)$, repeat the above steps as a new original signal and cycle for k times until $u_k(t)$ is a monotonic function:

$$\begin{cases} u_1(t)=x(t)-PF_1(t) \\ u_2(t)=u_1(t)-PF_2(t) \\ \vdots \\ u_k(t)=u_{k-1}(t)-PF_k(t) \end{cases} \quad (9)$$

The original signal will be decomposed into multiple pf components and a sum of $u_k(t)$:

$$x(t)=\sum_{p=1}^{k}PF_p(t)+u_k(t) \quad (10)$$

LMD simulation signal test and comparative analysis

Traditional signal processing methods, such as the Fourier transform and wavelet transform, use kernel functions to locally transform the original signal to obtain the corresponding local fluctuation characteristics [14]. According to the basic principles and characteristics of the above LMD method, the LMD method separates, recombines, and integrates the amplitude and frequency of the original signal into an overall time spectrum through the characteristic scale of the signal, reflecting the local fluctuation characteristic information contained in the original signal. Give a simulation signal for testing, and compare the influence of the Fourier transform method, wavelet transform method, wavelet transform method and LMD method on signal processing, signal $x(t)$.

$$x(t)=2(1+\cos(40\pi t))\cos(200\pi t)+10(1+\cos(40\pi t))\cos(600\pi t) \quad (11)$$

Among them, $t\in[0,1]$ and $x(t)$ are composed of six different frequency signals, namely 80Hz, 100Hz, 120Hz, 280Hz, 300Hz and 320Hz. The simulation signal has six fluctuation characteristic scales. The waveform diagram of $x(t)$ is shown in Figure 1, and the spectrum diagram of $x(t)$ is shown in Figure 2.

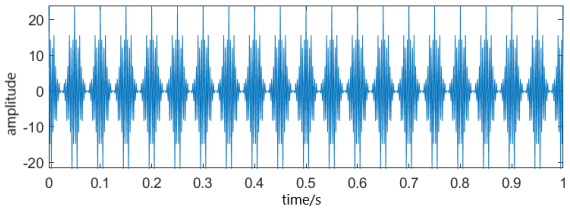

Figure 1. Simulation signal waveform

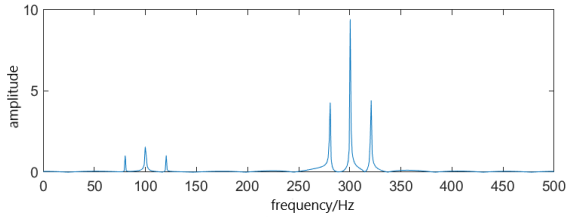

Figure 2. Spectrum diagram of simulation signal

The signals are processed by the wavelet transform method, the Fourier transform method, and the LMD method, respectively. The results are shown in Figure 3, Figure 4, and Figure 5.

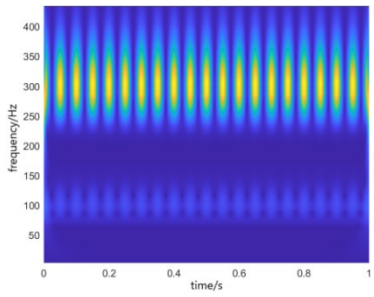

Figure 3. Wavelet transform time-frequency diagram

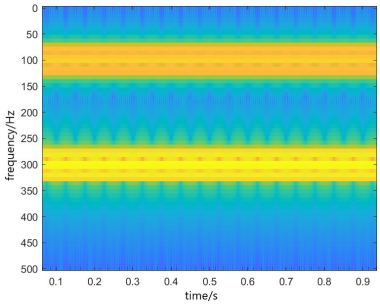

Figure 4. Fourier transform time-frequency diagram

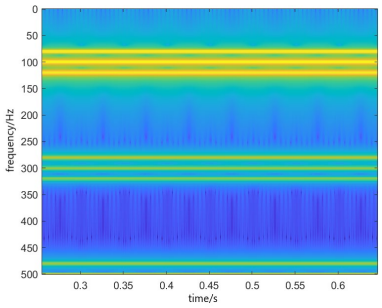

Figure 5. LMD time-frequency diagram

While all three methods are applicable to signal processing and represent basic signal characteristics, their outcomes exhibit considerable divergence. The wavelet transform provides superior frequency resolution for lower bands (80–120 Hz) compared to higher bands (280–320 Hz), yet its overall limited resolution and inability to maintain consistency across the spectrum restrict its effectiveness. The Fourier transform yields a stable but low-resolution time-frequency distribution due to fixed windowing and uncertainties, which poorly discriminates between fluctuation features. Conversely, the adaptive and independent decomposition of the LMD method enables high resolution across frequencies, delivering a clear separation of each fluctuation mode in the time-frequency representation with negligible cross-interference.

2.2. PSO and its related improvements

Improved PSO algorithm

In n-dimensional space, initialize the particle population [15-16]. The number of particles is m. Define the position $x_i = (x_{i1}, x_{i2}, \cdots, x_{id})$ and velocity $v_i = (v_{i1}, v_{i2}, \cdots, v_{id})$ of the ith particle in the population, including $i = 1,2,\cdots,m$. The optimal positions of individual and group of particles are $p_i = (p_{i1}, p_{i2}, \cdots, p_{id})$ and $p_g = (p_{g1}, p_{g2}, \cdots, p_{gd})$ respectively. Update the population. In iteration $k+1$, update the position and speed of particles according to the following formula:

$$v_{ij}^{k+1} = w \cdot v_{ij}^{k} + c_1 r_1 (p_{ij}^{k} - x_{ij}^{k}) + c_2 r_2 \left(p_{gj}^{k} - x_{ij}^{k}\right) \quad (12)$$
$$x_{ij}^{k+1} = x_{ij}^{k} + v_{ij}^{k+1}$$

Where, v_{ij}^{k+1} represents the velocity of particle in the j-th dimension in the $k+1$ iteration; x_{ij}^{k+1} represents the position of particle in the j-th dimension in the $k+1$ iteration; w is inertia weight r_1, r_2 represents a random number between $[0,1]$ c_1, c_2 is the learning factor; p_{ij}^{k} represents the individual extreme value of particle in the J-dimension in the $k+1$ iteration; p_{gj}^{k} represents the global extremum of the j-th dimension of particle in the $k+1$ iteration.

When PSO algorithm reaches local optimization. The update of particle velocity is entirely determined by. Because the inertia weight of the traditional fixed parameter PSO algorithm is usually smaller than 1, the particle velocity will update with an attenuation trend and may stop moving. That is, the algorithm will have premature convergence. If the attenuation trend is reduced, the convergence speed of the algorithm will be affected. Therefore, we can see that how to correctly evaluate the premature convergence is particularly important for the improvement of the next algorithm.

If the objective function value corresponding to particle position X_i in the population is the particle fitness value f_i, the overall average fitness value of the population is:

$$f_{avg} = \frac{1}{n}\sum_{i=1}^{n} f_i \quad (13)$$

If the fitness value of particles in the population is higher than f_{avg}, $f_{avg}^{'}$ is obtained by averaging, which is defined as follows:

$$\Delta = \left| f_m - f_{avg}^{'} \right| \quad (14)$$

Among them, Δ can be used as the criterion to evaluate premature convergence. If it is smaller, it can show that the whole population is closer to premature convergence.

According to the above analysis, the inertia weight should be adjusted adaptively according to the level of premature convergence of the group. In the initial stage of the Algorithm, the inertia weight should be reduced, and particles should be used for local optimization to speed up the convergence speed of the whole algorithm. In the later stage of the algorithm, the inertia weight should be increased to enable the particles to conduct global optimization. A large inertia weight is conducive to jumping out of the local optimization in the later stage and avoiding premature convergence as much as possible. The specific adjustment of inertia weight is as follows:

$$\begin{cases} w = w - (w - w_{\min}) \cdot \left(\dfrac{f_{\max} - f_i}{f_{\max} - f_{avg}^{'}} \right), f_i \geq f_{avg}^{'} \\ w, f_{avg}^{'} \leq f_i \leq f_{\max} \\ w = 1.5 - \dfrac{1}{1 + k_1 \cdot \exp(-k_2 \cdot \Delta)}, f_i \leq f_{avg} \end{cases} \quad (15)$$

The steps of the improved PSO algorithm are as follows, and the flow chart is shown in Figure 6:

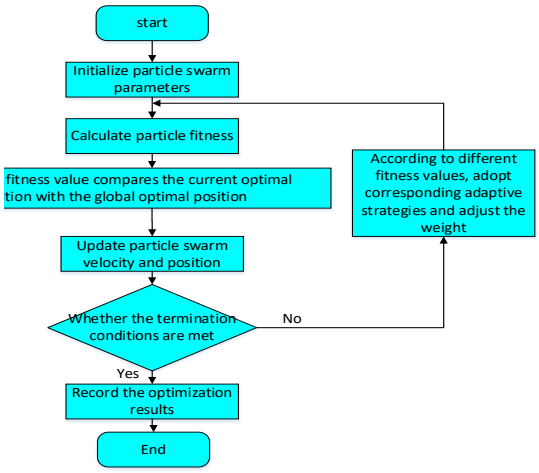

Figure 6. Improved PSO algorithm flow

1) Initialize particle swarm optimization and parameters;
2) Calculate the fitness value of each particle in the population;
3) If the fitness value is better than the current optimal position and the global optimal position, update the particle speed and position;
4) Judge whether the termination conditions are met. If step 6 is met, if step 5 is not met;
5) Adopt corresponding adaptive strategies according to different fitness values, adjust the weight, and turn to step 2;
6) Record the optimization results.

IPSO algorithm performance test

In order to better observe the optimization capability of the improved PSO algorithm, it was tested against a set of benchmark functions listed in Table 1. In order to highlight the advantages of the improved PSO algorithm, these test functions are used to test the standard PSO algorithm at the same time.

Table 1. Two test functions

Function name	Function form	Domain definition	optimal value
Sphere	$f_1 = \sum_{i=1}^{n} x_i^2$	$(-100,100)$	0
Girewank	$f_4 = 1/4000 \sum_{i=1}^{n} x_{i+1} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) +$	$(-600,600)$	0

Each of the above test functions is tested 30 times, respectively, and the parameters of the standard PSO algorithm and the improved PSO algorithm are set as follows: the number of population $m = 20$, the dimension of particles $n = 30$, and the maximum number of iterations $\max gen = 150$.

1) Sphere function

The image of the sphere function is shown in Figure 7, the optimization process of the two algorithms tested is shown in Figure 8, and the final results are shown in Table 2.

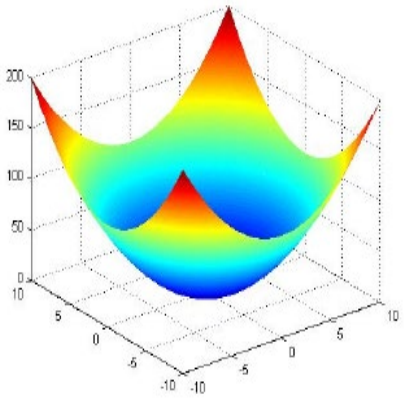

Figure 7. Sphere function image

Table 2. Sphere function test results

Function name	Test algorithm	Average convergence value	Optimal convergence value	Number of iterations
Sphere	PSO	0.5307	0.0185	134
	IPSO	0.0054	0.0053	24

2) Girewank function

The image of the girewank function is shown in Figure 9, the optimization process of the two algorithms tested is shown in Figure 10, and the final results are shown in Table 3.

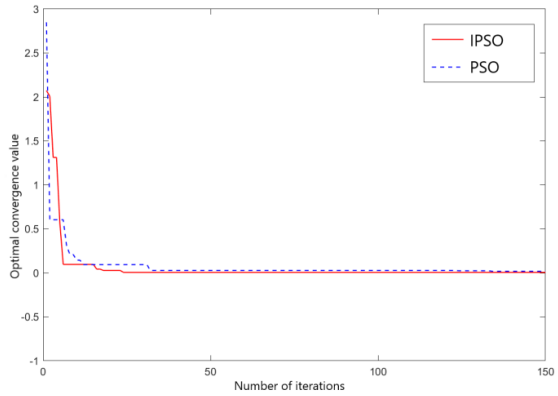

Figure 8. Comparison of Sphere function optimization

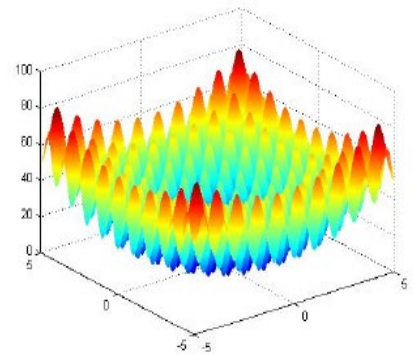

Figure 9. Girewank function image

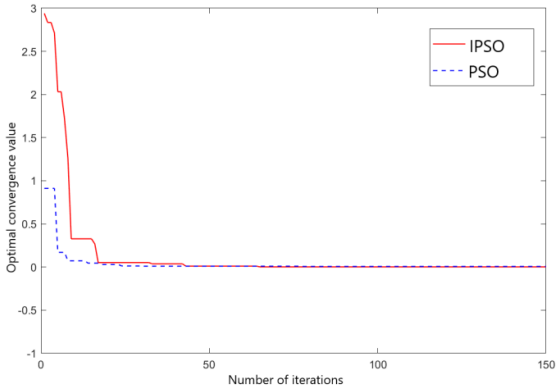

Figure 10. Comparison of Girewank function optimization

Table 3. Girewank function test results

Function name	Test algorithm	Average convergence value	Optimal convergence value	Number of iterations
Girewank	PSO	0.2641	0.0070	78
	IPSO	0.0046	0.0020	65

As evidenced by the results across various test functions, the proposed PSO algorithm achieves convergence values that exhibit a higher degree of proximity to the theoretical optimum. This capability notably prevents premature convergence by facilitating escape from local optima. Moreover, a marked reduction in the number of iterations required to attain the optimum, relative to the standard algorithm, validates the improved PSO's robust optimization capability and accelerated convergence speed.

2.3. Least Squares Support Vector Machine

Basic principles

LSVM transforms inequality constraints in SVM into equality constraints. Based on the quadratic programming method using the sum of error squares loss function instead of SVM, the quadratic programming problem is transformed into a linear matrix problem. The specific principle is as follows:

Given a set of data sets, $(x_i, y_i)_{i=1,2,\cdots,N}$ and x_j are the input vector of the j_{th} sample, y_j is the corresponding output, and N is the number of samples contained in the data set through nonlinear mapping $\varphi(x)$ samples to high-dimensional space.

$$y(x) = w \cdot \varphi(x) + b \tag{16}$$

Where w is the weight vector and b is the offset vector. According to the principle of structural minimization, the LSVM model can be expressed as:

$$\begin{cases} \min J(w,\xi) = \frac{1}{2}\|w^2\| + \frac{1}{2}\gamma\sum_{i=1}^{N}\xi_i^2, j = 1,2,\cdots,N \\ s.t.\ \ y_i = w^T \cdot \varphi(x_j) + b + \xi_i \end{cases} \tag{17}$$

Where γ is the penalty coefficient greater than 0 and ξ_j is the errorness. When solving the minimization problem, a Lagrange multiplier λ_i is added to form the Lagrange function, and the KKT condition is used. The LSVM model is transformed into:

$$y(x) = \sum_{j=1}^{N} \lambda_i K(x, x_j) + b \tag{18}$$

Where $K(x_i, x_j) = \varphi(x_i)^T \cdot \varphi(x_j)$ is the kernel function satisfying Mercer condition.

Selection of kernel function and optimization of its parameters

The choice of kernel function plays a crucial role in ensuring the correct classification performance of the LSVM. An appropriate kernel function is fundamental to building an

effective prediction model. In this paper, the radial basis function (RBF) is adopted, which is expressed as follows:

$$K(x,y)=\exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \tag{19}$$

Where σ is the core radius, $\sigma>0$. For any position of the basis function, there is a unique corresponding support vector; the size of the parameters is determined by the algorithm, and the radius is symmetrical and smooth. Therefore, this paper selects the radial basis function as the kernel function of LSVM.

Due to the size of the parameters of the SVM, the accuracy of the final prediction rate will be affected. Kernel radius σ and penalty factor C are variables in radial basis function kernel function. The size of kernel radius σ will directly affect the operation efficiency of LSVM. The size of penalty factor C will affect the error of LSVM. If the value is small, the error will increase. And if it is too large, the adaptability of LSVM will become weak. Therefore, it is necessary to select appropriate optimization means to optimize these two parameters. In this paper, the IPSO algorithm verified and analyzed above is used to optimize the parameters of kernel radius σ and penalty factor C in LSVM.

3. LMD-IPSO-LSVM short-term photovoltaic power prediction model

3.1. Construction of model

According to different types of weather, the original meteorological data are divided into three types: sunny day, cloudy day, rain, and snow. From the four seasons of spring, summer, autumn and winter, the day with the same weather type as the day to be predicted and the closest daily maximum and minimum temperature, irradiance and air humidity are selected as similar days in each quarter. Using similar days to train the model can improve the accuracy of the prediction.

According to the actual situation of the photovoltaic power station. The data that will be used when building the model include: 49-point power data, temperature data, irradiance data, and humidity data collected every 15 minutes from 6:00 am to 6:00 pm. The specific steps of building the model are as follows:

1) Use the following Eq. (20) to normalize and inverse normalize the data:

$$x_i^*=\frac{x_i-x_{\min}}{x_{\max}-x_{\min}} \tag{20}$$

Where x_i^* is the normalized data; x_i is the first data input; $x_{\min}$ is the minimum value in the input data; $x_{\max}$ is the maximum value in the input data.

The inverse normalization formula is as follows:

$$p_i=\frac{(p_i^*)(p_{\max}-p_{\min})}{2}+p_{\min} \tag{21}$$

Where, p_i is the output power of the photovoltaic system after inverse normalization; p_i^* is the normalized photovoltaic system output power at the second time point; $p_{\max}$ is the maximum output power of photovoltaic system; $p_{\min}$ is the minimum value of output power of photovoltaic system.

2) For the same weather type in spring, summer, autumn and winter, Euclidean distance is used to select similar daily data.

$$d_{ik}=\sqrt{\sum_{j=1}^{4}(x_{ij}-x_{kj})^2}\quad i,k\in[1,n],k\neq i \tag{22}$$

$X_{i1},X_{i2},X_{i3},X_{i4}$ represents the daily maximum and minimum temperature, irradiance and air humidity of the day to be predicted; $X_{k1},X_{k2},X_{k3},X_{k4}$ represent the daily maximum and minimum temperature, irradiance and air humidity on the day of the data set.

The normalized similar daily data are decomposed by LMD. Taking the output power of the photovoltaic system on cloudy days as an example, there are 196 points in four seasons of the year. The decomposition results are shown in Figure 11, Figure 12, Figure 13, and Figure 14.

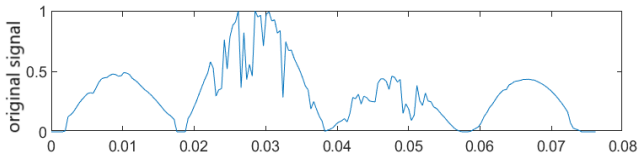

Figure 11. Similar day data on cloudy days

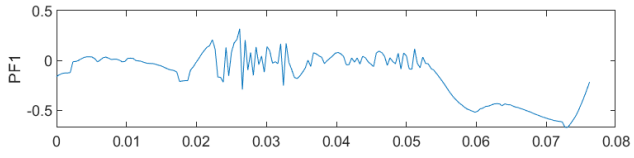

Figure 12. PF1 component on cloudy days

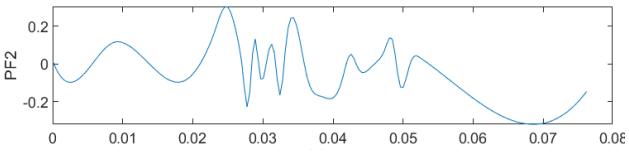

Figure 13. PF2 component on cloudy days

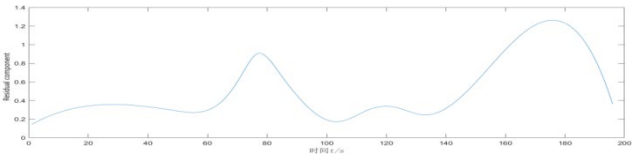

Figure 14. Cloudy residual component

4) The multi-input and multi-output mode is adopted to establish a model for each component for training. The temperature, irradiance, and humidity data at each time point on similar days are taken as inputs. The IPSO-LSVM model is established to predict the value of PF1 at each time point. The model training process is depicted in Figure 15.

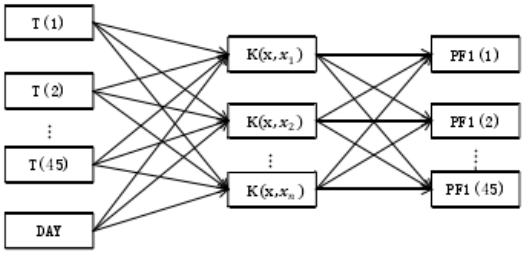

Figure 15. Model training

By analogy, the prediction models of each component under other weather types can be obtained.

5) The trained model is used to predict it, and the predicted values of each component at each time point are obtained. Then, it is equivalently weighted and inverse normalized by Eq. (21). The power prediction results for the day to be predicted can be obtained. The prediction process is shown in Figure 16.

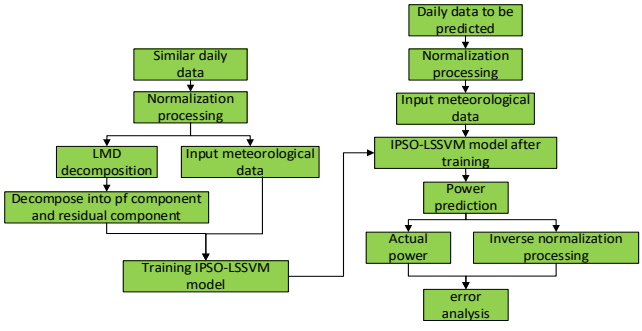

Figure 16. Overall forecasting process

3.2. LMD-IPSO-LSVM photovoltaic power prediction model

The involved principles and improved methods are synthesized, and the photovoltaic power prediction model based on LMD-IPSO-LSVM is built. The overall structure of the model is shown in Figure 17.

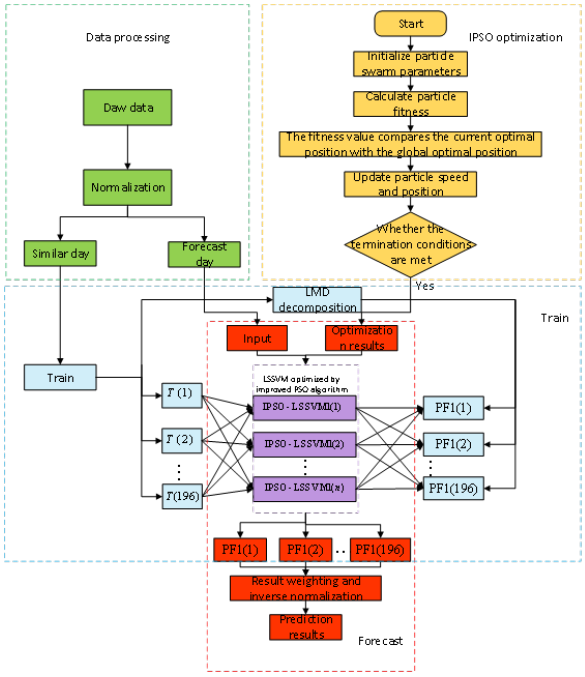

Figure 17. Structure diagram of prediction model based on LMD-IPSO-LSVM

The overall prediction model is divided into four parts, including the data processing part, the improved PSO algorithm optimization part, the training part, and the prediction part. The data processing part uses normalization to process the original meteorological data and photovoltaic power data. It can avoid large errors caused by input and output in terms of order of magnitude. The similar day data of the day to be predicted are selected by using the Euclidean distance, and then the prediction model is trained. Improve the accuracy of the overall prediction from the basic data. The improved PSO algorithm is verified by using the improved PSO kernel optimization algorithm. The optimized results are used in the LSVM algorithm for final prediction. In the training part, LMD is used to decompose the similar day data of the day to be predicted, processed by the data processing part, to train the model. An Ipso-LSVM prediction model is established for each component to train the model. In the prediction part, the trained improved PSO-LSVM prediction models are used to predict it. Finally, the photovoltaic predicted power of the day to be predicted can be obtained by equivalent weighting and inverse normalization of the results.

3.3. Error evaluation index

Normalized relative error (RE) and normalized mean relative error (MRE) are used to evaluate the predicted results. The smaller the value, the better the accuracy of the prediction model. The relevant formula is as follows:

$$RE = \frac{\left|W_{forecasting} - W_{true}\right|}{W_{total}} \times 100\% \quad (23)$$

$$MRE = \frac{1}{N}\sum_{t=1}^{N}\frac{\left|W_{forecasting} - W_{true}\right|}{W_{total}} \times 100\% \quad (24)$$

Where $W_{forecasting}$ is the predicted value; W_{true} is the tru3 value; W_{total} is the total installed capacity of photovoltaic; t is the time variable; N is the number of samples.

4. Experimental simulation

4.1. Original data

The proposed LMD-IPSO-LSVM-based photovoltaic power prediction model requires validation and analysis using actual data from a photovoltaic power plant. The experimental data are from some photovoltaic modules of a photovoltaic power station in Liaoning, China.Data including photovoltaic power, solar irradiance, temperature, humidity, and weather type were collected at 15-minute intervals. The acquisition period spanned from 6:00 AM to 6:00 PM daily, covering the entire year of 2020. According to different weather types, the sample data are divided into sunny days, cloudy days, rainy days, and snowy days. To enhance prediction accuracy, we explored the model's performance under different weather types; for each type, the evaluation was conducted as follows. one day is selected from the four quarters of spring, summer, autumn, and winter as the day to be predicted, and the four quarters are predicted at the same time.

4.2. Photovoltaic power prediction example

In order to compare and verify the performance and advantages of the model, standard PSO-LSVM and LSVM models are established respectively, and the samples are also simulated and tested. The results are compared with the model proposed in this paper. The results are as follows.

1) Sunny day

When the weather type is sunny, the prediction results obtained by the three prediction methods are shown in Figure 18, and the error results are shown in Figure 19.

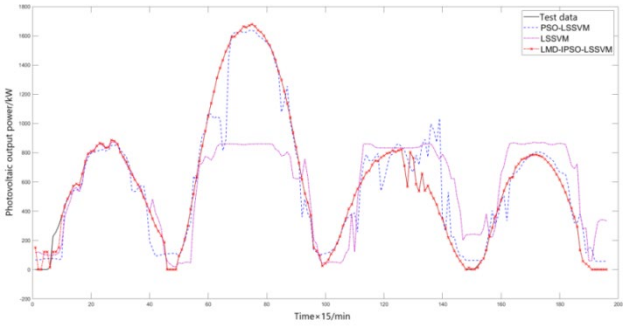

Figure 18. Photovoltaic power prediction curve under sunny weather in four quarters

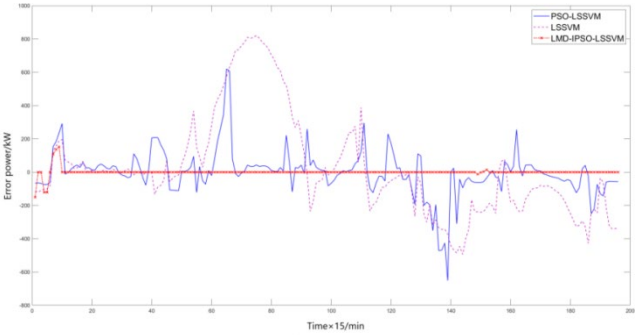

Figure 19. Photovoltaic power prediction error curve under sunny weather in four quarters

2) Cloudy day

When the weather type is sunny, the prediction results obtained by the three prediction methods are shown in Figure 20, and the error results are shown in Figure 21.

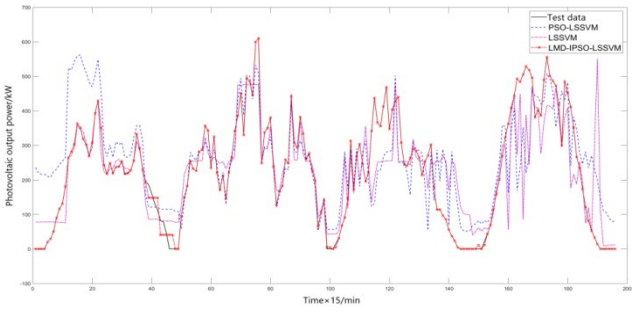

Figure 20. Photovoltaic power prediction curve under overcast weather in four quarters

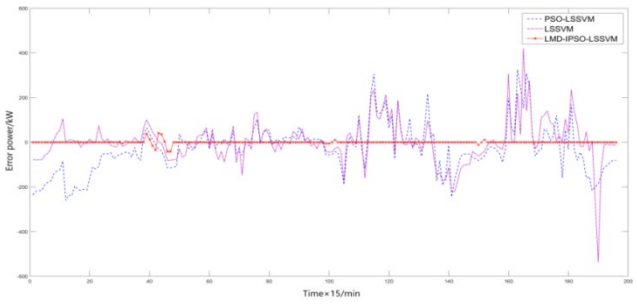

Figure 21. Photovoltaic power prediction error curve under overcast weather in four quarters

5) Rain and snow
When the weather type is rain and snow, the results predicted by the three methods are shown in Figure 22, and the error results are shown in Figure 23.

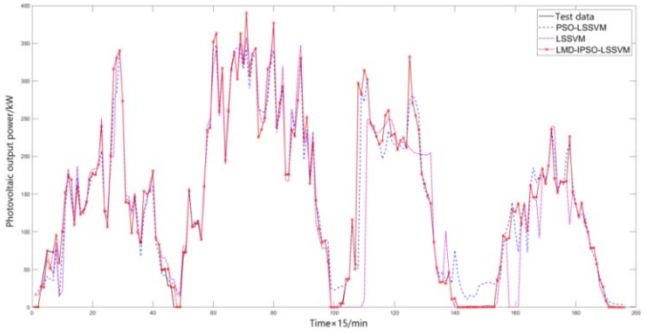

Figure 22. Photovoltaic power prediction curve in rainy and snowy weather in four quarters

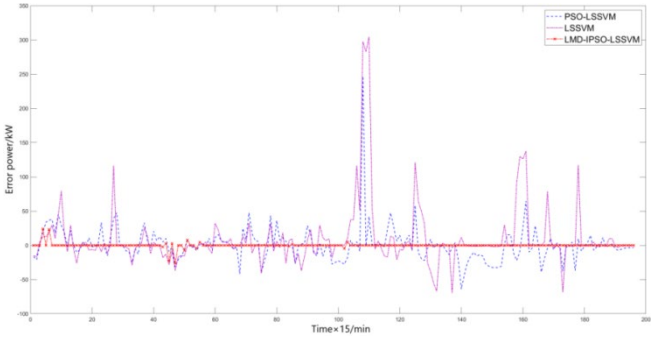

Figure 23. Photovoltaic power prediction error curve under rainy and snowy weather in four quarters

The analysis of the above results demonstrates that, regardless of weather conditions or season, the model consistently exhibits robust performance. The prediction results of the short-term photovoltaic power prediction model based on LMD-IPSO-LSVM proposed in this paper can fit the test data well, the error fluctuation is small, and the predicted value is also in perfect agreement with the real value. Compared with the other two methods, there is a large error. It shows that the method proposed in this paper can accurately predict the output power of a photovoltaic system under different weather types, and the range of error fluctuation is also small.

4.3. Comparison with the deep learning models

In order to testify to the effectiveness of LMD-IPSO-LSVM, Several experiments were carried out in this paper. Figure 24 is the Comparison with the Deep Learning Model. This figure presents a multi-perspective performance comparison between the proposed LMD-IPSO-LSVM model and mainstream deep learning methods:

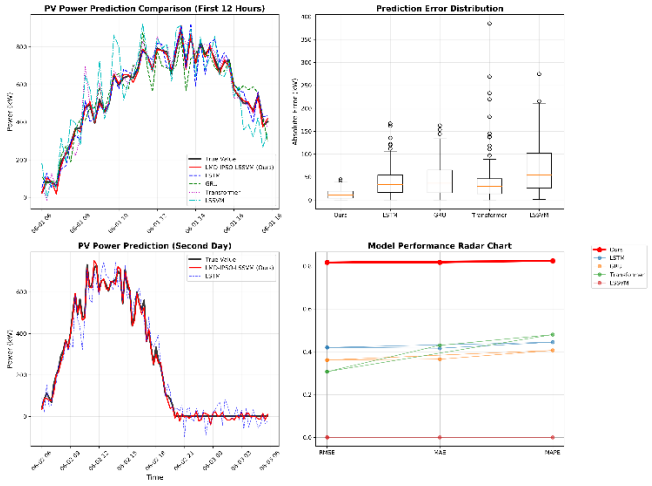

Figure 24. Comparison with Deep Learning Model

(i) **Top-left:** Comparison of prediction results for the first 12 hours, showing that the proposed model most closely matches the true values.
(ii) **Top-right:** Box plot of prediction error distribution, indicating that the proposed model achieves the smallest errors with the most concentrated distribution.
(iii) **Bottom-left:** Full-day prediction comparison for the second day, validating the model's adaptability under different weather conditions.
(iv) **Bottom-right:** Radar chart of performance metrics, demonstrating the superior performance of the proposed model across all three metrics: RMSE, MAE, and MAPE.

The results demonstrate that the LMD-IPSO-LSVM model outperforms the compared deep learning methods in both prediction accuracy and stability.
Figure 25 presents a comparative analysis of the proposed model against representative deep learning models (LSTM,

Transformer) across the entire prediction horizon. The following key observations can be made:

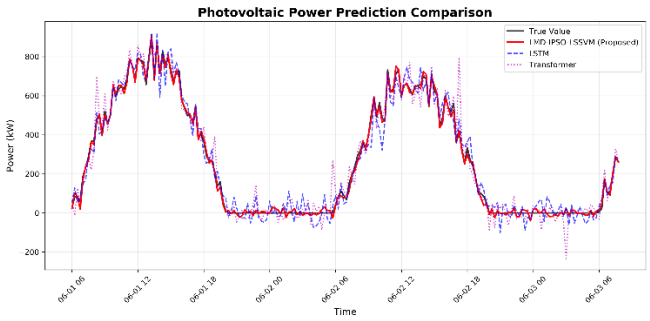

Figure 25. Comparison of Prediction Results from Main Models

(i) The prediction trajectory of the LMD-IPSO-LSVM model demonstrates near-perfect alignment with the ground truth values (black line).
(ii) The LSTM model exhibits significant temporal lag during periods of abrupt power fluctuations.
(iii) The Transformer model shows substantial deviations at specific time intervals.
(iv) The proposed model effectively captures rapid transient characteristics in photovoltaic power generation.

These findings validate the efficacy of integrating Local Mean Decomposition (LMD) and Improved Particle Swarm Optimization (IPSO) in enhancing forecasting precision.
Figure 26 illustrates the performance comparison between the proposed model and representative deep learning models (LSTM, Transformer) across the entire prediction horizon. The following observations can be clearly made.

(i) The forecast curve of the LMD-IPSO-LSVM model (red line) demonstrates near-perfect alignment with the actual values (black line).
(ii) The LSTM model (blue dashed line) exhibits significant lag during periods of sharp power fluctuations.
(iii) The Transformer model (purple dotted line) shows substantial deviations at certain time points.
(iv) The proposed model more effectively captures the rapid variation characteristics of photovoltaic power.

These results validate the effectiveness of Local Mean Decomposition (LMD) and Improved Particle Swarm Optimization (IPSO) in enhancing prediction accuracy.

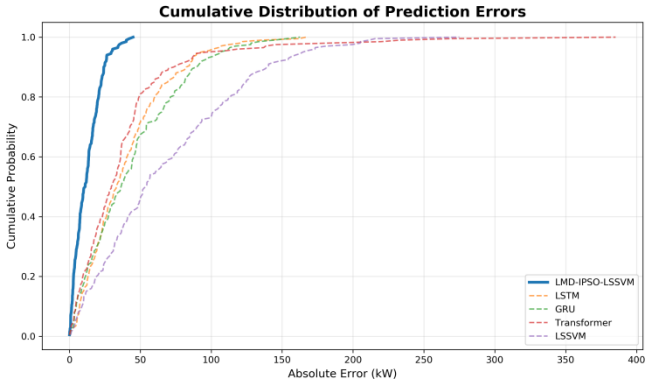

Figure 26. A Comparative Analysis of Cumulative Prediction Error Distributions

4.4. Error evaluation

The effectiveness of the proposed method was further verified by evaluating the prediction results for the three weather types against the normalized RE and MRE metrics (Eqs. 23 and 24). The evaluation results are shown graphically in Figures 27 – 29 and tabulated in Table 4.

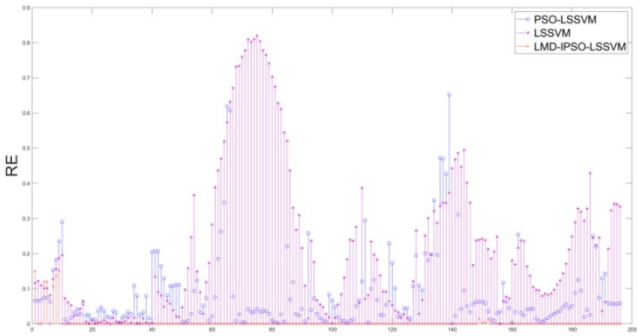

Figure 27. Normalized relative errors of three methods in sunny weather

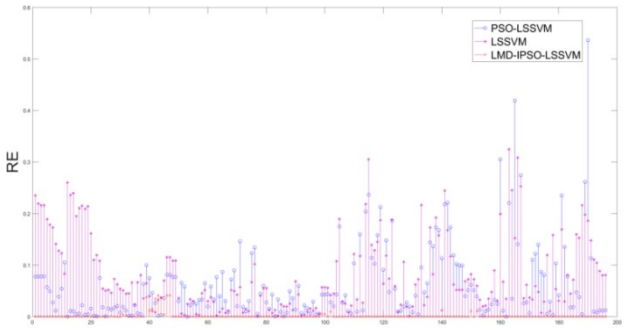

Figure 28. Normalized relative error of three methods in overcast weather

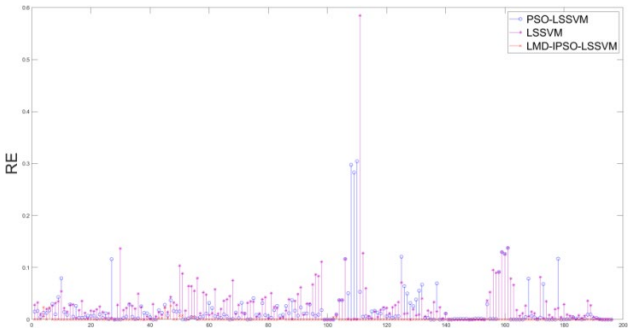

Figure 29. Normalized relative error of three methods in rainy and snowy weather

Table 4 Comparison of prediction errors of three methods

Prediction method	Model	MRE%
PSO-LSVM	sunny day	8.51
	overcast	6.66
	Rain and snow	2.06
	average value	5.74
LSVM	sunny day	11.46
	overcast	8.33
	Rain and snow	2.94
	average value	7.58
LMD-IPSO-LSVM	sunny day	0.42
	overcast	0.16
	Rain and snow	1.04
	average value	0.54

It can be seen that the minimum MRE error of three weather types predicted by this method is 0.16% and the maximum is 1.04%. The average MRE error corresponding to the three weather types is 0.54%. The minimum value of the average MRE error predicted by the PSO-LSVM method is 2.06%, and the maximum value is 8.51%. The average MRE error corresponding to the three weather types is 5.74%. The minimum error of MRE predicted by the LSVM method is 2.94% and the maximum error is 11.46%. The average MRE error corresponding to the three weather types is 7.58%. Using the method proposed in this paper, the MRE error is small and the error floating range is also small. It shows that the prediction accuracy is greatly improved compared with the other two methods.

Figure 30 illustrates the three-dimensional forms of five standard test functions used to evaluate the performance of optimization algorithms. These functions include:

(i) **Sphere:** A simple unimodal convex function for testing basic convergence capability.

(ii) **Rosenbrock:** A function featuring a narrow parabolic valley, used to examine algorithm performance in non-convex regions.

(iii) **Rastrigin:** A multimodal function with numerous local optima, assessing the algorithm's ability to escape local minima.

(iv) **Ackley:** A complex function containing multiple local optima, evaluating global exploration capability.

(v) **Griewank:** A function with regularly distributed local optima, testing precision search ability.

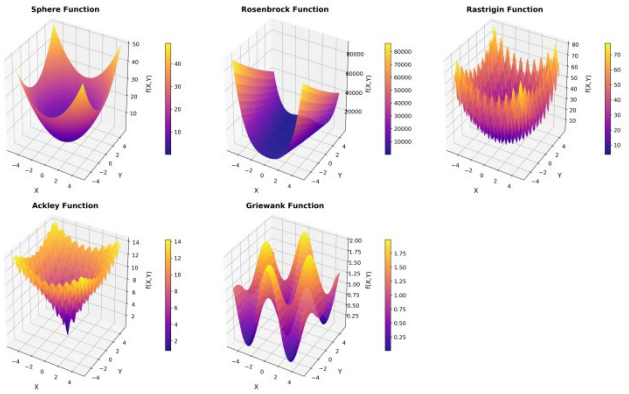

Figure 30. 3D Visualization of Test Functions

From Figure 30, it can be seen that the diverse characteristics of these test functions establish a solid foundation for comprehensively evaluating the performance of the IPSO algorithm.

Figure 31 presents a comparative analysis of the convergence processes between standard Particle Swarm Optimization (PSO) and Improved PSO (IPSO) across five test functions. Key observations include:

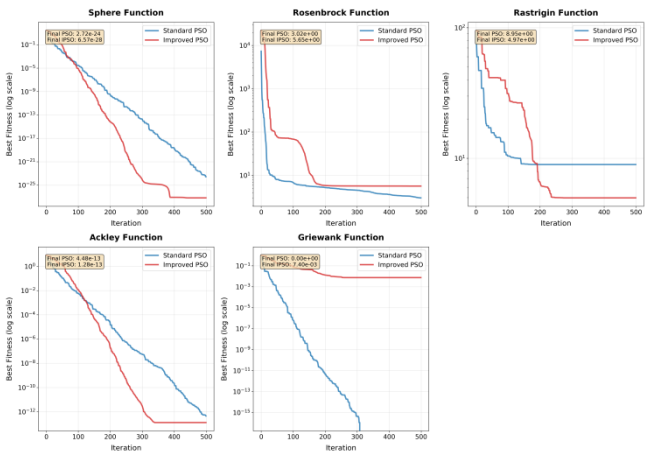

Figure 31. A Comparative Analysis of Convergence Curves between PSO and IPSO

(i) IPSO (red curve) demonstrates significantly faster convergence rates than standard PSO (blue curve) across all test functions.

(ii) The improved algorithm achieves lower fitness values, indicating enhanced optimization precision.
(iii) IPSO's superiority is particularly pronounced on multimodal functions (Rastrigin, Ackley), confirming that its modified mechanisms effectively strengthen the ability to escape local optima.
(iv) The text annotations on the convergence curves display the final fitness values upon algorithm termination, providing visual evidence of IPSO's performance advantages.

From Figure 31, it can be seen that these results validate that through the incorporation of linearly decreasing inertia weight and adaptive learning factors, IPSO successfully balances global exploration and local exploitation capabilities.
Figure 32 presents a comparative visualization of the average best fitness values achieved by PSO and IPSO across five test functions using a bar chart format. The following conclusions can be drawn as follows:

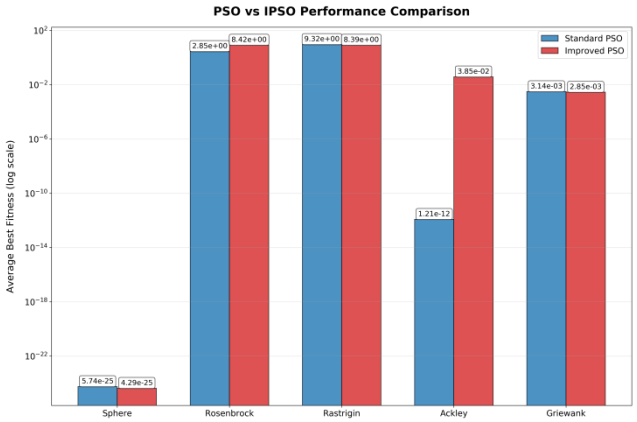

Figure 32. A Comparative Bar Chart of Performance between PSO and IPSO

(i) IPSO (red bars) demonstrates superior average performance over standard PSO (blue bars) across all test functions.
(ii) The degree of performance improvement varies with function characteristics, showing particularly significant gains on multimodal functions.
(iii) Numerical labels atop each bar precisely indicate the mean fitness values for both algorithms.
(iv) The logarithmic-scaled vertical axis effectively reveals performance differences across multiple orders of magnitude.

These statistical results validate the superiority and robustness of the IPSO algorithm when applied to diverse optimization problems.

5. Conclusion

In this paper, PSO-LSVM and LSVM models are established to predict the same test samples. The prediction results and error results are compared with the model established in this paper. At the same time, in order to further verify the accuracy, a variety of error evaluation indicators are used to evaluate the prediction results, and the results are analyzed. The final results show that the model established in this paper has good performance in different weather types and different seasons.

References

[1] Research on short-term photovoltaic power prediction ZHOU Y, ZHOU N R, GONG L H, et al. Prediction of photovoltaic power output based on similar day analysis, genetic algorithm, and extreme learning machine[J]. Energy, 2020, 204: 117894.
[2] Duan H M, Luo X L.A novel multivariable grey prediction model and its application in forecasting coal consumption[J]. ISA Transactions,2022,120:110-127.
[3] Zhou W, Ding S.Anovel discrete grey seasonal model and its applications[J]. Communications in Nonlinear Science and Numerical Simulation, 2021, 93:105493.
[4] Zhang Yujin, Yang Lingfan, Ge Shuangzhi, et al. Short-term photovoltaic power forecasting based on K-means algorithm and support vector machine[J]. Power System Protection and Control, 2018, 46 (21): 118-124.
[5] National Energy Administration. The 13th year plan for solar energy development [J] Solar energy, 2016 (12): 5-14.
[6] SHANGGUAN Xin, QIN Wenping, XIA Fuliang, et al. Pole-to-ground fault protection scheme for MMC multi-terminal flexible DC distribution network based on Pearson correlation of transient voltage[J]. High Voltage Engineering, 2020, 46(5): 1740-1749.
[7] Zheng S , Li D , Li Y ,et al.An adaptive weighted stacking ensemble framework for photovoltaic power generation forecasting with joint optimization of features and hyperparameters[J].Engineering Applications of Artificial Intelligence, 2025, 144(000). DOI:10.1016/j.engappai.2025.110075.
[8] Jing Bo Research on short-term power prediction method of photovoltaic power station [D] Jiangsu University, 2017.
[9] Yongqian L, Hao Z, Jic Y, etc. Hybrid relevance vector machine model or wind power forecasting[C]. International Conference on Renewable Power Generation. 1ET, 2016.
[10] Zhang Y,Li S,Li J,et al.A time power-based grey model with Caputo fractional derivative and its application to the prediction of renewable energy consumption[J]. Chaos, Solitons & Fractals, 2022, 164:111275 0.
[11] Zhou W, Wu X,10Ding S, et al.Application of a novel discrete grey model for forecasting natural gas consumption: A case study of Jiangsu Province in China[J]. Energy, 2020, 200: 117443.
[12] Malvoni M, Giorgi M G D, Congedo P M. Data on photovoltaic power forecasting models for Mediterranean climate[J]. Data in Brief, 2016, 7(C): 1639-1642.
[13] Xu Z C, Dun M, Wu LFE. Evaluating the effect of sample length on forecasting validity of FGM(1,1)[J]. Alexandria Engineering Journal, 2020, 59:4687-4698.
[14] Quaiyum S, Khan Y, Rahman S. Artificial neural network-based short-term load forecasting of power system[J].

International Journal of Computer Applications, 2011, 30(4): 1-7.
[15] Huang R,Wei C,Wang B,et al.Well performance prediction based on Long Short-Term Memory (LSTM)neural network[J]. Journal of Petroleum Science and Engineering, 2022, 208: 109686.
[16] Zhang J, Xia P. An improved PSO algorithm for parameter identification of nonlinear dynamic hysteretic models [J]. Journal of Sound and Vibration, 2017(5): 153-167.