



Article

# Electric Vehicle Charging Load Forecasting Based on K-Means++-GRU-KSVR

Renxue Shang and Yongjun Ma \*

School of Artificial Intelligence, Tianjin University of Science and Technology, Tianjin 300450, China;  
shangrenxue@mail.tust.edu.cn

\* Correspondence: yjma@tust.edu.cn

**Abstract:** The accurate short-term forecasting of an electric vehicle (EV) load is crucial for the reliable operation of a power grid and for effectively reducing energy consumption. Due to the fluctuations in EV charging loads, particularly the significant load variation between commercial and non-commercial areas, global models often suffer from prediction errors when forecasting loads. To address this issue, this paper proposes a regional forecasting method based on K-means++ clustering and deep learning algorithms. First, the K-means++ algorithm was used to partition the data into different regions, and an independent load-forecasting model was established for each region. Then, a combination of kernel support vector regression (KSVR) and gated recurrent unit (GRU) models was used to handle nonlinear features and time-dependent data, where particle swarm optimization (PSO) further optimized the model parameters to improve the forecasting accuracy. Finally, a weighted summation method was used to integrate the forecast results from each region, resulting in a more accurate overall load forecast. The experimental results show that the proposed model provided better prediction performance by capturing the spatiotemporal characteristics of the EV charging load, effectively addressing the challenges posed by regional differences, and outperforming the single-model forecasts.



**Citation:** Shang, R.; Ma, Y. Electric Vehicle Charging Load Forecasting Based on K-Means++-GRU-KSVR. *World Electr. Veh. J.* **2024**, *15*, 582. <https://doi.org/10.3390/wevj15120582>

Academic Editor: Michael Fowler

Received: 12 November 2024

Revised: 10 December 2024

Accepted: 13 December 2024

Published: 18 December 2024



**Copyright:** © 2024 by the authors. Published by MDPI on behalf of the World Electric Vehicle Association. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** electric vehicle power load; K-means++; gated recurrent unit; kernel support vector regression

## 1. Introduction

With the rapid expansion of the EV market, the prediction of EV charging loads has become increasingly important in power system management [1,2]. On one hand, centralized EV charging can lead to a sharp increase in the grid load, placing significant pressure on the power system [3,4]. If these load fluctuations are not accurately predicted, it can result in an imbalance between the power supply and demand, potentially leading to a system instability or power outages [5]. On the other hand, accurate load forecasting not only helps the grid better integrate renewable energy but also optimizes energy management and scheduling [6]. However, despite the continuous development of EV charging load prediction technologies, many challenges remain [7]. EV charging loads have significant spatiotemporal characteristics [8]. In the time dimension, charging loads are influenced by factors such as daily, seasonal, and holiday variations [9], whereas in the spatial dimension, the charging demand varies across regions, with significant differences between urban and suburban areas, as well as between commercial and non-commercial areas [10]. Therefore, how to simultaneously handle these spatiotemporal characteristics and build accurate spatiotemporal forecasting models is an urgent problem that needs to be addressed [11].

Traditional regression analysis, time series models (such as ARIMA [12] and SARIMA [13]), and statistical methods (such as MRA [14] and GLM [15]) have been widely applied to the spatiotemporal forecasting of EV charging loads. Reference [16] captured the regular changes in the charging load using time series methods, particularly the effects of

time dimensions, such as different seasons and holidays. The SARIMA model can handle seasonal fluctuations, enabling the model to better predict the trends in charging load fluctuations, especially in cases with periodic and seasonal patterns. Reference [17] used Monte Carlo simulations to predict charging loads and circuit trip-chain techniques effectively. A spatiotemporal model was developed to predict charging load demands under different scenarios, considering factors like traffic conditions and temperature. These traditional methods have some limitations when dealing with the spatiotemporal characteristics of EV charging load prediction, especially in handling nonlinearity, dynamic changes, and complex spatiotemporal structures.

As the complexity of microgrid load data increases, machine learning and deep learning methods have demonstrated strong capabilities in handling nonlinearity. Reference [18] proposed an electric vehicle charging load forecasting method based on Variational Mode Decomposition (VMD), Sparrow Search Algorithm (SSA), and Support Vector Regression (SVR), aiming to address the nonlinearity and complexity issues in electric vehicle load forecasting. Reference [19] proposes a spatiotemporal multi-graph convolutional network (ST-MGCN) based on the attention mechanism for predicting electric vehicle (EV) charging station loads. This method leverages the spatial and temporal dependencies of the surrounding network and improves the prediction accuracy by incorporating the attention mechanism, particularly when handling dynamic and heterogeneous traffic and charging patterns. The model focuses on relevant spatiotemporal factors through attention layers and adaptively selects key regions for load forecasting. This approach effectively captures and utilizes the complex patterns of the EV charging load, enhancing the prediction performance. Reference [20] combines multiple deep learning models, integrating CNN, LSTM, and Transformer models to perform deep learning modeling on time series data to further improve the prediction accuracy. These models automatically extract complex spatiotemporal features from the data without excessive manual intervention. Reference [21] proposed a model based on parallel structured spatiotemporal mutual residual graph convolution and bidirectional long short-term memory (CBi-LSTM) for predicting the charging demand of EVCS. This model fully considers the realistic topology of the EVCS network, external interference factors, and spatiotemporal dependencies. By constructing a mutual adjacency matrix, integrating graph convolution networks and CBi-LSTM, and adopting a parallel structure for prediction, it provides an effective solution for load forecasting of electric vehicle charging stations and efficient operation in the energy market. Reference [22] proposes an EV charging load prediction method based on spectral clustering and deep learning (SC-CNN-LSTM). This method addresses the variability and uncertainty issues in EV load prediction. Reference [23] presents a hybrid structure based on deep learning that combines CNN, GRU, attention mechanism (AM), and autoregressive (AR) models. This model not only effectively handles the time series data of EV charging loads but also provides more reliable load forecasts through probabilistic prediction methods, significantly improving the prediction accuracy.

The above references use deep learning techniques to solve the spatiotemporal characteristics of EV charging load forecasting, either individually or in combination. Temporal features are addressed through methods like LSTM or Recurrent Neural Networks (RNNs), while spatial differences are modeled using CNNs, Graph Convolutional Networks (GCNs), and other methods. These approaches effectively improve the accuracy of the predictions, especially in complex real-world environments, where they better adapt to the fluctuations and regional differences in the EV charging load. However, most deep learning methods, such as a CNN and LSTM, typically assume that the spatial characteristics of spatiotemporal data are uniform or can be learned through complex network structures. In reality, however, the charging demand varies significantly across regions, especially between commercial and non-commercial areas, where load fluctuations can differ greatly.

To solve the problem of charging load forecasting with significant regional differences, combining K-means++ clustering with deep learning algorithms for regional forecasting provides an effective solution for EV load prediction. This study made the following contributions:

- First, the K-means++ clustering method was used to partition different regions. Each region was modeled independently according to its specific spatiotemporal characteristics (such as the differences between commercial and non-commercial zones), which avoided the prediction errors that can occur in global models when there are significant regional differences. Each region's load pattern was analyzed, and a corresponding feature matrix was constructed.
- Independent models were trained for different regions to capture the unique characteristics of each dataset. By combining the advantages of KSVR in handling the nonlinear, high-dimensional data and the GRU in processing the time-dependent data, the GRU model could flexibly capture the time dependencies in the commercial area load data, especially in environments with large load fluctuations and complex time patterns. The GRU was effective in capturing the dynamic characteristics of the load changes. On the other hand, the KSVR model leveraged its strength in dealing with high-dimensional data and nonlinear problems, which made it suitable for the non-commercial area, where the load fluctuations were smaller and the data were more stable. By processing the time dependencies and nonlinear features of the two regions in parallel, we could ensure accuracy while reducing the computational complexity and training time. PSO was used to optimize the parameters of the KSVR and GRU, further improving the model's prediction performance.
- To integrate the predictions from different regions, a weighted summation method was employed. Specifically, the prediction results for each region were weighted based on their importance, and the overall load prediction was obtained through weighted summation. The weighted results of the hybrid model were compared with those from the individual models.

This paper is organized as follows: Section 1 provides an overview of the research background and a review of the existing literature on EV charging load forecasting. Section 2 outlines the research goals and the theoretical approaches employed for the short-term forecasting of EV charging loads. Section 3 details the processes of data preprocessing, feature selection, and the application of K-means++ clustering to the raw load data. In Section 4, data analysis is conducted, and various models are compared using specific case studies to assess the performance of the proposed forecasting method. Section 5 highlights the limitations of this study and offers suggestions for future research. Finally, the conclusion summarizes the key contributions of this work.

## 2. Materials and Methods

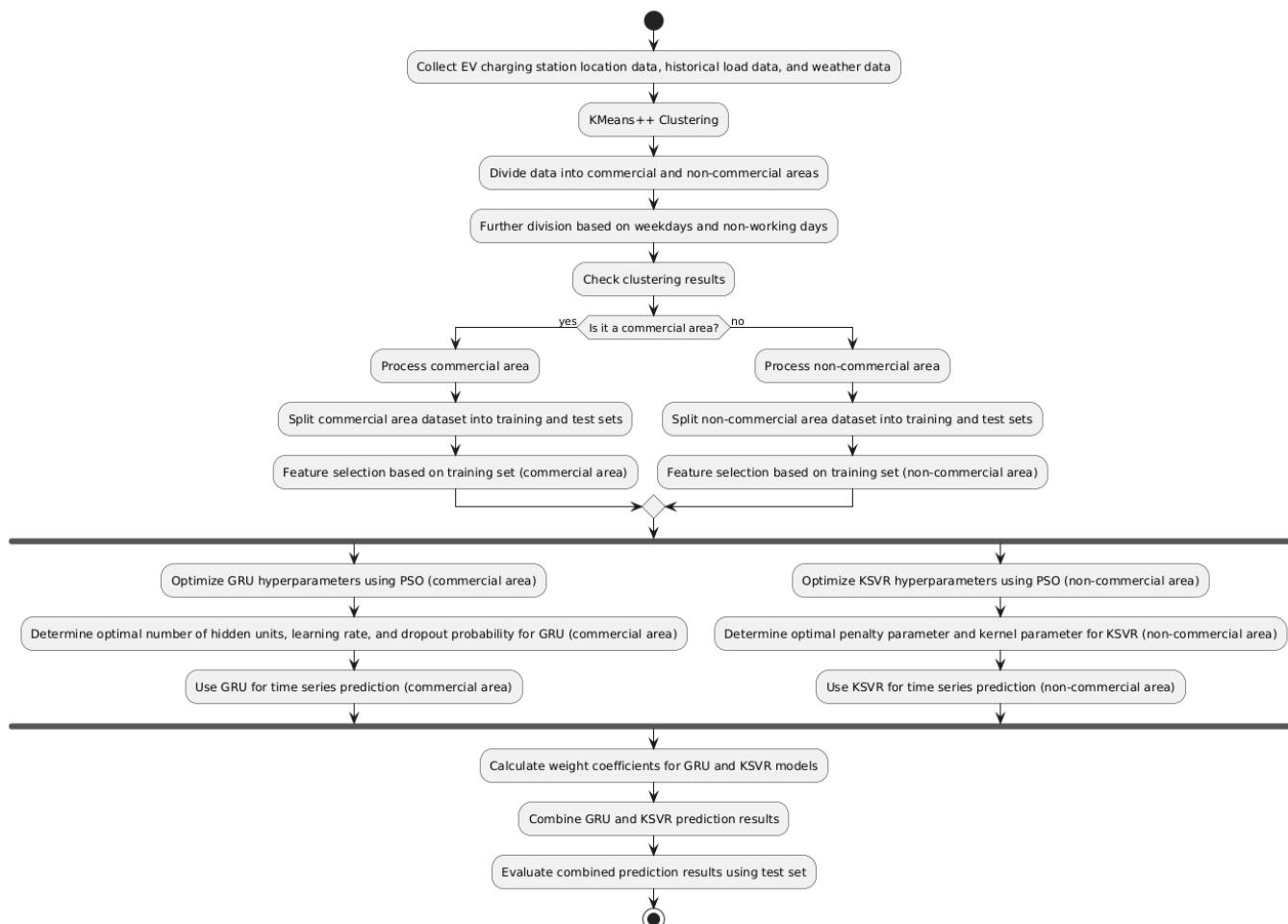
### 2.1. K-Means++-GRU-KSVR Load-Forecasting Model

The charging load of electric vehicles is influenced by various complex factors, among which traffic conditions and user behavior patterns are the key variables that directly impact the load's nonlinearity and regional variability. This paper proposes a K-means++-GRU-KSVR load-forecasting model, combining K-means++ clustering with a deep neural network GRU and KSVR to address these issues. The basic process framework is shown in Figure 1.

This study first utilized clustering analysis techniques to categorize the distribution data of the EV charging stations in the commercial and non-commercial areas of Shenzhen. Subsequently, the load data from these two areas were collected and analyzed to identify the load characteristics of each region. Through correlation analysis, features strongly correlated with the load in commercial and non-commercial areas were identified. This classification method helped capture the load patterns in different regions at different times and simplified the forecasting process to ensure the accurate capture of load variation patterns.

For each cluster of data, a parallel execution approach that combined a GRU and KSVR was employed for the time series forecasting. A GRU excels at capturing time-dependent patterns of power loads, especially in data with long-term dependencies, effectively handling load fluctuations over time. Simultaneously, KSVR, through kernel functions (such as a RBF kernel, linear kernel, or polynomial kernel), maps low-dimensional data into

a higher-dimensional space, efficiently capturing nonlinear relationships to improve the prediction accuracy. The parallel execution of a GRU and KSVR allows the GRU to focus on extracting temporal dependencies while KSVR simultaneously optimizes predictions for nonlinear relationships, significantly reducing the computation time and enhancing the overall forecasting efficiency.



**Figure 1.** The basic process framework of the K-means++-GRU-KSVR model.

To further improve the predictive performance, the PSO algorithm was used to optimize the parameters of both the GRU and KSVR. Based on these optimizations, the weight coefficients were calculated to construct an integrated GRU-KSVR forecasting model. This approach not only leverages the strengths of both algorithms but also enhances the computational efficiency through parallel execution, making it particularly suitable for handling the complex and dynamic characteristics of load-forecasting data.

## 2.2. K-Means++ Clustering Model

The K-means++ model is an enhanced version of the K-means clustering algorithm and was designed to improve the initialization of centroids, which can significantly affect the final clustering results. The K-means++ algorithm aims to partition a dataset into  $K$  clusters while minimizing the within-cluster sum of squares (WCSS), ensuring that the distance between the data points and their cluster centroids is as small as possible. This method is widely applied in various fields, including text analysis, image segmentation, and time series analysis. The main improvement in K-means++ lies in the way initial centroids are selected. Instead of randomly choosing initial centroids, K-means++ carefully selects them to spread them out across the data, leading to better a clustering performance

and faster convergence. This process involves initializing the centroids using a probability distribution proportional to the squared distance from the nearest already chosen centroid, followed by iterating until the centroids stabilize or the maximum number of iterations is reached [24,25].

The core goal remains minimizing the within-cluster sum of squares (WCSS):

$$\text{WCSS} = \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2 \quad (1)$$

where  $k$  is the number of clusters,  $S_i$  represents the set of data points in the  $i$ -th cluster,  $x$  is a data point, and  $\mu_i$  is the centroid of the  $i$ -th cluster. The initial centroids are chosen as follows: (1) The first centroid  $\mu_i$  is chosen randomly from the data points. (2) For each subsequent centroid, a data point  $x$  is chosen with a probability proportional to its squared distance from the nearest already chosen centroid:

$$P(x) = \frac{d(x)^2}{\sum_{x \in X} d(x)^2} \quad (2)$$

where  $d(x)$  is the distance between  $x$  and the nearest centroid already chosen.

After the initialization, the algorithm proceeds with the iterative k-means process, which includes assigning data points and updating centroids:

(1) Euclidean distance calculation: the Euclidean distance between a data point  $x$  and a centroid  $\mu_i$  is calculated as follows:

$$d(x, \mu_i) = \sqrt{\sum_{j=1}^n (x_j - \mu_{ij})^2} \quad (3)$$

where  $x_j$  is the  $j$ -th coordinate of data point  $x$ , and  $\mu_{ij}$  is the  $j$ -th coordinate of centroid  $\mu_i$ .

(2) Assignment of data points: each data point is assigned to the nearest cluster based on the minimum Euclidean distance:

$$S_i^{(t)} = \{x_p : \min_k d(x_p, \mu_k^{(t)})\} \quad (4)$$

where  $S_i^{(t)}$  is the set of points assigned to the  $i$ -th cluster at iteration  $t$ .

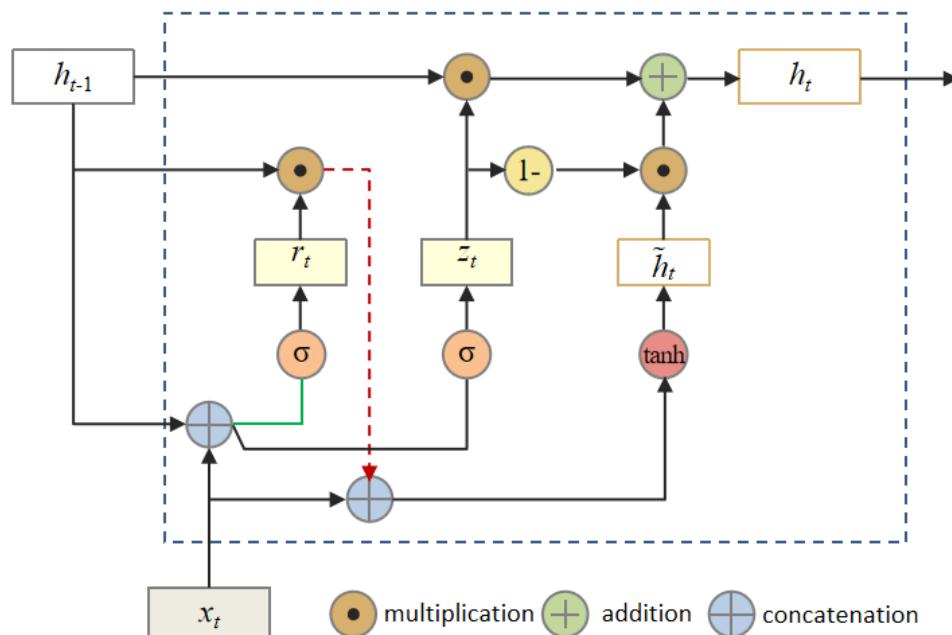
(3) Centroid update: the centroids are updated by calculating the mean of all points assigned to each cluster:

$$\mu_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x \in S_i^{(t)}} x \quad (5)$$

where  $|S_i^{(t)}|$  is the number of points in the  $i$ -th cluster at iteration  $t$ . The K-means++ initialization ensures a better starting point for clustering, reducing the likelihood of poor local minima and improving the convergence speed compared with the standard K-means algorithm.

### 2.3. Gated Recurrent Unit

A GRU is a simplified variant of an RNN designed to address the problems of gradient vanishing and gradient explosion in long-sequence data. Similar to LSTM networks, a GRU uses gating mechanisms to control the flow of information in the hidden state; however, it has a simpler structure, with only an update gate and a reset gate [26]. The internal structure of the GRU is shown in Figure 2.



**Figure 2.** Schematic diagram of GRU gate control structure.

The GRU employs gating mechanisms to control the flow of information. The key components are as follows [27].

The update gate determines how much of the previous hidden state should be retained. It is calculated as follows:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (6)$$

where  $z_t$  is the update gate at time  $t$ ,  $W_z$  is the weight matrix for the update gate,  $h_{t-1}$  is the previous hidden state,  $x_t$  is the current input, and  $\sigma$  is the sigmoid activation function.

The reset gate determines how much of the previous hidden state to forget. It is computed as follows:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (7)$$

where  $r_t$  is the reset gate at time  $t$  and  $W_r$  is the weight matrix for the reset gate.

The current memory content  $\tilde{h}_t$  is calculated using the reset gate:

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \odot h_{t-1}, x_t]) \quad (8)$$

where  $\tilde{h}_t$  is the candidate hidden state,  $W_h$  is the weight matrix for the hidden state,  $\odot$  represents the element-wise multiplication, and  $\tanh$  is the hyperbolic tangent activation function.

The final hidden state  $h_t$  is a combination of the previous hidden state and the current memory content:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (9)$$

where  $h_t$  is the hidden state at time  $t$ .

#### 2.4. Kernel Support Vector Regression (KSVR) Model

KSVR is an extension of support vector regression (SVR) that uses kernel functions to handle nonlinear relationships in the data by mapping input data into a higher-dimensional space. The goal of KSVR is to find an optimal hyperplane in this higher-dimensional space to minimize the prediction errors within a certain tolerance range ( $\epsilon$ -insensitive zone).

The primary objective of KSVR is to minimize the following function:

$$\min_{w,b} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \quad (10)$$

where  $w$  is the weight vector,  $b$  is the bias term,  $C$  is the penalty parameter controlling the trade-off between model complexity and training error, and  $\xi_i$  and  $\xi_i^*$  are slack variables that measure the deviation of the predictions from the actual target values.

KSVR employs an  $\epsilon$ -insensitive loss function to quantify the prediction errors [28]. The loss function is defined as follows:

$$L(\hat{y}_i, y_i) = \begin{cases} 0 & \text{if } |\hat{y}_i - y_i| \leq \epsilon \\ |\hat{y}_i - y_i| - \epsilon & \text{otherwise} \end{cases} \quad (11)$$

where  $\hat{y}_i$  is the predicted value from the model,  $y_i$  is the actual target value, and  $\epsilon$  is the specified margin of tolerance for the error.

The prediction function for KSVR is expressed as follows:

$$\hat{y}(x) = w^T \phi(x) + b \quad (12)$$

where  $\hat{y}(x)$  is the predicted value for the input data point  $x$ , and  $\phi(x)$  is the feature mapping function that transforms the input data into a higher-dimensional space through a kernel function.

The support vectors in KSVR are the samples that are most relevant to defining the decision boundary. These samples satisfy the following conditions [29]:

$$y_i - (w^T \phi(x_i) + b) \geq \epsilon \quad \text{or} \quad (w^T \phi(x_i) + b) - y_i \geq \epsilon \quad (13)$$

where  $y_i$  is the actual value of the training sample, and  $x_i$  is the feature vector of the training sample.

To solve the optimization problem in KSVR, the Lagrange multiplier method is used to reformulate the objective function into its dual form:

$$\max_{\alpha_i, \alpha_i^*} \quad \sum_{i=1}^n (\alpha_i - \alpha_i^*) - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(x_i, x_j) \quad (14)$$

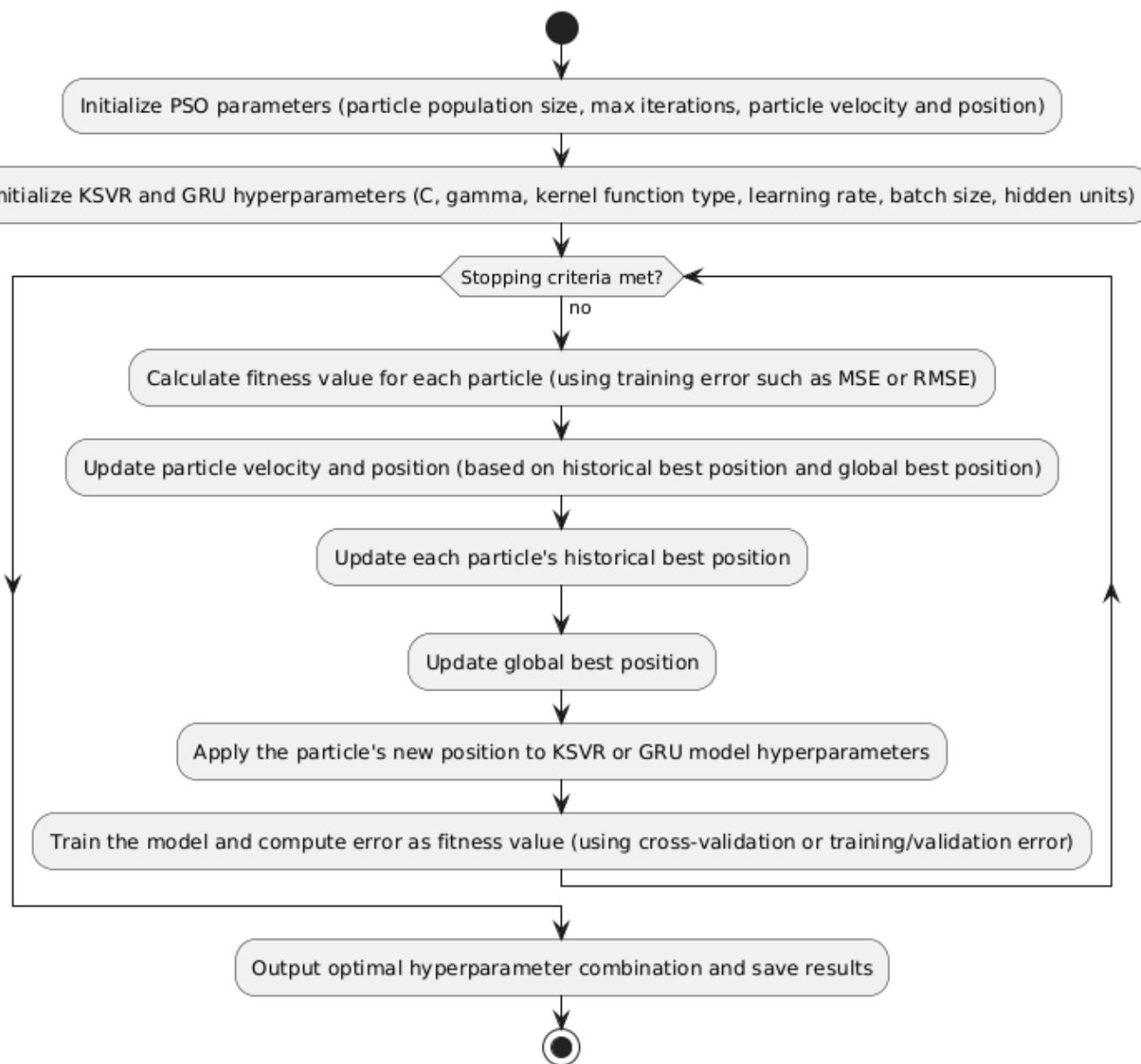
where  $\alpha_i$  and  $\alpha_i^*$  are the Lagrange multipliers corresponding to the constraints, and  $K(x_i, x_j)$  is the kernel function that computes the similarity between the input features. Common kernel functions include the radial basis function (RBF), polynomial kernel, and linear kernel. These kernel functions allow KSVR to operate in a high-dimensional space without explicitly transforming the data.

KSVR, with its ability to use kernel functions, is highly effective for nonlinear regression tasks, making it particularly useful for handling high-dimensional data and complex relationships in the input features. By leveraging the strengths of the SVM and kernel methods, KSVR provides a robust framework for making accurate predictions in various regression applications.

## 2.5. PSO Hyperparameter Optimization

PSO [30] is a swarm-intelligence-based optimization algorithm that solves optimization problems by simulating the foraging behavior of bird flocks. PSO is particularly well-suited for solving continuous optimization problems, as it iterates the position and velocity of each particle in the swarm to search for the optimal solution in the search space. In PSO, the optimization of hyperparameters is a crucial step, as these parameters control how the particles update their positions and velocities, which, in turn, affects the algorithm's convergence speed and global exploration capability.

When using a GRU, the parameters that primarily affect the performance include the number of hidden layers, number of GRU layers, dropout probability, learning rate, and optimizer. For KSVR, the key parameters affecting the performance include the penalty function and slack variables. A trade-off between the accuracy and model complexity must be considered when selecting these parameter values. To overcome the limitations of the broad search for optimal solutions, the PSO algorithm is employed to find the optimal parameters. The PSO hyperparameter optimization process is illustrated in Figure 3.



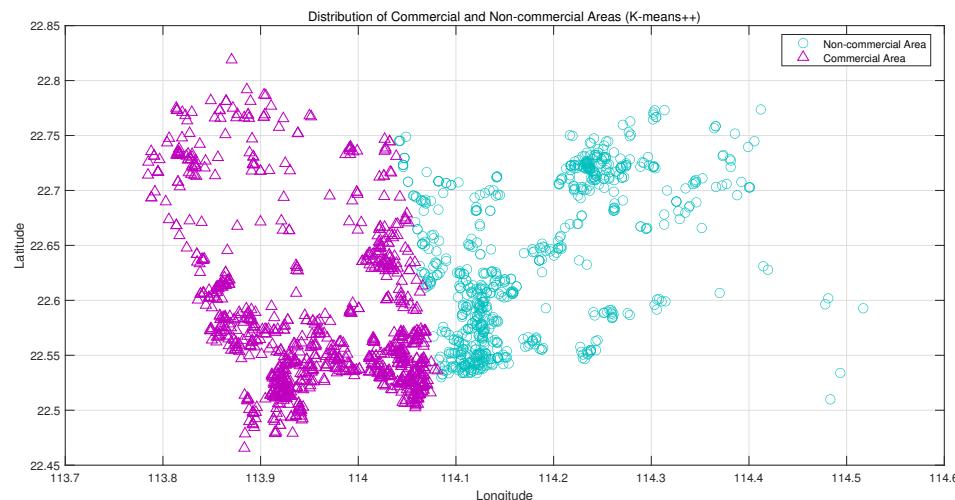
**Figure 3.** PSO hyperparameter optimization process.

### 3. K-Means++-Based GRU-KSVR Load Forecasting

#### 3.1. K-Means++ Clustering Analysis

The dataset used in this study was based on actual records from a specific area. The load-forecasting research was conducted using one year of sample load data and influencing factors from September 2022 to September 2023. The sampling interval between the data points was one hour, and the dataset included coordinate information for 1682 public EV charging stations. A preliminary analysis indicated that there were no

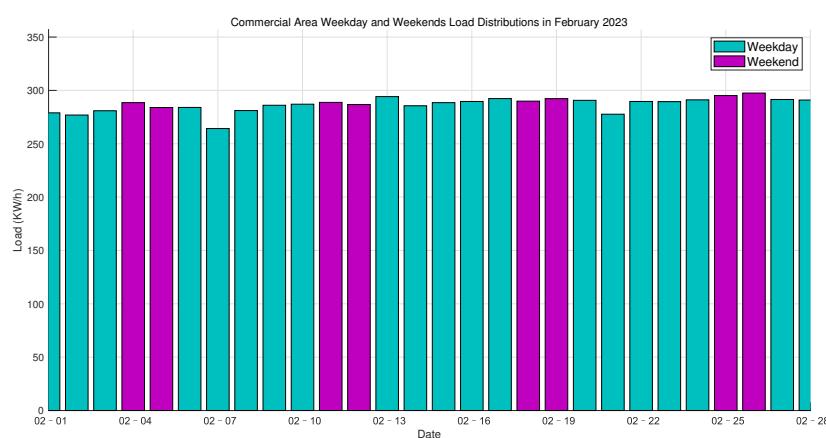
missing or duplicate values in the sampled time continuum. The distribution map of the commercial and non-commercial areas obtained after the clustering analysis is shown in Figure 4.



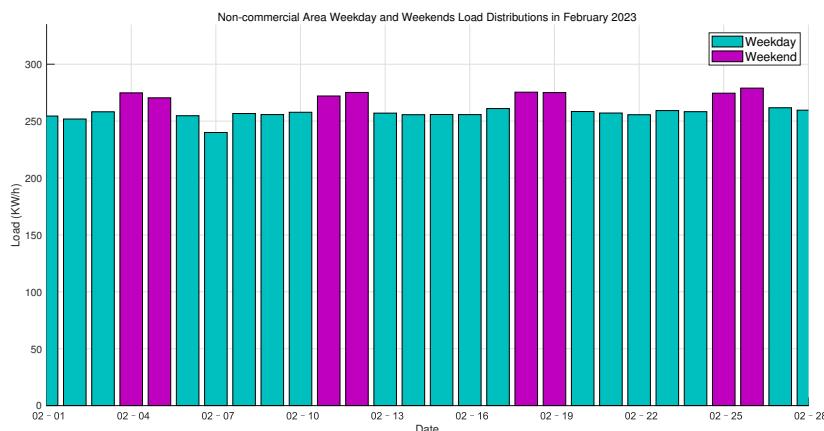
**Figure 4.** Clustering analysis results.

The results of the clustering analysis reveal significant differences in the distributions of the EV charging stations between the commercial and non-commercial areas. The commercial area contained 965 charging stations, which were typically located in the city's central business district. These areas were often densely populated, with frequent commercial activity and a high traffic flow, which led to a higher charging demand. The non-commercial area contained 717 charging stations and usually consisted of residential neighborhoods, suburbs, or industrial zones with relatively fewer people. Compared with the commercial area, the charging stations in the non-commercial areas were more widely dispersed.

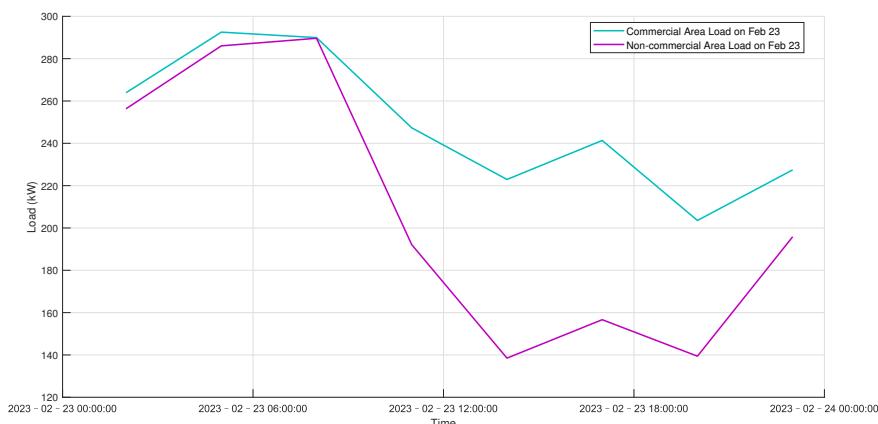
Further analysis was conducted on the load distribution differences between the commercial and non-commercial areas during weekdays and weekends, as shown in Figures 5 and 6. A typical day was selected to analyze the load distribution characteristics and differences between the two areas on weekdays and weekends, as shown in Figure 7.



**Figure 5.** Loads on weekdays and weekends in the commercial area in February.



**Figure 6.** Loads on weekdays and weekends in the non-commercial area in February.



**Figure 7.** Load variation curves for commercial and non-commercial areas on February 23rd.

A comparative analysis revealed that as the core area of the city, the commercial zone experienced a higher traffic flow, particularly during peak hours. This high vehicle mobility led to a greater charging load demand, with significant fluctuations in the load. Whether on weekdays or non-working days, the charging demand was concentrated during the morning (7–9 a.m.) and evening (5–7 p.m.) rush hours. In contrast, the load fluctuations in the non-commercial area were relatively small. Even on weekdays, the charging behavior in the non-commercial areas mainly occurred in the evening after the residents returned home. On the non-working days, the load was relatively higher, with a charging demand potentially spread throughout the day, especially during the night and early morning.

### 3.2. Influencing Factors Analysis

#### 3.2.1. Correlation Coefficient

The Spearman rank correlation coefficient is a non-parametric measure used to assess the relationship between two variables, describing the degree of monotonic correlation between them [31]. The value of the Spearman rank correlation coefficient ranges from  $-1$  to  $1$ , where  $1$  indicates a perfect positive correlation,  $-1$  indicates a perfect negative correlation, and  $0$  indicates no monotonic correlation. The Spearman correlation coefficient is commonly used to measure the relationship between two ranked variables or when the assumptions of linearity for the Pearson correlation are not met. The formula is as follows:

$$r_s = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \quad (15)$$

where  $r_s$  is the Spearman Rank Correlation Coefficient, which measures the monotonic relationship between two variables, with a value range of  $-1 \leq r_s \leq 1$ ;  $n$  is the number of data points, i.e., the number of observations for variables  $x$  and  $y$ ;  $d_i$  represents the rank difference for each pair of corresponding values, defined as  $d_i = R(x_i) - R(y_i)$ , where  $R(x_i)$  and  $R(y_i)$  are the ranks of the  $i$ -th values of variables  $x$  and  $y$ , respectively; and  $\sum d_i^2$  is the sum of the squared rank differences for all data points. The Spearman rank correlation coefficient is particularly useful for ordinal data or when the relationship between the variables is not strictly linear.

### 3.2.2. Correlation Analysis

An electric vehicle charging load is influenced by various factors. Through analysis, it was found that in addition to time-related factors, such as whether it was a weekday, other factors include the meteorological conditions and charging frequency. The meteorological data were sourced from a meteorological data statistics website and included the humidity, temperature, wind speed (Ff), visibility (VV), precipitation (nRAIN), Ground Atmospheric Pressure (P), and Sea Level Atmospheric Pressure (Po), as shown in Table 1.

**Table 1.** Meteorological influencing factors.

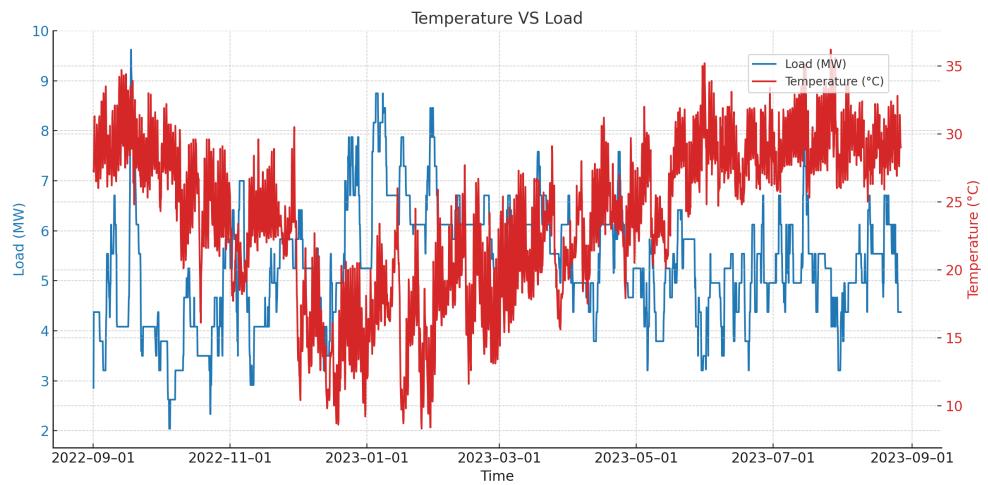
Variable Name	Unit
Temperature	°C
Po	Pascal
P	Pascal
Humidity	%
nRAIN	mm/h
Ff	m/s
VV	m

To gain a clearer understanding of the correlation between the meteorological factors and the EV charging load, Figure 8 shows the relationship between the temperature and load. Overall, there was a certain degree of negative correlation between the temperature and EV charging load. When the temperature was lower, the charging load tended to increase. This may have been because, under cold weather conditions, EVs require more energy to start and heat the interior, leading to more frequent charging. Additionally, in lower temperatures, EV owners may habitually choose to charge their vehicles after each use to prevent battery performance degradation, which could also contribute to the increase in the charging load. In warmer weather, the EV battery is less affected by temperature, and its range remains more stable, so users do not need to charge as frequently, which may lead to a decrease in the charging load.

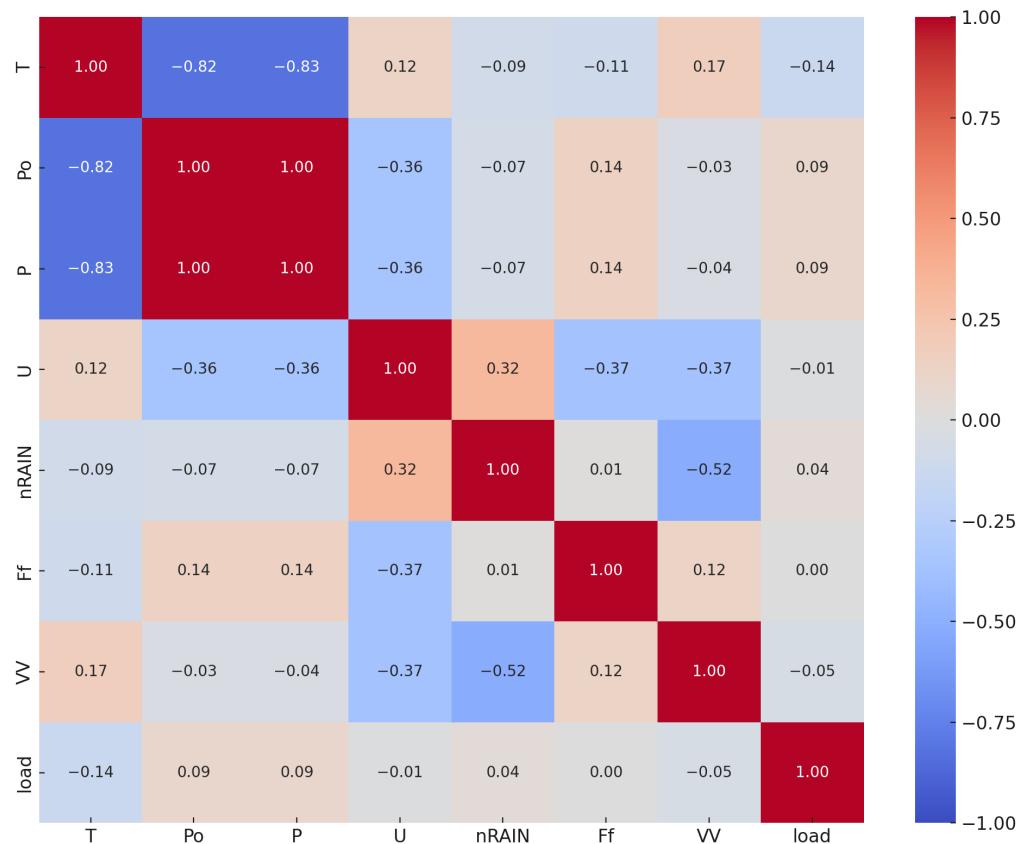
Due to the strong periodicity of EV charging frequency, the use of time characteristic factors can indirectly reflect the frequency demand for EV charging. The time characteristic factors extracted from the charging load data in the commercial and non-commercial areas mainly included date features: weekdays and weekends, and time features: hour (24 h format), peak hours (7–9 a.m., 5–7 p.m.), and off-peak hours (9 a.m.–5 p.m., 7 p.m.–12 a.m.).

A large number of feature variables increases the cost of model training and complicates the fitting process. Therefore, conducting a correlation analysis is crucial for selecting variables that are highly relevant to the model training. In this study, the Spearman rank correlation coefficient was used to determine the correlation between the EV charging load and meteorological factors, as shown in Figure 9, as well as the correlation between the commercial and non-commercial area loads under different time characteristics, as shown in Table 2. Since the charging demand in the non-commercial areas may lag behind that in commercial areas, cross-correlation analysis was used to determine whether there were lagged or synchronous changes in the load between the two areas during certain time periods. Considering the time-lag effects in the time series data, a time series sliding window

method was used to calculate the correlation between the commercial and non-commercial area loads at different lag times, with the results shown in Table 3.



**Figure 8.** Relationship between temperature and sample load.



**Figure 9.** Heat map of meteorological factor correlation.

**Table 2.** Correlation of load fluctuations between peak and off-peak areas.

Time Period	Workday	Non-Workday	Correlation Coefficient
Morning peak	High	Low	0.75
Noon	Medium	Medium	0.50
Evening peak	High	Medium	0.80
Night	Low	Low	0.30

**Table 3.** Load fluctuation analysis by lag time (time correlation).

Lag Time (Hours)	Correlation Coefficient
0	0.75
1	0.65
2	0.45

As shown in Figure 9, the variables with the higher correlation coefficients included temperature, sea level pressure, and surface pressure, which were  $-0.14$ ,  $0.09$ , and  $0.09$ , respectively. Since there was a high correlation between the sea level pressure and surface pressure, only one of these variables was selected for the model training. As shown in Table 2, the correlation of the charging load fluctuations during the peak hours on weekdays was stronger between the commercial and non-commercial areas, with correlation coefficients of  $0.75$  and  $0.80$ , respectively. This indicates that the charging demand trends in both areas were very consistent during commuting hours. The correlation was lower during the midday period, especially on non-working days, where the correlation coefficient at noon was  $0.50$ , suggesting that the charging demands in the commercial and non-commercial areas were more independent at this time. The nighttime charging demand was higher in the non-commercial areas, particularly on non-working days, which led to a reduction in the correlation ( $0.30$ ). As shown in Table 3, the time series correlation analysis indicated that the load fluctuations in the non-commercial areas lagged behind those in the commercial areas by about one hour during peak periods, with the correlation coefficient dropping to  $0.65$ . This suggests that the high load demand in commercial areas may cause a delayed response in the charging behavior in non-commercial areas. Based on these analytical results, this study ultimately selected temperature, pressure, hour, weekdays, and non-working days as the common feature variables for the prediction model. The peak time identifiers were selected as specific time feature variables for the commercial area, while the late peak hours, nighttime, the low peak hours on weekdays, and the one-hour lag of the commercial area load were selected as the specific time feature variables for the non-commercial area.

### 3.3. Model Evaluation Metrics

To compare the performances and predictive accuracies of different models, this study utilized the R-squared ( $R^2$ ), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) to quantitatively assess the prediction accuracy of each model. Smaller values of the  $R^2$ , MAE, and MAPE indicate higher model prediction accuracies [32]. The calculation method for each metric is as follows:

R-squared ( $R^2$ ):

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (16)$$

where  $y_i$  are the observed values,  $\hat{y}_i$  are the predicted values, and  $\bar{y}$  is the mean of the observed values.

Mean Absolute Error (MAE):

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (17)$$

Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (18)$$

## 4. Case Study

### 4.1. Dataset Preprocessing

The platform environment used in this experiment was the Windows 10 operating system, equipped with an Intel Core i5-12400F CPU and 8 GB of RAM. The model implementation was carried out using MATLAB. The learning rate in this study was set to 0.01. The learning rate controls the magnitude of weight updates during each iteration. A learning rate that is too high may lead to instability in the model training, while one that is too low could cause slow training or result in convergence to a local optimum. Through preliminary experiments and optimization, we found that 0.01 was a reasonable starting value, as it allowed the model to converge within a reasonable training time while avoiding overfitting. The dataset was split using a standard ratio of 80% for training and 20% for testing. This split effectively evaluated the model's generalization ability, especially in time series data, which ensured the independence of the model training and evaluation. The continuous variables, such as the charging load and charging duration for commercial and non-commercial areas, were standardized using the z-score normalization method to eliminate scale differences between the feature variables. The specific formula is as follows:

$$z = \frac{(X - \mu)}{\sigma} \quad (19)$$

where  $X$  is the value of the feature to be standardized,  $\mu$  is the mean of the feature values, and  $\sigma$  is the standard deviation of the feature values.

Next, for the discrete time features (such as weekdays/weekends, hours, peak/off-peak times), One-hot encoding was applied. The encoding method is shown in Table 4. To capture the sequential features, time sliding windows were set separately for commercial and non-commercial areas. In commercial areas, due to the large fluctuations in demand during the peak periods, the sliding windows can be set slightly shorter to capture rapidly changing load patterns. In non-commercial areas, due to a relatively stable charging demand, the sliding windows can be appropriately extended to capture nighttime and long-duration charging behaviors. Finally, the standardized and One-hot encoded feature matrix of the commercial area was input into the GRU model. The GRU model was capable of capturing peak charging demand and time dependencies in the commercial area. The standardized feature matrix of the non-commercial area was input into the KSVR model, which used an RBF kernel function because it excels at handling nonlinear features, making it suitable for the complex load patterns in non-commercial areas. After training the model weights, the models were validated on the test set, and relevant evaluation metrics were calculated.

**Table 4.** One-hot encoding example.

Time Feature	Weekday	Weekend	Morning Peak	Noon	Evening Peak
Weekday morning peak	1	0	1	0	0
Weekday noon	1	0	0	1	0
Weekday evening peak	1	0	0	0	1
weekend morning peak	0	1	1	0	0
weekend noon	0	1	0	1	0
weekend evening peak	0	1	0	0	1

#### 4.2. Comparison Between Parallel Execution and Independent Execution

In this study, we also conducted a comparison experiment between the independent execution and parallel execution of the GRU and KSVR to evaluate the impact of different execution methods on the model performance. The experiments were conducted using the same dataset and model configurations, and the efficiency and performance of both the independent and parallel executions were assessed. Table 5 summarizes the performance comparison between the independent execution and parallel execution modes.

**Table 5.** Performance comparison between independent and parallel execution.

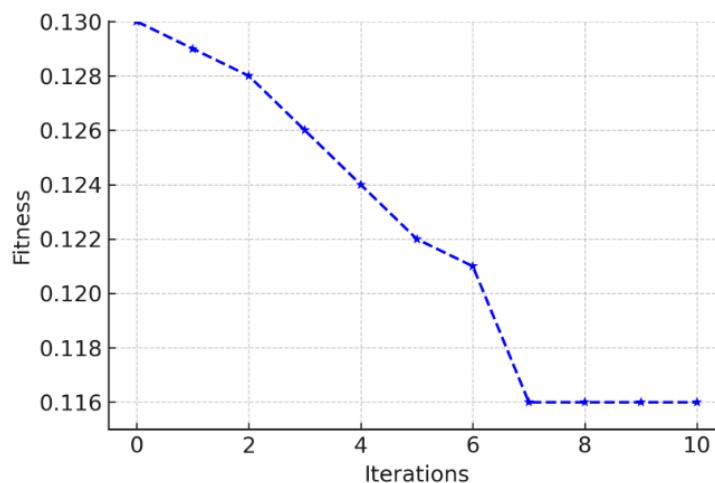
Execution Method	Average Computation Time (s)	CPU Usage (%)	Memory Usage (MB)
Independent	520	45	2000
Parallel	320	85	2500

From Table 5, it can be observed that the parallel execution mode significantly outperformed the independent execution mode in terms of the computation time, reducing the computation time from 520 seconds to 320 seconds. However, in terms of the resource consumption, the parallel execution mode exhibited higher CPU usage and memory consumption compared with the independent execution mode, with the CPU usage reaching 85%, while it was only 45% under the independent execution mode. The parallel execution mode demonstrated a notable advantage in computation time, making it suitable for use in scenarios where hardware resources are sufficient to enhance the model training and prediction efficiency. However, the higher CPU and memory usage indicate that this approach may not be suitable in resource-constrained environments. Therefore, a trade-off between real-time performance and resource consumption is required to select an appropriate execution method.

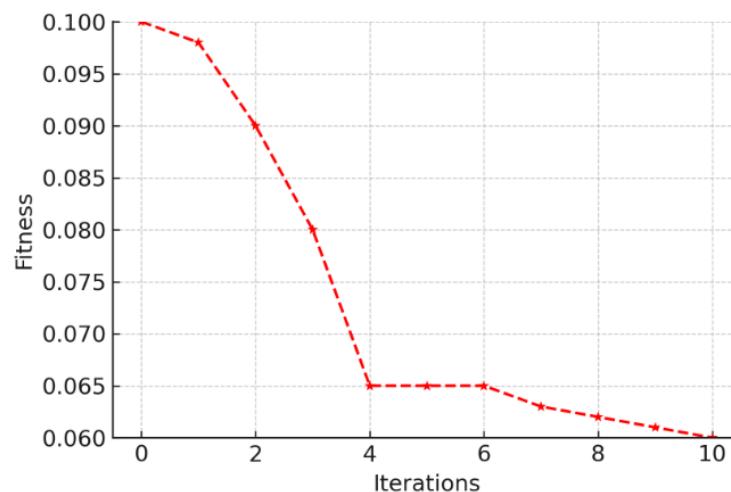
In summary, the regional prediction improved the prediction accuracy for different areas, which enhanced the overall load-forecasting performance. By more effectively capturing the characteristic differences of each region, the regional prediction allowed the load-forecasting models to be optimized for each specific area, which improved the overall accuracy. The regional prediction enabled the unique features of each region to be modeled separately, which resulted in a higher global prediction accuracy. When the prediction results of different regions were aggregated, this method generated an overall forecast that more closely reflected the actual load conditions, which reduced the potential biases that could arise in global predictions.

#### 4.3. Results and Analysis

To verify the predictive performance of the models in the commercial and non-commercial areas, the data were separately fed into the KSVR and GRU models. Each model underwent hyperparameter optimization using PSO, with the number of iterations set to 10 and a population size of 30. Figure 10 shows the PSO process for the GRU, while Figure 11 depicts the PSO process for the KSVR. As shown in Figures 10 and 11, the PSO fitness converged rapidly. The parameter optimization results for the GRU and SVR are presented in Table 6.



**Figure 10.** PSO-optimized GRU process.



**Figure 11.** PSO-optimized KSVR process.

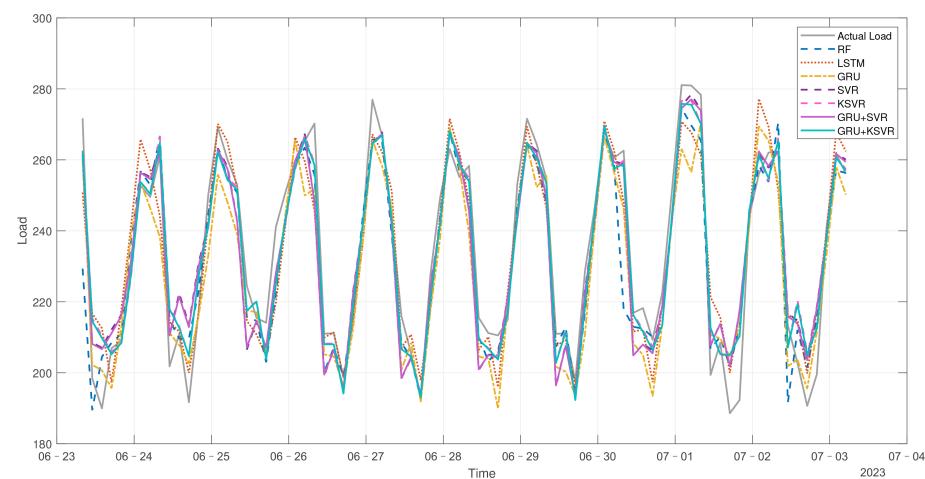
**Table 6.** Results of hyperparameter optimization for GRU and KSVR models.

Model Type	Hyperparameter	Value	Validation Loss	Accuracy
GRU	Learning rate	0.01	0.25	89%
GRU	Number of layers	2	0.23	90%
GRU	Batch size	64	0.22	91%
KSVR	Kernel type	RBF	0.30	85%
KSVR	C	10	0.28	86%
KSVR	Gamma	0.1	0.27	87%

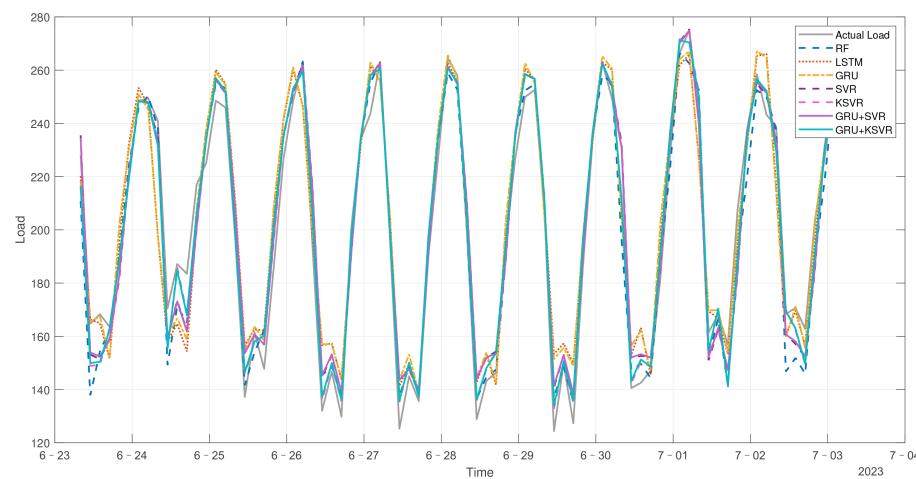
After the training, these models were used to predict the load on the test set. Figure 12 illustrates that in the commercial areas, the prediction curve of the GRU model closely aligned with the actual load trend, indicating its strong ability to capture dynamic changes in the charging load, especially in areas with high load fluctuations. Other models, such as the KSVR, RF, and LSTM, also performed well; however, compared with the GRU, they showed slightly larger errors at certain peaks and troughs, suggesting that they were less effective at capturing the rapid load changes. Figure 13 illustrates that in the non-commercial areas, the prediction curve of the KSVR model closely matched the actual load values, particularly during the periods of stable load. This demonstrates that the KSVR

effectively captured the load trends in the stable load environments, where it performed the best in the non-commercial areas. The GRU, RF, and LSTM models performed slightly worse in the non-commercial areas, especially in regions with smaller load fluctuations, making the KSVR more suitable for the load prediction in such regions. The combined use of the GRU and KSVR models significantly improved the performance compared with using a single model (such as the GRU or KSVR alone), particularly in terms of the prediction accuracy across different area types. Table 7 presents the key evaluation metrics for the proposed model in this study, including  $R^2$ , MSE, and MAPE. When processing the actual load data, the GRU-KSVR model outperformed both the GRU and KSVR models alone.

From Table 8, it can be seen that in both regions, the model initialized with K-means++ (K-means++-GRU-KSVR) outperformed the traditional K-means initialization method (K-means-GRU-KSVR) across all evaluation metrics. In the commercial area, the K-means++ method reduced the MAE to 6.20, lowered the MAPE to 2.84%, and maintained an  $R^2$  of 0.91. In the non-commercial area, the K-means++ approach achieved an MAE of 6.77, a MAPE of 3.80%, and an  $R^2$  of 0.97, indicating that the K-means++ initialization enhanced the model prediction accuracy. Although the K-means++ method performed well in both areas, overall, the model showed slightly better predictive performance in the non-commercial area, especially in terms of the  $R^2$  (0.97 for the non-commercial area compared with 0.91 for the commercial area). This may have been because the non-commercial area exhibited smaller load fluctuations, which made it more suitable for models like the KSVR and GRU. In summary, the use of K-means++ significantly improved the performance of the GRU-KSVR combination model, providing greater stability and accuracy, particularly in commercial areas with larger load fluctuations. This demonstrated that optimized clustering initialization methods, such as K-means++, could enhance the model prediction capabilities in complex scenarios.



**Figure 12.** Commercial area model learning curve.



**Figure 13.** Non-commercial area model learning curve.

**Table 7.** Performance metrics comparison .

Area	Model	MAE	MAPE	R <sup>2</sup>
Commercial area	GRU	8.94	4.00%	0.81
	SVR	6.75	3.04%	0.90
	RF	11.52	4.71%	0.67
	LSTM	9.48	4.27%	0.79
	KSVR	6.23	2.86%	0.91
	GRU-SVR	6.72	3.03%	0.90
	GRU-KSVR	6.20	2.84%	0.91
Non-commercial area	GRU	8.84	4.91%	0.93
	SVR	6.92	3.89%	0.96
	RF	9.18	4.87%	0.93
	LSTM	9.23	5.21%	0.93
	KSVR	6.84	3.94%	0.97
	GRU-SVR	6.86	3.86%	0.97
	GRU-KSVR	6.77	3.80%	0.97

**Table 8.** Performance metrics comparison.

Area	Model	MAE	MAPE	R <sup>2</sup>
Commercial area	K-means-GRU-KSVR	6.33	2.95%	0.91
	K-means++-GRU-KSVR	6.20	2.84%	0.91
Non-commercial area	K-means-GRU-KSVR	6.80	3.90%	0.97
	K-means++-GRU-KSVR	6.77	3.80%	0.97

This study compares the proposed K-means++-GRU-KSVR model with several mainstream electric vehicle charging load forecasting methods, as shown in Table 9. It is important to note that the datasets used by these methods differ, which may introduce certain limitations to the comparative results of prediction performance.

**Table 9.** Performance Comparison of K-means++-GRU-KSVR with Recent Methods.

Reference	Model Used	MAE	RMSE	MAPE (%)	R <sup>2</sup>
Goswami et al. [16]	ARIMA	158	190	13.73	0.735
Goswami et al. [16]	SARIMA	134	200	10.7	0.876
Shi et al. [19]	STMGCN	55.834	81.340	10.243	-
Xiong et al. [20]	CNN+LSTM+Transformer	0.120	0.300	-	-
Kim et al. [21]	mRGC-CBi-LSTM	0.151	0.276	-	0.902
Commercial Area	K-means++-GRU-KSVR	6.20	8.60	2.84	0.91
Non-commercial Area	K-means++-GRU-KSVR	6.77	8.54	3.80	0.97

The data comparison indicates that traditional methods (ARIMA and SARIMA) perform moderately in capturing changes in electric vehicle charging load. Although SARIMA shows lower MAE and MAPE compared to ARIMA, the overall errors remain relatively high, limiting prediction accuracy. Deep learning-based methods, such as STMGCN and hybrid models (CNN+LSTM+Transformer and mRGC-CBi-LSTM), demonstrate excellent performance in handling complex spatiotemporal relationships and nonlinear patterns. Among these, CNN+LSTM+Transformer achieves the highest accuracy with extremely low MAE and RMSE. However, the proposed K-means++-GRU-KSVR model performs exceptionally well in both commercial and non-commercial areas. Particularly, in commercial areas with complex load fluctuations, its MAPE is significantly lower than other methods. Meanwhile, in non-commercial areas, it maintains high stability and prediction capability. In conclusion, K-means++-GRU-KSVR is a robust and highly accurate electric vehicle charging load forecasting model.

In summary, regional forecasting enhanced the prediction accuracy for the different areas, which improved the overall effectiveness of the load forecasting across the entire region. By capturing the characteristic differences between areas more effectively, the regional forecasting allowed for the optimization of load-prediction models for each specific area, which increased the accuracy at a global level. The regional forecasting enabled separate modeling of each area's unique characteristics, which led to an improved accuracy on a global scale. When the prediction results of different areas were combined, this approach generated an overall forecast more closely aligned with the actual load conditions, which reduced the biases that may occur in holistic forecasting.

## 5. Discussion

This study validated the effectiveness of the K-means++ clustering-based GRU-KSVR combined model for short-term electric vehicle charging load prediction through a case analysis. The experimental results indicate that this model could effectively predict short-term electric vehicle charging load trends, which are crucial for load management and sustainable energy planning in the power grid. By incorporating K-means++ clustering to account for the regional differences in charging patterns; then applying the GRU and KSVR models to different regions; and finally, optimizing the combination through weighted adjustments, the model demonstrated strong adaptability in various scenarios. Compared with traditional single-model predictions, the proposed combined model performed better at capturing the complex non-linear characteristics of charging loads. Specifically, K-means++ clustering effectively identified the features of different charging behavior groups, which allowed the GRU and KSVR to model the dynamics of different regions more accurately. Additionally, the introduction of the weighted optimization method significantly improved the overall accuracy of the model, which confirmed the effectiveness of the multi-model integration. Therefore, combining the strengths of multiple models may become a key direction for future research in complex tasks, like electric vehicle load forecasting. Despite achieving promising results, this study had several limitations. First, during the

model training, deep learning models (such as a GRU) usually require large amounts of high-quality historical data, while the dataset used in this study was relatively small. This data limitation may affect the model's generalization ability. Therefore, collecting more long-term charging load data will be crucial for further validating and optimizing the model in future research. Additionally, the load data used in this study mainly came from electric vehicle charging stations in specific regions. Future research should consider the behavioral differences of various types of electric vehicle users and the unique load characteristics of different areas (such as commercial zones, residential zones, and public charging stations). Another aspect worth considering is the model's applicability. Although the model showed good predictive performance in the experiments, its suitability for other load-prediction scenarios (such as city-level grid loads or industrial loads) has not yet been verified. Future research should expand into other scenarios and consider diverse influencing factors, such as seasonal changes and weather conditions, to further enhance the model's robustness and applicability. In conclusion, the proposed K-means++ clustering and GRU-KSVR combined prediction model demonstrated strong potential in electric vehicle charging load forecasting. However, further validation and optimization on larger-scale datasets are needed to ensure its broad applicability and stability in practical applications.

## 6. Conclusions

To improve the accuracy of short-term load forecasting for electric vehicle charging and to support energy planning and management, this paper proposes a K-means++ clustering-based GRU-KSVR combined model. Through theoretical analysis and case studies, the following conclusions were drawn:

- The proposed model effectively addresses the complexity and non-linearity of electric vehicle charging load data by utilizing K-means++ clustering to divide charging behaviors into distinct regions. This enables the GRU and KSVR models to capture the dynamic characteristics of each region more accurately, significantly enhancing the precision of short-term load forecasting.
- The combination of GRU and KSVR models provides a balanced approach, where the GRU handles time-dependent sequences while the KSVR fine-tunes the final predictions. This integrated method optimizes the model's prediction capabilities for both stable and fluctuating load scenarios.
- By applying weighted optimization to the outputs of GRU and KSVR, the model achieved a higher forecasting accuracy compared with using individual models alone. The flexibility of the weighted combination allowed the model to better adapt to different regions and conditions, which provided reliable predictions under various circumstances.
- Improving the accuracy of short-term load forecasting through the proposed method enables better management of energy resources, allowing for optimized energy consumption, reduced operational costs, and minimized environmental impact. This forecasting model equips grid operators and stakeholders with essential predictive insights, empowering them to make informed decisions regarding energy procurement, infrastructure investments, and daily operational adjustments.

In summary, the K-means++-GRU-KSVR model demonstrated its potential as an effective solution for electric vehicle charging load forecasting, providing valuable support for the sustainable management and planning of energy resources in modern power systems.

**Author Contributions:** Conceptualization, R.S.; Methodology, R.S.; Software, R.S.; Writing—original draft, R.S.; Writing—review and editing, R.S. and Y.M.; Supervision, Y.M. All authors read and agreed to the published version of this manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The raw data supporting the conclusions of this article will be made available by the authors on request.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

- Yaghoubi, E.; Yaghoubi, E.; Khamees, A.; Razmi, D.; Lu, T. A Systematic Review and Meta-analysis of Machine Learning, Deep Learning, and Ensemble Learning Approaches in Predicting EV Charging Behavior. *Eng. Appl. Artif. Intell.* **2024**, *135*, 108789. [[CrossRef](#)]
- Yin, W.; Ji, J. Research on EV Charging Load Forecasting and Orderly Charging Scheduling Based on Model Fusion. *Energy* **2024**, *290*, 130126. [[CrossRef](#)]
- Andrenacci, N.; Valentini, M.P. A Literature Review on the Charging Behaviour of Private Electric Vehicles. *Appl. Sci.* **2023**, *13*, 12877. [[CrossRef](#)]
- Schey, S.; Scoffield, D.; Smart, J. A First Look at the Impact of Electric Vehicle Charging on the Electric Grid in The EV Project. *World Electr. Veh. J.* **2012**, *5*, 667–678. [[CrossRef](#)]
- Shahzad, S.; Zaheer, A.; Ullah, H.S.; Kilic, H.; Gono, R.; Jasiński, M.; Leonowicz, Z. Short-Term Load Forecasting Models: A Review of Challenges, Progress, and the Road Ahead. *Energies* **2023**, *16*, 4060. [[CrossRef](#)]
- Liu, Z.; Chen, X.; Liang, X.; Huang, S.; Zhao, Y. Research on Sustainable Form Design of NEV Vehicle Based on Particle Swarm Algorithm Optimized Support Vector Regression. *Sustainability* **2024**, *16*, 7812. [[CrossRef](#)]
- Li, X.; Jiang, M.; Cai, D.; Song, W.; Sun, Y. A Hybrid Forecasting Model for Electricity Demand in Sustainable Power Systems Based on Support Vector Machine. *Energies* **2024**, *17*, 4377. [[CrossRef](#)]
- Gao, Z.J.; Dai, R.; Guo, X.F.; Ning, Y. Research on Multi-Scale Signal Processing and Efficient Model Construction Strategies in Lithium-Ion Battery State Prediction. *J. Electrochem. Soc.* **2024**, *171*, 120509. [[CrossRef](#)]
- Yang, L.; Li, Z.; Wang, X.; Liu, Y. A Data-Driven Model for Electric Vehicle Charging Load Prediction Based on GRU and Attention Mechanism. *J. Power Sources* **2024**, *480*, 228817. [[CrossRef](#)]
- Almeshiae, E.; Soltan, H. A Review of Load Forecasting of the Production in an Electric Power Generation System Using ARIMA Model. *Renew. Sustain. Energy Rev.* **2011**, *15*, 2052–2056. [[CrossRef](#)]
- Contreras, J.; Espínola, R.; Nogales, F.J.; Conejo, A.J. ARIMA Models to Predict Next-Day Electricity Prices. *IEEE Trans. Power Syst.* **2003**, *18*, 1014–1020. [[CrossRef](#)]
- K., V.S.M.B.; Chakraborty, P.; Pal, M. Planning of fast charging infrastructure for electric vehicles in a distribution system and prediction of dynamic price. *Int. J. Electr. Power Energy Syst.* **2024**, *155*, 109502. [[CrossRef](#)]
- Wahyudi, A. J.; Febriani, F. Time-series forecasting of particulate organic carbon on the Sunda Shelf: Comparative performance of the SARIMA and SARIMAX models. *Reg. Stud. Mar. Sci.* **2024**, *80*, 103863. [[CrossRef](#)].
- Toledo-Mercado, E.; Soto, I. Adaptive MRA Digital Filter with Wavelet and NARX Neural Network for Ground-Based Transmission Spectra. In Proceedings of the 2024 IEEE International Conference on Automation and the 26th Chilean Association of Automatic Control (ICA-ACCA), Santiago, Chile, 20–23 October 2024; pp. 1–8. [[CrossRef](#)]
- Chen, L.; Liu, P. A GLM-Based Approach for Predicting Electric Vehicle Charging Load in Mixed Residential and Commercial Areas. *Energy Convers. Manag.* **2024**, *212*, 112–123. [[CrossRef](#)]
- Goswami, K.; Kandali, A. B. Power Demand Forecasting Using Data-Driven Predictive Approaches: ARIMA and SARIMA for Real-Time Load Data of Assam. In Proceedings of the 2020 International Conference on Computational Performance Evaluation (ComPE), Shillong, India, 2–4 July 2020; pp. 570–574. [[CrossRef](#)]
- Jawad, S.; Liu, J. Electrical Vehicle Charging Load Mobility Analysis Based on Spatial-Temporal Model in Traffic-Distribution Networks. In Proceedings of the 2023 Panda Forum on Power and Energy (PandaFPE), Chengdu, China, 27–30 April 2023; pp. 700–704. [[CrossRef](#)]
- Wu, Y.; Cong, P.; Wang, Y. Charging Load Forecasting of Electric Vehicles Based on VMD–SSA–SVR. *IEEE Trans. Transp. Electrif.* **2024**, *10*, 3349–3362. [[CrossRef](#)]
- Shi, J.; Zhang, W.; Bao, Y.; Gao, D.W.; Wang, Z. Load Forecasting of Electric Vehicle Charging Stations: Attention Based Spatiotemporal Multi-Graph Convolutional Networks. *IEEE Trans. Smart Grid* **2024**, *15*, 3016–3027. [[CrossRef](#)]
- Xiong, X.; Zhou, L. A Method of Short-Term Load Forecasting at Electric Vehicle Charging Stations Through Combining Multiple Deep Learning Models. In Proceedings of the 2023 2nd Asia Power and Electrical Technology Conference (APET), Shanghai, China, 28–30 December 2023; pp. 740–744. [[CrossRef](#)]
- Kim, H.J.; Kim, M.K. Spatial-Temporal Graph Convolutional-Based Recurrent Network for Electric Vehicle Charging Stations Demand Forecasting in Energy Market. *IEEE Trans. Smart Grid* **2024**, *15*, 3979–3993. [[CrossRef](#)]
- Xin, F.; Yang, X.; Wang, B.; Xu, R.; Mei, F.; Zheng, J. Research on Electric Vehicle Charging Load Prediction Method Based on Spectral Clustering and Deep Learning Network. *Front. Energy Res.* **2024**, *12*, 1294453. [[CrossRef](#)]
- Jahromi, A.J.; Masoudi, M.R.; Mohammadi, M.; Afrasiabi, S. Improving Electric Vehicle Charging Forecasting: A Hybrid Deep Learning Approach for Probabilistic Predictions. *IET Gener. Transm. Distrib.* **2024**, *18*, 1267–1283. [[CrossRef](#)]
- Jain, A.K. Data clustering: 50 years beyond K-means. *Pattern Recognit. Lett.* **2010**, *31*, 651–666. [[CrossRef](#)]
- Yang, Y.; Cheng, C. Construction of Accurate Student Funding Group Portraits Based on K-means Algorithm. In Proceedings of the 2023 8th International Conference on Information System Engineering (ICISE), Dalian, China, 23–25 June 2023; pp. 154–158. [[CrossRef](#)]

26. Niu, Q.; Li, X. A High-Performance Web Attack Detection Method Based on CNN-GRU Model. In Proceedings of the 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, 12–14 June 2020; pp. 804–808. [[CrossRef](#)]
27. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555. [[CrossRef](#)]
28. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [[CrossRef](#)]
29. Zhan, Y.; Zhang, H.; Liu, Y. Forecast of Meteorological and Hydrological Features Based on SVR Model. In Proceedings of the 2021 4th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), Changsha, China, 26–28 March 2021; pp. 579–583. [[CrossRef](#)]
30. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; pp. 1942–1948. [[CrossRef](#)]
31. Benesty, J.; Chen, J.; Huang, Y.; Cohen, I. Pearson correlation coefficient. In *Noise Reduction in Speech Processing*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–4. [[CrossRef](#)]
32. Chai, T.; Draxler, R.R. Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature. *Geosci. Model Dev.* **2014**, *7*, 1247–1250. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.