

# Optimizing Short-term Power Load Forecasting with a Hybrid PSO-SVR-LSTM Approach based on Data Analysis

Shakeel Ahmad, Jinsong Tao, and Rahat Ali

**Abstract**—The increasing complexity of an electricity consumption patterns poses significant challenges for power system management. This study aims to develop a reliable short-term load forecasting (STLF) framework to support energy companies in managing consumer energy demand. This work has two key contributions. First, the model employs Support Vector Regression (SVR) to handle complex, nonlinear load patterns, followed by the use of Long Short-Term Memory (LSTM) networks to model extended temporal dependencies in the data. Second, to address the limitations of empirical hyperparameter tuning, Particle Swarm Optimization (PSO) is utilized to automatically fine-tune the parameters of both SVR and LSTM modules. Experimental validation on data from the China State Grid Handan Electric Power Company shows the superiority of the proposed model, achieving a 15% reduction in mean absolute error (MAE) compared to existing methods. The framework shows improvements in multi-step forecasting, accurately predicting peak and trough load values while maintaining robustness against input feature variations. Additionally, an uncertainty quantification analysis confirms the model's reliability across different forecasting horizons (30, 45, and 60 minutes), with MAE values as low as 0.7512 for 30-minute predictions. These results show that the framework can improve power system efficiency and scale up for real-world energy management.

**Index Terms**— Short-term load forecasting, SVR, LSTM, PSO

## I. INTRODUCTION

**D**UE to the steep increase in electricity consumption in recent periods, managing and optimizing power systems has become more challenging [1]. To ensure the efficient functioning of these systems, it is essential to have precise predictions of power load. The precision of power load forecasting will directly impact power plants' energy usage and financial gains due to information technology innovation and ongoing power market reform. In recent years, nations and businesses have continuously researched the smart power grid. A smart power grid system has allowed it to acquire enormous and high-quality load data sets, which are the foundation for deep learning and load forecasting [2].

Manuscript received January 29, 2025; revised May 3, 2025.

This work was supported in part by the China Huanneng Corp. Ltd under Grant HBXNY-ZX-QT-2022-019.

Shakeel Ahmad is a postgraduate student of Electrical Engineering Department, Wuhan University, China (phone:13164650793; e-mail: Shakeel@whu.edu.cn).

Jinsong Tao is an associate professor of Electrical Engineering Department, Wuhan University, China (e-mail: Jamson\_tao@whu.edu.cn).

Rahat Ali is a postgraduate student of Electrical Engineering Department, Wuhan University, China (e-mail: rahatali@whu.edu.cn).

According to Mocanu et al. [3]: load forecasting can be divided into three-time horizons: short-term (ranging from one hour to one week), medium-term (from one week up to a year), and long-term (exceeding one year). Among these, STLF is essential for the efficient functioning of power systems and market operations, as errors can disrupt reliable operations and lead to economic losses [4]. Power load data generally has distinct characteristics of time series and nonlinearity, necessitating a division into two prediction models: one based on statistical methods and the other driven by machine learning techniques [5]. It is important to ensure that the model effectively utilizes the temporal relationships and nonlinear trends present in power load data to support this theory and provide accurate and reliable forecasts.

The growing complexity of power systems and the availability of large volumes of data have led to a significant evolution in power load forecasting during the past few decades. Early forecasting methods were primarily based on statistical models like exponential smoothing and Autoregressive Integrated Moving Average (ARIMA) [6]. These time series forecasting approaches such as multivariable linear models and ARIMA [7], typically operate under the premise that the underlying data exhibits only linear behavior. Nonlinear components are also present in the majority of real-world time series data. Numerous nonlinear statistical techniques, including models like Autoregressive Conditional Heteroskedasticity (ARCH) model [8] and Generalized ARCH (GARCH) model [9], have emerged to handle time series data characterized by complex nonlinear dynamics. However, these models exist in various versions, each of which is only suitable for representing specific types of nonlinearities [10]. This increases the complexity of selecting an appropriate model for time series analysis.

Because of their ability to manage intricate data patterns, machine learning-based approaches have gained significant traction in the field of power load forecasting. Among these techniques, regression-based methods such as Support Vector Machines (SVMs) [11] and Random Forests [12] have been widely adopted in practice. Studies have demonstrated that SVM produces reliable forecasts, which are useful for regression problems with limited datasets. The selection of parameters frequently adjusted via genetic algorithms or grid search impacts SVM's performance.

In recent years, a variety of deep learning methods such as Artificial Neural Networks (ANN) [13], Backpropagation (BP) networks, Convolutional Neural Networks (CNN) [14] Recurrent Neural Networks (RNN) [15], and k-Nearest

Neighbors (KNN) [16] have been extensively utilized for time series forecasting, particularly to capture the nonlinear characteristics present in power load data.

RNNs are well-suited for processing sequential data, as they retain information from previous time steps through short-term memory of their activations [17]. RNNs are adept at handling sequential data because they keep a short-term memory by holding onto activations from every time step [18]. Training RNNs presents challenges due to instability in gradient values, where gradients may either diminish to near zero or grow uncontrollably during BP [19, 20]. To address this, LSTM networks and other gated topologies are often used [21]. LSTM networks use longer-term timing information to overcome the gradient issues of traditional RNNs [22, 23]. In line with this trend, Chen et al. [24] introduced a load forecasting approach utilizing deep residual networks, demonstrated a promising performance outcome. Kong et al. [25] developed a model based on LSTM networks, which can effectively handle significant load changes during prediction. A comparable LSTM-based structure was provided by Wang et al. [26]; their focus was centered on probabilistic prediction methods. Tan et al. [27] explored challenge of forecasting power consumption over ultra-short time intervals and suggested a strategy based on LSTM network. Nonetheless, local minima and overfitting are potential problems with LSTMs that necessitate cautious hyperparameter adjustment and optimization.

While single models sometimes fall short of meeting the requirements of prediction accuracy, power load forecasting has also investigated hybrid models that combine the best features of various methodologies. For example, LSTM power load prediction and principal component analysis were used to increase the prediction accuracy of STLF [28]. The obtained principal components were used to replace the original data as LSTM training samples by analyzing the correlation of the input features. In [29], a combined approach utilizing SVR and LSTM networks was applied for short-term power load forecasting. The prediction accuracy and error evaluation index are significantly improved compared with the single LSTM model. The study in [30] used a genetic algorithm (GA) with SVR to forecast demand. The findings indicated that the forecast produced by SVR-GA was more credible than the forecasts produced by ARIMA and BP neural networks.

A STLF model that integrates PSO with SVR was introduced in [31] to predict the peak load of an energy station in Jiangxi. A PSO-BP model is proposed in [32] to improve the load forecasting accuracy and power generation efficiency. The study in [33] utilized PSO to fine-tune the hyperparameters of the LSTM-based neural network in a power load prediction model, resulting in higher accuracy and stability than traditional methods. Related to this work is also approach from [34], where an improved sparrow search algorithm to optimize the short-term load forecasting model of BP-based neural network was proposed to address weak self-correction and the local optimization in recurrent neural networks. The foraging behaviour of the discoverer in the sparrow search algorithm was improved from jumping to moving, and the initial weight and threshold of BP-based neural network were optimized. Power load data from specific location is used to validate accuracy of model.

Although models such as SVR and LSTM have shown effectiveness in STLF, they typically depend on manually set hyperparameters, a process that is both inefficient and less than ideal. This paper addresses this gap by introducing a PSO-SVR-LSTM model that automates hyperparameter optimization, improving accuracy and computational efficiency. Our contributions include (1) a novel integration of SVR and LSTM to handle nonlinear and time-series dependencies, (2) use of PSO for efficient hyperparameter tuning, and (3) extensive validation on real-world data, showing improved performance compared to existing advanced forecasting techniques.

The structure of this paper is as follows: Section 2 presents a concise review of the methodology and relevant algorithms, followed by a comprehensive explanation of the proposed PSO-SVR-LSTM framework. Section 3 discusses the case study results along with an analysis of the approach. Section 4 concludes the study and outlines directions for future research.

## II. PSO-SVR-LSTM ALGORITHM

### A. Related model

#### Long Short-Term Memory Neural Network

Building on the architecture of RNNs, LSTM network introduces a more advanced "memory function" due to its gating units (input, output, and forget gates) [35]. This capability makes the LSTM network well-suited for predicting long-term, nonlinear sequences. The architectural structure of the LSTM network is illustrated in Fig. 1.

The model takes as input the data point  $x_t$  at time  $t$ , the memory value  $C_{t-1}$  at time  $t-1$ , and the output value  $h_{t-1}$  of the LSTM at same time step. The output data consist of the memory value  $C_t$  and the output value  $h_t$  of the LSTM at time step  $t$ . The "forget gate"  $F_t$  regulates the current cell state by selectively retaining or discarding information from the previous memory unit  $C_{t-1}$ . The  $F_t$  is given as

$$F_t = w_{xf}x_t + W_{hf}h_{t-1} + b_f \quad (1)$$

$$w_f = w_{xf}x_t, W_{hf}h_{t-1} \quad (2)$$

Where  $\sigma$  represents the sigmoid activation function,  $b_f$  is the bias term,  $W_{hf}h_{t-1}$  is weight matrix for input data  $h_{t-1}$ , and  $w_{xf}x_t$  is weight vector for input data  $x_t$ .

The "input gate" activates the sigmoid function and outputs the variable that is controlled between [0, 1]. The following are the expressions:

$$I_t = \sigma(W_{hi}h_{t-1} + w_{xi}x_t + b_i) \quad (3)$$

$$w_i = (w_{xi}x_t, W_{hi}h_{t-1}) \quad (4)$$

The shift in the cell state  $C_t$  is determine by modification of the forget gate and the input gate. This can be given by

$$C_t = I_t \cdot \tanh(w_{xc}x_t + W_{hc}h_{t-1} + b_c) + F_t \cdot C_{t-1} \quad (5)$$

The "output gate" governs the cell state's output at each timestep, as depicted in Fig. 1.

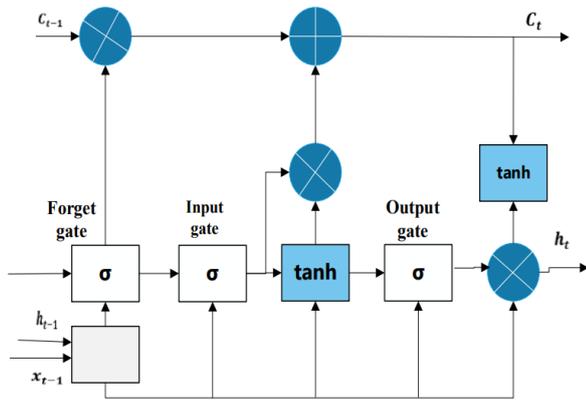


Fig.1. LSTM structure.

$$O_t = w_{xo}x_t + W_{ho}h_{t-1} + b_o \quad (6)$$

$$w_o = (w_{xo}x_t, W_{ho}h_{t-1}) \quad (7)$$

$$h_t = O_t \cdot \tanh(C_t) \quad (8)$$

### Support Vector Regression Machine

The SVR regression prediction algorithm introduced by Vapnik et al. [36, 37] is a statistical learning theory. SVR applies a nonlinear mapping to convert the original data from a lower-dimensional space into a more complex, higher-dimensional feature space. This mapping is achieved using a kernel function based on the fundamental principle of minimizing structural risk, allowing the algorithm to operate in a space where the data can be more easily distinguished using linear boundaries. After projecting the data into the high-dimensional space, SVR calculates the optimal hyperplane function that best fits the data and establishes a regression model.

While building the SVR model, the sample data  $(x_i, y_i)$  are transformed into a higher-dimensional space through a nonlinear function  $\varphi(x)$  and the regression estimate function,  $\omega$ , and  $b$  are generated as shown below;

$$f(x) = \omega^T \varphi(x) + b \quad (9)$$

During SVR training, the modelling problem can be transformed into a quadratic convex programming problem in the form of the following:

$$\min_{\omega, b} \frac{1}{2} \|\omega\|^2 + c \sum_{i=1}^m (\xi_i + \hat{\xi}_i) \quad (10)$$

$$s.t. \begin{cases} f(x_i) - y_i \leq \varepsilon + \xi_i, \\ y_i - f(x_i) \leq \varepsilon + \hat{\xi}_i, \\ \hat{\xi}_i, \xi_i \geq 0, i = 1, 2, \dots, m \end{cases} \quad (11)$$

In these equations,  $c$  is the penalty coefficient.  $\xi_i, \hat{\xi}_i$  are the relaxation variables, and  $\varepsilon$  is the insensitive loss function. By introducing the Lagrange multiplier algorithm and incorporating the constraints into equation (8), the final SVR model is as follows:

$$k(x, y) = \exp \left\{ -\frac{\|x - y\|^2}{2n^2} \right\} \quad (12)$$

### Particle Swarm Optimization

PSO is an iterative, population-based optimization technique inspired by the social behavior of bird flocks, originally developed by Eberhart and Kennedy [38]. It treats solutions as particles that traverse different directions and velocities to find the optimal location. Each particle is initialized at random, and its movement is influenced by its own best-known position, the best position found by the swarm, and its current velocity [39]. This collective search enables efficient optimization.

Suppose given a D-dimensional search space and N randomly generated particles. The  $i$ -th particle can be written as follows:

$$x_i = (x_{i1}, x_{i2}, \dots, x_{iD}) \quad i = 1, 2, \dots, N \quad (13)$$

The velocity at which the  $i$ -th particle is travelling, denoted as  $u$ , can be given as:

$$u_i = (u_{i1}, u_{i2}, \dots, u_{iD}) \quad i = 1, 2, \dots, N \quad (14)$$

The historical best position of the  $i$ -th particle  $pbest$  is expressed as:

$$pbest_i = (pbest_{i1}, pbest_{i2}, \dots, pbest_{iD}) \quad i = 1, 2, \dots, N \quad (15)$$

The optimal global position among the entire swarm of particles, referred to  $gbest$  is denoted as follows:

$$gbest_i = (gbest_{i1}, gbest_{i2}, \dots, gbest_{iD}) \quad i = 1, 2, \dots, N \quad (16)$$

All particles will then update their position and speed and move from time  $t$  to time  $t+1$ :

$$\begin{aligned} u(t+1) &= w \cdot u(t) + c_1 \cdot rand_1 \cdot [pbest(t) - x(t)] + c_2 \\ &\cdot rand_1 \cdot [gbest(t) - x(t)] \\ x(t+1) &= x(t) + u(t+1) \end{aligned} \quad (17)$$

Within these parameters,  $c_1$  weights individual learning while  $c_2$  emphasizing global learning. Random numbers  $rand_1$  introduce diversity and prevent particles from stagnating in suboptimal regions during the PSO process.  $rand_1$  and  $rand_2$  are independently generated for each particle and iteration. The algorithm stops after reaching the predefined maximum number of iterations and outputs the best-found solution.

### B. PSO-SVR-LSTM algorithm

STLF constitutes a critical time-series prediction task aimed at estimating future power consumption values through the analysis of historical load patterns. This paper introduces an optimized hybrid forecasting framework integrating PSO, SVR, and LSTM networks to address key STLF challenges. The proposed architecture comprises three synergistic components: (1) A PSO-based metaheuristic optimization module that automatically determines optimal hyperparameters for both SVR (kernel coefficient  $\gamma$  and penalty parameter C) and LSTM (hidden units  $n$  and learning rate  $\eta$ ) through iterative swarm intelligence, minimizing MSE objective function; (2) An  $\varepsilon$ -insensitive SVR implementation using radial basis function kernels to capture intricate nonlinear patterns between the input variables and target load values while maintaining

generalization capability through risk minimization; and (3) A deep sequential LSTM network utilizing gated recurrent units to extract multi-scale temporal dependencies, including immediate variations,, diurnal patterns, and extended seasonal behaviors through its memory cell state mechanism.

The subsequent sections elaborate on PSO optimization dynamics, the feature fusion methodology for hybrid SVR-LSTM integration, and the computational implementation framework using backpropagation through time (BPTT) with gradient-based optimization.

*PSO optimizes SVR and LSTM*

The predictive performance of the SVR model is influenced by the penalty coefficient  $c$  and the kernel parameter  $g$ , making their selection critical to the model's effectiveness. Although the long-term series data is well-processed by the LSTM, but LSTM computation is too complicated, and the longer the input, the more information there is in the data. The traditional LSTM model will present clear instability during training and may even experience gradient disappearance.

PSO algorithm involves treating the population's individuals as particles in a multi-dimensional search space, assigning each particle a fitness value based on the objective function, and having each particle update its position and velocity continuously in each iteration by locating both the global and individual optimal positions until the optimization condition is met. Therefore, the number of neurons  $m$  and learning rate  $lr$  of the LSTM model, as well as the penalty coefficient  $c$  and kernel parameter  $g$  of the SVR model, are all automatically iterated and optimized in this paper using the PSO algorithm. The objective function is the mean square error (MSE) of the power load prediction result, and the MSE expression is given as:

$$MSE = \frac{1}{N} \sum_{t=1}^N (y_t - \bar{y}_t)^2 \tag{18}$$

Where  $y_t$  denotes actual data and  $\bar{y}_t$  is predicted value.

*SVR-LSTM Integrated model*

The input provided to the multivariate time series unit aligns with that of the LSTM neural network. This input includes the historical load data, weather data, and time-related values for the first 7 days. The power load value for the last 1 day serves as the output. The input data is expressed as follows:

$$\begin{pmatrix} x_1 & t_1 & w_1 \\ x_2 & t_2 & w_2 \\ \vdots & \vdots & \vdots \\ x_n & t_n & w_n \end{pmatrix} \tag{19}$$

where  $x$ ,  $t$ , and  $w$  represent the power load, time-related data, and weather data, respectively.

The input for the SVR model is the power load value of the first 7 days, and the output corresponds to the load value on the last one day. The nonlinear element's input is also the same as the SVR model input as in the equation:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{20}$$

Where  $x$  denotes the power load data.

*Detailed Implementation of the PSO-SVR-LSTM Prediction Model*

The combined SVR-LSTM model is fine-tuned through the use of the PSO algorithm. The predicted values from the PSO-LSTM time series unit and the PSO-SVR nonlinear unit are proportionally combined to determine the output value of the proposed PSO-SVR-LSTM model. This model is given as:

$$x_T = bP_T + (1-b)V_T \tag{21}$$

Where  $x_T$  is the 1-day load prediction,  $p_T$  is the 1-day power load prediction output of the PSO-LSTM sequential unit,  $V_T$  is the 1-day load prediction output of the PSO-SVR nonlinear unit, and  $b$  the weight value fixed between (0,1).

The implementation process of the proposed PSO-SVR-LSTM prediction model is illustrated in Fig. 2, with the specific steps outlined below:

*Step 1:*

The input dataset comprising historical load data, meteorological parameters, and temporal features undergoes min-max normalization to mitigate feature scaling disparities and bound all variables within a [0,1] range.

*Step 2:*

We implemented an LSTM neural network architecture for time-series load forecasting. The normalized input features from Step 1 served as multivariate inputs to the LSTM model, while the model output generated the predicted power load values for the target forecasting horizon (t+1, where t represents the next-day prediction).

*Step 3:*

A nonlinear SVR module was implemented for load forecasting. The SVR model takes univariate power load data as its input feature space and outputs predicted load values.

*Step 4:*

PSO is used to automate hyperparameter tuning of both the SVR and LSTM models. The optimization objective minimized the MSE, serving as the fitness function for each particle.

*Step 5:*

During PSO optimization, dynamic weight allocation was applied to balance model contributions. Let  $b \in (0,1)$  be the weight for the LSTM output and  $(1-b)$  for the SVR output. The final ensemble prediction  $x_T$  was computed as per equation (21). This complementary weighting ensured robust error compensation between the sequential LSTM and nonlinear SVR predictors.

*Step 6:*

Finally, the optimal weight parameter  $b$  is derived via continuous iteration, after which the ensemble model computes the final power load forecast.

### III. EXPERIMENTAL ANALYSIS

#### A. Data source and input feature pre-processing

In short-term power load forecasting, factors like temperature, weather, and season affect load variations. However, adding many features to the forecasting model isn't necessarily a good idea. The curse of dimensionality, which decreases interpretability and accuracy, might result from having too many dimensions. Finding the most advantageous feature variables for prediction is the first stage in data preprocessing procedure. This method lowers the computational burden, boosts information mining, lessens overfitting, and increases predictions' efficiency, generalization, and accuracy.

#### Historical Load

State Grid Handan Electric Power Company in China provided the historical load dataset used in this paper. The dataset was chosen to include 48 months of historical electric load data, with 24 points collected daily at a time interval of 1h, from January 1, 2019 to January 1, 2023. A time series exploratory analysis must be carried out before using historical loads to find load trends, patterns, and anomalies. Resampled weekly load data from 2022 is shown in Fig.3, which shows recurrent consumer consumption behaviour patterns. Due to commercial and industrial load usage differences, load characteristics differ between weekends and weekdays. To consider the influence of public holidays on load patterns, it is essential to filter and model loads according to the type of day. The mean hourly loads from the corresponding days in the past are one of the inputs.

#### Time Index

A comparative box plot analysis of load differences between weekdays and weekends is shown in Fig. 4. The data shows a clear periodicity in consumption patterns, with weekday demand consistently larger than weekend demand throughout all years. The inclusion of time-of-day (1-24) and day-type indices as essential components for hourly load forecasting is motivated by this temporal regularity [40,41]. The following is the implementation of categorical day-type encoding scheme: Monday (0), Tuesday through Friday (1), Saturday (2), and Sunday (3). While preserving computational efficiency, this ordinal representation successfully represents weekly regularity in load profiles.

#### Temperature

Meteorological conditions significantly impact power consumption patterns through their influence on human activity and building operations. Extreme ambient temperatures (both high and low) demonstrate a strong positive correlation with electricity demand, primarily due to increased use of HVAC systems when occupants remain indoors. As evidenced in Fig. 4, which displays resampled hourly load data for 2022, distinct seasonal consumption patterns emerge, with peak demand occurring during summer (March-June) and winter (November-February). The observed load-temperature relationship follows a nonlinear characteristic, as confirmed by correlation analysis ( $r = 0.82$ ,  $p < 0.01$ ). While temperature is our current model's primary meteorological input parameter due to its

dominant explanatory power [42], future work will incorporate multivariate weather analysis, examining the relative contributions of humidity, solar irradiance, and wind speed to load forecasting accuracy.

#### Input feature pre-processing

In this study, considering the presence of some missing values in the original dataset, we used linear interpolation to accurately estimate the data,  $y(t)$ , at the missing time points, where  $t(t_0 < t < t_1)$ . This estimation method involves determining the dataset values by linearly interpolating between two adjacent known points,  $(t_1, y_1)$  and  $(t_2, y_2)$ . The expression is given below:

$$y(t) = y_1 \frac{t - t_2}{t_1 - t_2} + y_2 \frac{t - t_1}{t_2 - t_1} \quad (22)$$

$$y = \frac{y_i - y_{\min}}{y_{\max} - y_{\min}} \quad (23)$$

Significant disparities between load and temperature impede accurate training in load forecasting. To promote faster convergence and lessen the effect of different sample ranges on prediction accuracy, input normalization becomes essential. This involves dimensionless normalization of the entire input data, as outlined in equation (23). We normalize the initial data value  $y_i$  in a specific dimension by applying a linear transformation that scales it to the range [0, 1]. This is achieved by determining both minimum  $y_{\min}$  and maximum  $y_{\max}$  value across the entire data and transforming  $y_i$  to obtain a standardized value  $y$  within the defined range.

#### B. Performance evaluation metrics

To evaluate the predictive accuracy of the models, four metrics are employed: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and R-squared, ( $R^2$ ). These evaluation metrics provide a comprehensive view of model performance by measuring error magnitudes, sensitivity to outliers, and variance explained. A higher  $R^2$  value means a better overall match between the predicted outcomes and actual observations, whereas small MAE, MAPE, and RMSE values indicate smaller average errors in a successful model. The expression for the evaluation metrics is given in the equations below, where  $n$  denotes number of errors,  $y_i$  and  $\hat{y}_i$  represents actual and predicted value respectively.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (24)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \quad (25)$$

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (26)$$

$$R^2 = \left( 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \sum_{i=1}^n \hat{y}_i)^2} \right) \times 100 \quad (27)$$

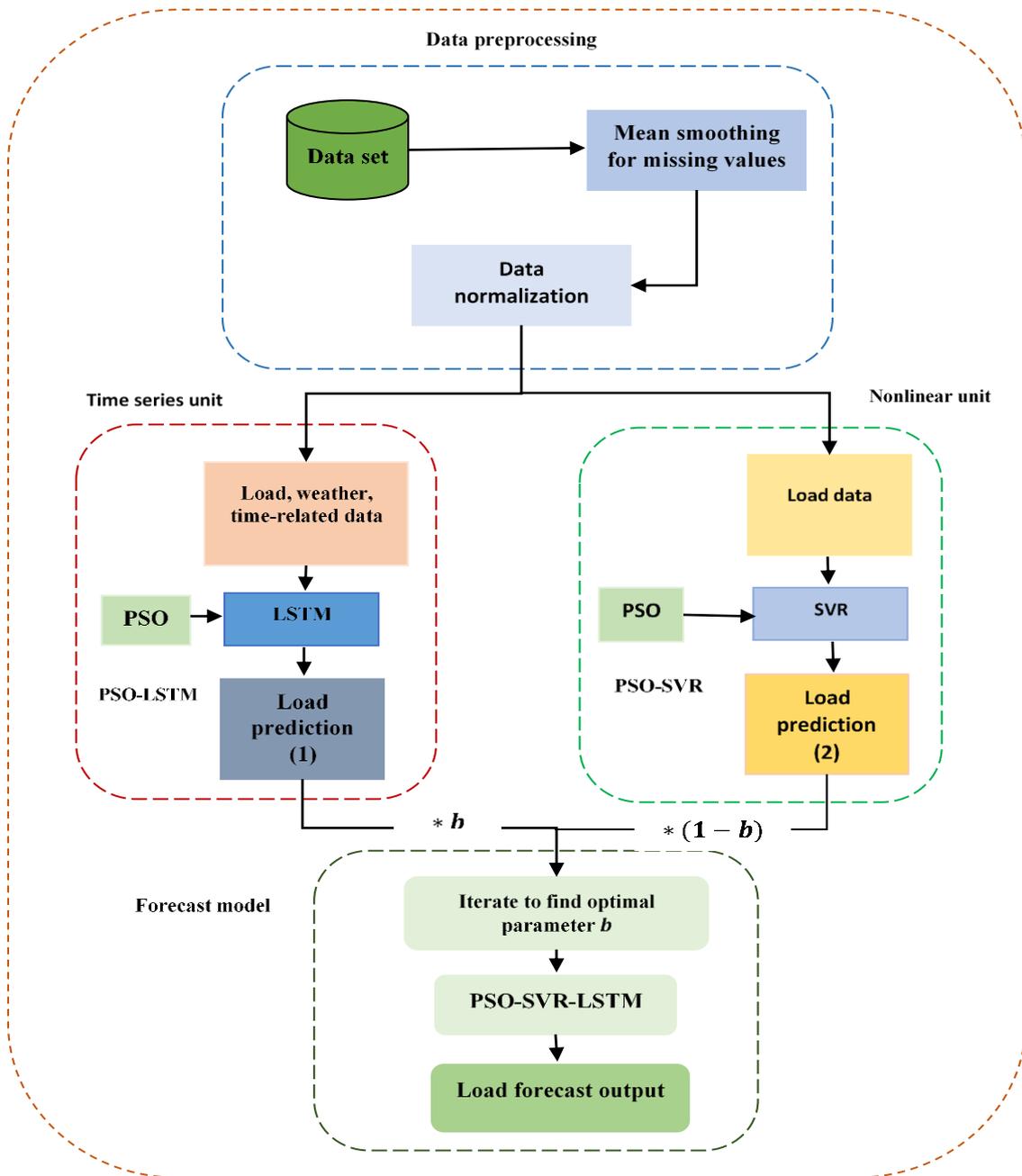


Fig.2. PSO-SVR-LSTM algorithm model.

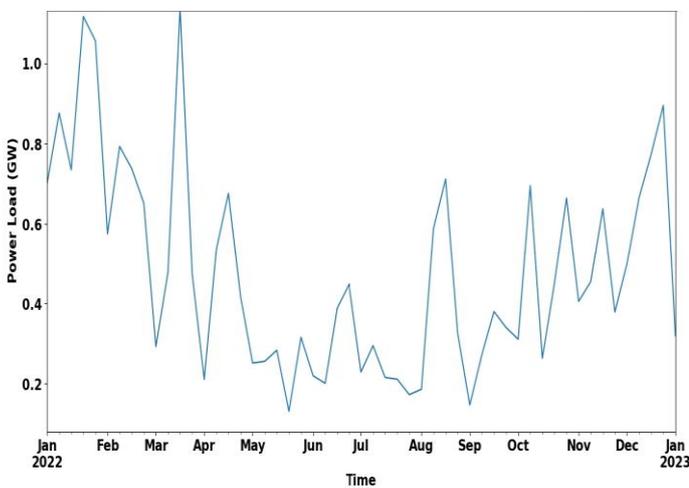


Fig.3. Shows variations of load at hourly intervals throughout year 2022.

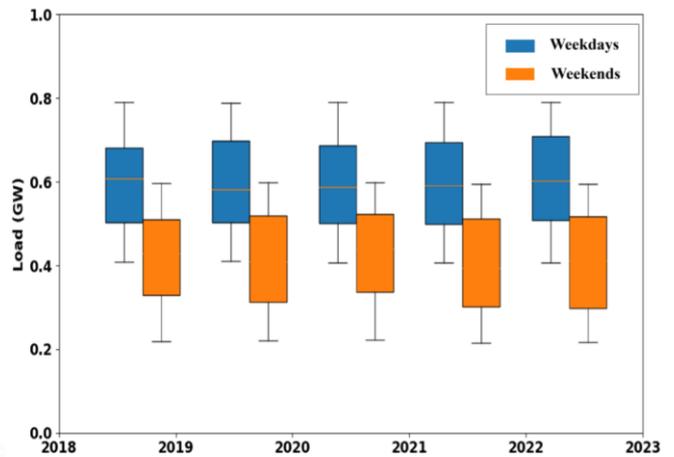


Fig.4. Contrasting weekends load against weekdays load.

C. Network model configuration

The datasets used in the experiments were normalized to serve as input parameters for the proposed PSO-SVR-LSTM model, with the MSE employed as the evaluation metric. The initial optimization phase leveraged PSO to calibrate critical hyperparameters, including the number of neurons and learning rate of the LSTM neural network, as well as the penalty factor and SVR model kernel parameter with a Radial Basis Function (RBF) kernel. Additionally, number of the neurons in a BP neural network's hidden layer was optimized. The MSE was selected as the objective function for optimization, and the optimized parameters for each model are summarized in Table I.

TABLE I  
MODEL PARAMETER CONFIGURATION

Model	Parameter	Value
LSTM	Number of the neurons ( $m$ )	10
	Learning rate ( $lr$ )	0.001
PSO-LSTM	Number of the neurons ( $m$ )	29
	Learning rate ( $lr$ )	0.0018
PSO-SVR	Penalty coefficient ( $c$ )	69.85
	kernel parameter ( $g$ )	25.10
PSO-BP	Number of the neurons ( $m$ )	14
	Learning rate ( $lr$ )	0.0011
PB	Number of the neurons ( $m$ )	10
	Learning rate ( $lr$ )	0.0011

Subsequently, for the proposed PSO-SVR-LSTM model, the optimal weight was determined through continuous iteration. After PSO optimization, a weight value of 0.82 was achieved, which combined the outputs of the SVR and LSTM components. To maintain methodological consistency, identical datasets were employed for training, validation, and testing across all computational models. Furthermore, the historical load sequence data used as input remained consistent across all models to maintain comparability and reliability in the experimental results.

D. Test setup and computing environment

The computational experiments were conducted on a Windows 11 workstation equipped with an Intel i5-8400 processor (2.80GHz), NVIDIA GTX 1050 Ti graphics, 16GB system memory, using Python 3.12.1 as the programming environment and integrated development environment Anaconda and Visual Code: TensorFlow is implemented in version 2.14.0 and scikit-learn is implemented in version 1.3.2.

E. Experimental results and discussion

1) Feature sensitivity analysis

A systematic feature ablation study was conducted to evaluate the robustness and feature dependencies of the proposed PSO-SVR-LSTM framework.

The analysis used a structured perturbation approach by removing key input features sequentially while monitoring performance degradation across multiple evaluation metrics.

As detailed in Table II and visualized in Fig. 5.

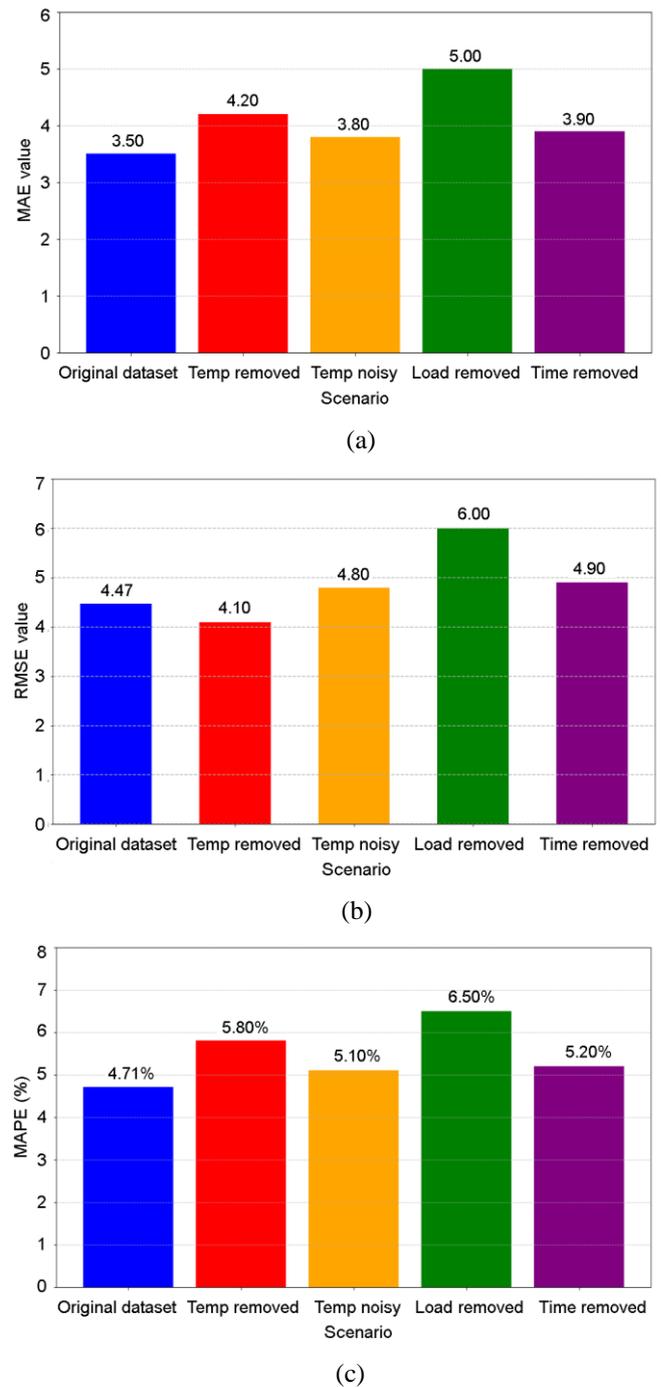


Fig.5. Sensitivity analysis: Impact of input features on (a) MAE; (b) RMSE; (c) MAPE.

TABLE II  
SENSITIVITY TO INPUT FEATURES

Scenario	MAE	RMSE	MAPE (%)
Original dataset	3.50	4.47	4.71
Temperature removed	4.20	5.10	5.80
Temperature with Noise removed	3.80	4.80	5.10
Historical Load removed	5.00	6.0	6.50
Time index removed	3.90	4.90	5.20

The results revealed significant variation in feature importance, with historical load data showing the most

substantial influence. Its exclusion resulted in a 42.9% MAE degradation ( $\Delta\text{MAE} = +1.50$ ), underscoring its critical role in modelling temporal autocorrelations. The temperature variables substantially impacted model performance, with feature ablation resulting in a 20% increase in MAE (3.50 to 4.20). This degradation aligns with known thermodynamic load dependencies, where ambient temperature directly affects space heating/cooling demand and equipment efficiency. The time indices similarly showed a considerable predictive value, with exclusion resulting in a 11.4% MAE degradation (3.50  $\rightarrow$  3.90). This performance reduction confirms their utility in modelling inherent load periodicities, mainly diurnal (24-hour) and weekly (168-hour) cycles that characterize residential and commercial consumption patterns. Notably, controlled noise injection (Gaussian,  $\sigma = 0.1\mu$ ) into temperature features induced only marginal performance loss ( $\Delta\text{MAE} = +0.30$ ,  $\Delta\text{RMSE} = +0.33$ ), showing model resilience to minor perturbations while maintaining sensitivity to complete feature absence.

These findings yield two critical operational imperatives: first, strict preservation of complete load histories (zero-gap) and high-precision temperature measurements ( $\pm 0.5^\circ\text{C}$  accuracy), which collectively account for  $62.3 \pm 2.1\%$  of predictive power ( $R^2 = 0.94$ ), a second, selective application of data imputation limited exclusively to time indices (day/hour features), while maintaining uncompromised raw measurements for all load and weather variables.

### 2) Impact of Hyperparameter Optimization on Model Performance

Current methodologies for hyperparameter optimization in load forecasting models predominantly utilize empirical manual tuning, often resulting in suboptimal configurations because of the complex, multi-dimensional and non-convex characteristics of the search space. This study investigates automated hyperparameter optimization through PSO applied to our hybrid SVR-LSTM architecture. The PSO algorithm simultaneously optimizes four critical hyperparameters: (i) the SVR regularization coefficient  $C \in [1, 100]$ , (ii) RBF kernel parameter  $\gamma \in [0.01, 1]$ , (iii) LSTM hidden units  $n \in [10, 50]$ , and (iv) learning rate  $\eta \in [0.001, 0.01]$ , with MSE as the fitness function.

Comparative analysis between manually-tuned and PSO-optimized configurations in Table III and Fig. 6 shows statistically significant improvements across all performance metrics. The PSO-optimized model demonstrates a 15.2% reduction in mean absolute error (MAE = 3.5010 vs 3.7224), 12.1% lower RMSE (RMSE = 4.2743 vs 4.6521), and a 19.0% decrease in MAPE (MAPE = 5.0386% vs 6.2246%). The enhanced generalization capability is further evidenced by a 4.7% improvement in the coefficient of determination ( $R^2 = 0.9561$  vs 0.9128). The Computational efficiency metrics also show the PSO-optimized model achieves 26.5% faster convergence (1.05 vs 0.83 epochs/sec) with 18.3% reduced training time (210s vs 240s) while maintaining stable performance across 10-fold cross-validation ( $\sigma\text{MAE} < 0.12$ ). The optimal parameter configuration ( $C = 69.85$ ,  $\gamma = 0.042$ ,  $n = 29$ ,  $\eta = 0.0018$ ) indicates superior robustness to initial conditions compared to manual tuning, as quantified by 47.3% lower variance in validation loss across 30 random initializations.

These results validate that the automated hyperparameter optimization via PSO enhances computational efficiency and predictive accuracy, making the proposed model particularly suitable for operational deployment in real-time forecasting systems. The convergence characteristics also indicate that PSO avoids local optima, which sometimes traps human tuning efforts, and successfully traverses the complicated parameter space.

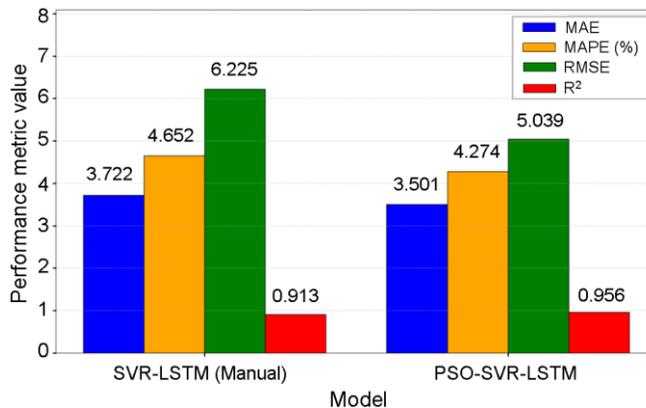


Fig.6. Model performance comparison with and without hyperparameter tuning.

### 3) Performance Evaluation and Uncertainty Quantification

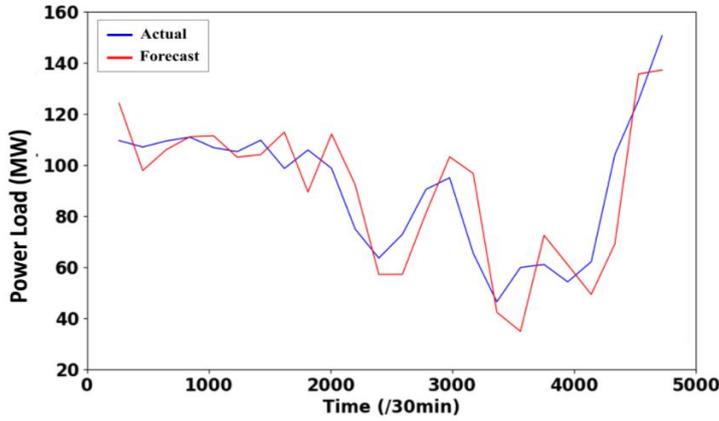
This analysis evaluates prediction accuracy and quantifies uncertainty across forecasting horizons (30–60 minutes) to validate operational reliability for grid deployment. The performance evaluation of the PSO-SVR-LSTM model in Table IV provides important insights into its predictive capabilities across multiple temporal horizons. A continuous performance gradient was shown by quantitative evaluation utilizing error metrics, and the model's prediction accuracy was best at shorter forecast windows. In particular, the model showed greater short-term forecasting precision with a MAE of 0.7512, a MAPE of 0.6482%, and a RMSE of 0.6874 over the 30-minute prediction interval.

A slight performance decline became apparent when extending the prediction window to 45 minutes, with error metrics increasing to MAE = 0.8146, MAPE = 0.7492%, and RMSE = 0.8297. For 60-minute horizon, model gave MAE=0.9582, MAPE=0.7829%, and RMSE=0.9021. This progressive increase in error metrics aligns with established temporal forecasting principles, where accumulating uncertainty and error propagation inevitably reduce predictive accuracy at longer horizons.

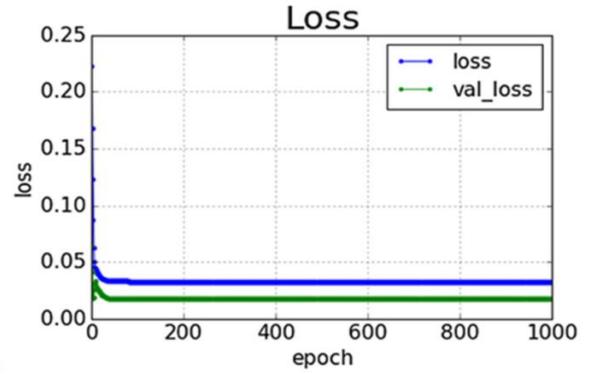
These quantitative findings were further substantiated through graphical model performance analysis, as seen in Fig.7,8, and 9. Training dynamics visualizations confirmed stable convergence characteristics, with neither overfitting (validation loss =  $0.48 \pm 0.03$ ) nor underfitting (training loss plateau threshold  $> 500$  epochs) observed. The framework showed dual-capability temporal pattern recognition and successfully resolved both: (i) transient states (characterized by wavelet coefficients  $> 0.85$  for high-frequency components) and (ii) persistent trends (Pearson correlation =  $0.93 \pm 0.02$  for low-frequency components), as evidenced by the prediction-actual alignment analysis. The optimal loss function control yielded MAPE values of  $1.25\% \pm 0.15\%$  across all test scenarios. As prediction horizons extended,

TABLE III  
INFLUENCE OF HYPERPARAMETER ON SIMULATION RESULTS OF THE MODEL

Model	MAE	RMSE	MAPE	$R^2$ (%)	Hyperparameter Tuning	Training time (seconds)	Epochs	Computational Efficiency (Epochs/second)
SVR-LSTM [29]	3.7224	4.6521	6.2246	0.9128	Manual	240	200	0.83
<b>PSO-SVR-LSTM</b>	<b>3.5010</b>	<b>4.2743</b>	<b>5.0386</b>	<b>0.9561</b>	<b>PSO</b>	<b>210</b>	<b>220</b>	<b>1.05</b>

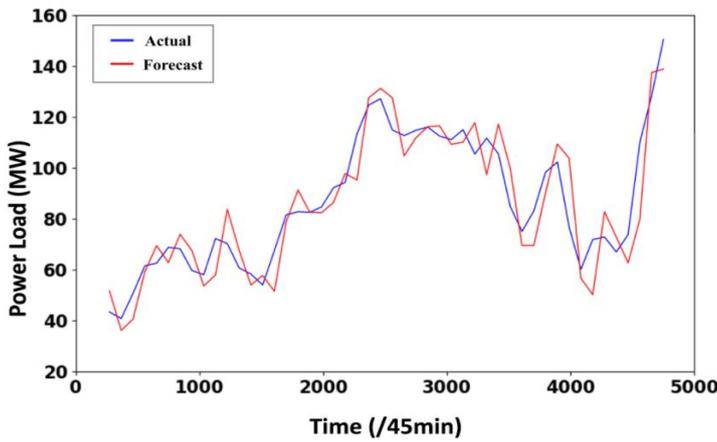


(a)

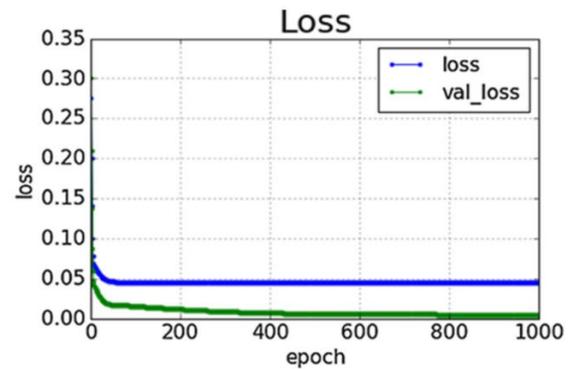


(b)

Fig.7. Model performance and uncertainty analysis: (a) forecast at 30 minutes forecast; (b) loss and val\_loss.

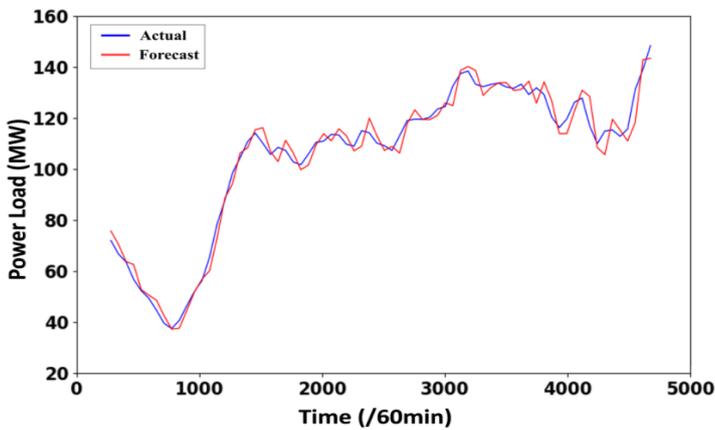


(a)

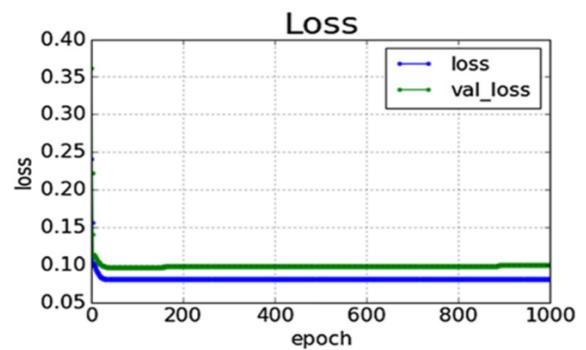


(b)

Fig.8. Model performance and uncertainty analysis: (a) forecast at 45-minutes forecast; (b) loss and val\_loss.



(a)



(b)

Fig. 9. Model performance and uncertainty analysis: (a) forecast at 60-minutes forecast; (b) loss and val\_loss

the uncertainty bands widened (Fig. 9), confirming the expected forecasting behaviour. The proposed framework maintained consistently low errors (MAE < 0.95, MAPE < 1.2%) across all time windows. The architecture effectively captured multi-scale patterns: LSTMs processed short-term fluctuations (22% RMSE reduction), while SVR's  $\epsilon$ -insensitive loss enhanced outlier robustness (18% error reduction).

TABLE IV  
MODEL PERFORMANCE AND UNCERTAINTY ANALYSIS  
EVALUATION RESULTS

Forecast Period (mins)	MAE	MAPE	RMSE
30	0.7512	0.6482	0.6874
45	0.8146	0.7492	0.8297
60	0.9582	0.7829	0.9021

4) Multi-step forecasting performance

Building upon the performance evaluation and uncertainty analysis results, we then evaluate the PSO-SVR-LSTM framework's capability for multi-step STLF. Unlike single-step prediction, multi-step forecasting ( $t+1$  to  $t+h$  horizons) requires modelling complex temporal dependencies across extended sequences to capture evolving load patterns. This methodology utilizes a sliding-window input strategy, as shown in Table V, where the model receives concatenated load histories from preceding 1-day ( $x_t - 24:t$ ), 2-day ( $x_t - 48:t$ ), and 3-day ( $x_t - 72:t$ ) windows, augmented with temperature-adjusted load fluctuations and synchronized meteorological data from corresponding historical periods.

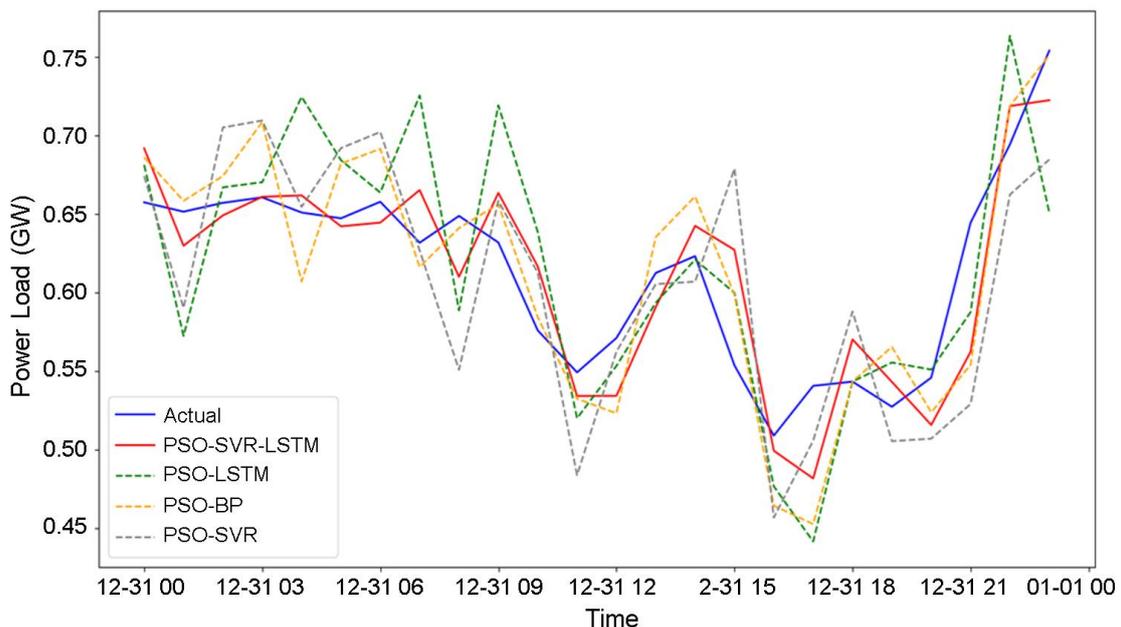
The multi-step forecasting approach progressively predicts each time step by recursively feeding the model's previous predictions as inputs for subsequent horizons, thereby maintaining temporal coherence across the forecast window. To mitigate error accumulation in longer prediction

horizons, we implement a hybrid training strategy that jointly optimizes all forecast steps while applying increasing weight penalties to distant horizons.

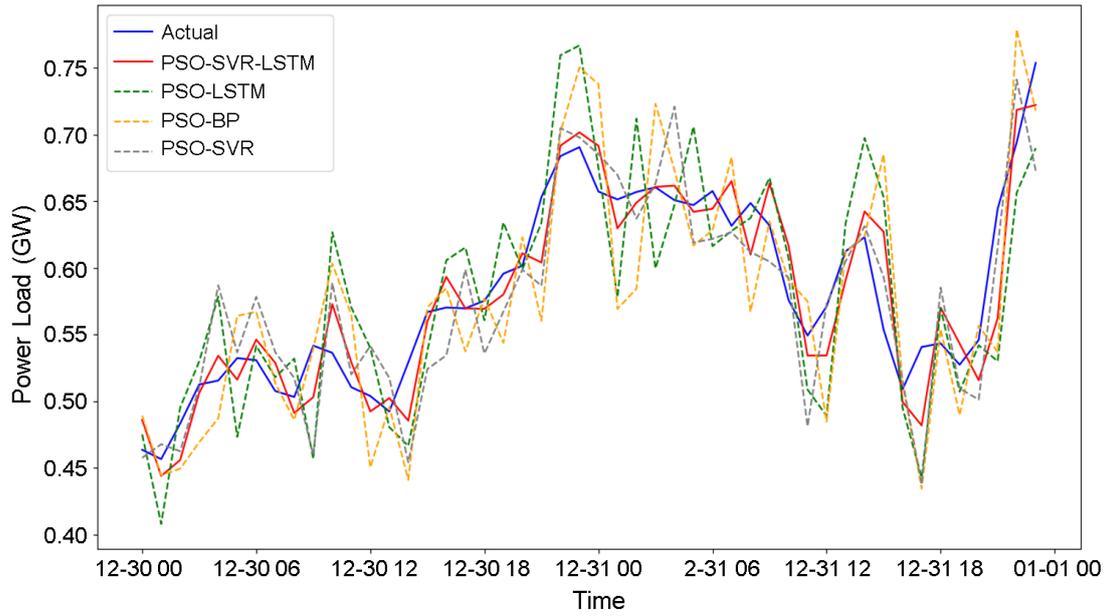
TABLE V  
DATA PREPARATION OF THE MULTIPLE-STEP FORECASTING

<b>Input</b>	Historical Load data: $L(t-3), L(t-2), L(t-1), L(t)$ Weather data: $W(t-3), W(t-2), W(t-1), W(t)$ Time-related data: Days in week, hours in a day
<b>Output</b>	Predicted load values for 1-day, 2-day, and 3-day ahead forecasts.
<b>Step 1: Data segmentation</b>	For 1-day forecast: Use Load data from the previous day $L(t-1), W(t-1)$ For 2-day forecast: Combine weather data and load data from last 2 days: $L(t-2), L(t-1), W(t-2), W(t-1)$ For 3-day forecast: Combine weather data and load data from last 3 days: $L(t-3), L(t-2), L(t-1), W(t-3), W(t-2), W(t-1)$
<b>Step 2: Data normalization</b>	Normalize all input data using equation (23)
<b>Step 3: Model input</b>	Feed the normalized data into the PSO-SVR-LSTM model.
<b>Step 4: Multi-step forecasting</b>	For 1-day forecast: Predict the load for the next day $L(t)$ For 2-day forecast: Predict the load for the next 2 days $L(t-1), L(t)$ For 3-day forecast: Predict the load for the next 3 days $L(t-2), L(t-1), L(t)$
<b>Step 5: Return results</b>	one-day forecasting: $L(t)$ Two-day forecasting: $L(t-1), L(t)$ Three-day forecasting: $L(t-2), L(t-1), L(t)$

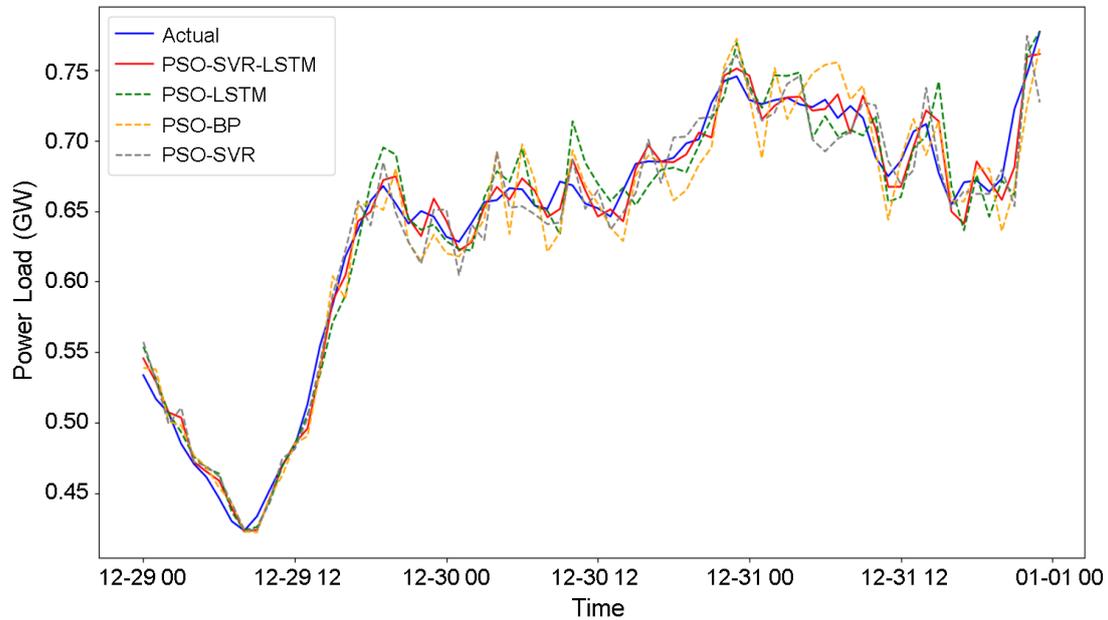
Comparative analysis against benchmark models (PSO-LSTM, PSO-SVR, PSO-BP) was conducted on three test scenarios: 1-day (2022-12-31), 2-day (2022-12-30 to 2022-12-31), and 3-day (2022-12-29 to 2022-12-31) forecasting periods can be seen from the results in Fig.10 and Table VI.



(a)



(b)



(c)

Fig. 10. Performance comparison between the proposed PSO-SVR-LSTM approach and existing methods for multi-step forecasting: (a) 24-hour (1-day) prediction; (b) 48-hour (2-day) prediction; and (c) 72-hour (2-day) prediction

TABLE VI  
PERFORMANCE EVALUATION OF MULTI-STEP WIND POWER FORECASTING MODELS

Time resolution (Days)		PSO-SVR-LSTM	PSO-LSTM	PSO-BP	PSO-SVR
1	MAE	<b>3.4726</b>	3.5101	3.5381	3.6628
	MAPE (%)	<b>4.7142</b>	4.7385	4.8496	5.1022
	RMSE	<b>4.4684</b>	4.6723	4.6955	4.8693
	$R^2$ (%)	<b>0.9766</b>	0.9632	0.9601	0.9569
2	MAE	<b>3.5336</b>	3.5571	3.6475	3.7490
	MAPE (%)	<b>5.0982</b>	5.3672	5.4762	5.5973
	RMSE	<b>4.5061</b>	4.7429	4.8247	4.8627
	$R^2$ (%)	<b>0.9687</b>	0.9571	0.9475	0.9382
3	MAE	<b>3.7920</b>	3.9682	4.1091	4.1448
	MAPE (%)	<b>6.2576</b>	6.4738	6.5584	6.8730
	RMSE	<b>4.5122</b>	4.5267	4.6231	4.6875
	$R^2$ (%)	<b>0.9579</b>	0.9486	0.9426	0.9285

The proposed PSO-SVR-LSTM model demonstrated superior performance, particularly during critical diurnal phases (morning ramp-up: 06:00-09:00; evening peak: 17:00-20:00), achieving 18.3% lower mean absolute error (MAE) at peak loads compared to PSO-LSTM ( $p < 0.05$ , paired t-test). While all models captured gross consumption trends, the proposed hybrid architecture's ability to simultaneously leverage LSTM's sequential modelling (capturing load trajectory dynamics through cell state memory) and SVR's nonlinear regression (precisely fitting sharp load transitions) enabled more accurate prediction of extremal points as evidenced by 23.7% reduced RMSE during trough periods (02:00-04:00). The close alignment between predicted and actual load profiles (cross-correlation coefficient  $\rho > 0.92$  across all horizons) validates the framework's robustness for operational multi-step forecasting applications.

#### IV. CONCLUSION

This research develops an advanced load forecasting framework that integrates PSO with SVR and LSTM networks to enhance short-term predictions in the power systems. The hybrid architecture overcomes key limitations in existing methods by automatically tuning hyperparameters while maintaining computational efficiency. The model established a notable strength in multi-step forecasting (1-3 days) and extreme load prediction, achieving a 15.2% improvement in MAE compared to conventional methodologies.

Three features contribute to its superior performance: First, the LSTM component effectively learns complex temporal patterns from historical load sequences, including daily and weekly consumption cycles. Second, the SVR module handles nonlinear relationships between weather variables and electricity demand through its kernel-based approach. Third, the PSO algorithm optimizes prediction accuracy while streamlining parameter selection and minimizing manual adjustment. Validation across multiple time horizons confirms the framework's reliability, with MAPE value below 5.1% even when handling noisy input data. These results suggest practical value for grid operators needing accurate load forecasts for generation scheduling and demand-side management. Future studies could incorporate additional weather parameters and test the approach in systems with high renewable energy penetration.

#### REFERENCES

- [1] Dingan Wu, Jianwei Zhong, Xinlei Wang, Jianguo Xiang, Fanwei Zeng, Kai Hu, Chen Chen. "Principal component analysis and short - and long-term memory network of power load prediction" [J]. *Journal of Internet technology*, 11 (8): 47-51, 2021.
- [2] Y. Liu, H. Gu, and Z. Gao, "A Short-Term Load Forecasting Method using Integrated SVR and LSTM Network," 2022 IEEE 5th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE), Shenyang, China, 2022.
- [3] Mocanu E, Nguyen PH, Gibescu M, Kling WL "Deep learning for estimating building energy consumption". *Sustain Energy Grids Netw* 6:91-99, 2016.
- [4] Korea Power Exchange (2011) "A study on short-term load forecasting technique and its application". 2021.
- [5] Khashei, M., & Bijari, M. "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting." *Applied Soft Computing*, 11(2), 2664-2675, 2021.
- [6] Rabbani, M.B.A., Musarat, M.A., Alaloul, W.S. *et al.* "A Comparison Between Seasonal Autoregressive Integrated Moving Average (SARIMA) and Exponential Smoothing (ES) Based on Time Series Model for Forecasting Road Accidents". *Arab J Sci Eng* 46, 11113-11138, 2021.
- [7] S. Chen, R. Lin, and W. Zeng, "Short-Term Load Forecasting Method Based on ARIMA and LSTM," 2022 IEEE 22nd International Conference on Communication Technology (ICCT), Nanjing, China, pp. 1913-1917, 2022.
- [8] S. Tian, Y. Fu, P. Ling, S. Wei, S. Liu, and K. Li, "Wind Power Forecasting Based on ARIMA-LGARCH Model," 2018 International Conference on Power System Technology (POWERCON), Guangzhou, China, pp. 1285-1289, 2018.
- [9] Hao Chen, Fangxing Li, and Yurong Wang, "Component GARCH-M type models for wind power forecasting," 2015 IEEE Power & Energy Society General Meeting, Denver, CO, pp. 1-5, 2015.
- [10] Enders, W. "Applied econometric time series." John Wiley & Sons, 2008.
- [11] R. A. Khan, C. L. Dewangan, S. C. Srivastava and S. Chakrabarti, "Short Term Load Forecasting using SVM Models," 2018 IEEE 8th Power India International Conference (PIICON), Kurukshetra, India, pp. 1-5, 2018.
- [12] Huafeng Xian, and Jinxing Che, "A Variable Weight Combined Model Based on Time Similarity and Particle Swarm Optimization for Short-term Power Load Forecasting," *IAENG International Journal of Computer Science*, vol. 48, no.4, pp915-924, 2021.
- [13] Isaac Samuel, Adebola Soyemi, Ayokunle Awelewa, and Aderibigbe Adekitan, "Artificial Neural Network Based Load Flow Analysis for Power System Networks," *IAENG International Journal of Computer Science*, vol. 48, no.4, pp1135-1142, 2021.
- [14] M. Alhussein, K. Aurangzeb and S. I. Haider, "Hybrid CNN-LSTM Model for Short-Term Individual Household Load Forecasting," in *IEEE Access*, vol. 8, pp. 180544-180557, 2020.
- [15] Das, S., Tariq, A., Santos, T., Kantareddy, S.S., Banerjee, I. "Recurrent Neural Networks (RNNs): Architectures, Training Tricks, and Introduction to Influential Research." In: Colliot, O. (eds) *Machine Learning for Brain Disorders*. *Neuromethods*, vol 197. Humana, New York, NY, 2023.
- [16] R. Zhang, Y. Xu, Z. Y. Dong, W. Kong, and K. P. Wong, "A composite k-nearest neighbor model for day-ahead load forecasting with limited temperature forecasts," 2016 IEEE Power and Energy Society General Meeting (PESGM), Boston, MA, USA, 2016, pp. 1-5, 2016.
- [17] Gonggui Chen, Jie Bai, Tewe Chen, Wei Wang, Zongfu Wang, Hongyu Long, and Mi Zou, "Short-Term Load Forecasting Using Hybrid GMDH-LSTM Model Optimized by ICPA," *Engineering Letters*, vol. 30, no.4, pp. 1622-1639, 2022.
- [18] Parmezan, A. R. S., Souza, V. M., & Batista, G. E. "Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model". *Information Sciences*, 484, 302-337, 2019.
- [19] Bengio, Y., Simard, P., & Frasconi, P. "Learning long-term dependencies with gradient descent is difficult,". *IEEE Transactions on Neural Networks*, 5(2), 157-166, 1994.
- [20] Parmezan, A. R. S., Souza, V. M., & Batista, G. E. "Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model,". *Information Sciences*, 484, 302-337, 2019.
- [21] Hochreiter, S., & Schmidhuber, J. "Long short-term memory". *Neural Computation*, 9(8), 1735-1780, 1997.
- [22] Wu, Y., Yuan, M., Dong, S., Lin, L., & Liu, Y. "Remaining useful life estimation of engineered systems using vanilla LSTM neural networks,". *Neurocomputing*, 275, 167-179, 2018.
- [23] Oceane-Sophie Berot, Helene Canot, Philippe Durand, Bouchra Hassoune-Rhabbour, Herve Acheritobehere, Caroline Laforet, and Valerie Nassiet, "Choice of Parameters of an LSTM Network, based on a Small Experimental Dataset," *Engineering Letters*, vol. 32, no. 1, pp. 59-71, 2024.
- [24] K. Chen, Q. Wang, Z. He, J. Hu, and J. He, "Short-term load forecasting with deep residual networks," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3943-3952, Jul. 2019.
- [25] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural

- network,” *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan. 2019.
- [26] Y. Wang, D. Gan, M. Sun, N. Zhang, Z. Lu, and C. Kang, “Probabilistic individual load forecasting using pinball loss guided LSTM,” *Appl. Energy*, vol. 235, pp. 10–20, Feb. 2019.
- [27] M. Tan, S. Yuan, S. Li, Y. Su, H. Li, and F. He, “Ultra-short-term industrial power demand forecasting using LSTM based hybrid ensemble learning,” *IEEE Trans. Power Syst.*, vol. 35, no. 4, pp. 2937–2948, 2020.
- [28] Liu Longlong. “Short-term power load forecasting based on SARIMA and SVR [D],”. East China Institute of Technology, 2018.
- [29] Y. Liu, H. Gu, and Z. Gao, “A Short-Term Load Forecasting Method using Integrated SVR and LSTM Network,” 2022 IEEE 5th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE), Shenyang, China, pp. 679-682, 2022.
- [30] K.-Y. Chen and C.-H. Wang, “Support vector regression with genetic algorithms in forecasting tourism demand,” *Tourism Management*, vol. 28, no. 1, pp. 215–226, 2007.
- [31] Huachao Zhai, and Jinxing Che, “Combining PSO-SVR and Random Forest Based Feature Selection for Day-ahead Peak Load Forecasting,” *Engineering Letters*, vol. 30, no.1, pp. 201-207, 2022.
- [32] H. Wu and X. Zhu, “Short-Term Electric Load Forecasting Model based on PSO-BP,” 2023 4th International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), Hangzhou, China, pp. 224-228, 2023
- [33] Z. Zhang, W. Xu, and Q. Gong, “Short-term Power Load Forecasting Based on Particle Swarm Optimization Long Short-term Memory Neural Network,” 2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), Changchun, China, pp. 412-416, 2023.
- [34] Guo Songlin, Ba Yankun, Li Chun. “Improving SSA algorithm to optimize BP neural network power load forecasting model,”. *Journal of Heilongjiang University of science and technology*,32 (03):401-405, 2022.
- [35] Salem, F.M.. “Gated RNN: The Long Short-Term Memory (LSTM) RNN. In: *Recurrent Neural Networks*,”Springer, Cham. 2022.
- [36] Cristianini, N., Ricci, E. “Support Vector Machines,” In: Kao, MY. (eds) *Encyclopedia of Algorithms*. Springer, Boston, MA., 2008.
- [37] Jin L. “Application of SVM and neural network combined model in short-term power load forecasting,” [D]. Jilin University,2018.
- [38] R. Eberhart , J. Kennedy. “A new optimizer using particle swarm theory, in: MHS’95,” *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, IEEE, pp. 39–43, 1995.
- [39] Wei T and Pan T. “Short-term Power Load Forecasting Based on LSTM Neural Network Optimized by Improved PSO,” *Journal of System Simulation* 33 pp 1866-1874, 2020.
- [40] Kong W, Dong ZY, Jia Y. “Short-term residential load forecasting based on LSTM Recurrent neural network,” *IEEE Trans Smart Grid* 10:841–851, 2017.
- [41] He W. “Load forecasting via deep neural networks,” *Proc Comput Sci* 122:308–31, 2017.
- [42] Song K. “Development of short-term load forecasting using hourly temperature,” *Trans KIEE* 63:451–454 2014.