applied sciences

MDPI

Article

Self-Adaptive Clustering Model Based on Variable Time-Series Similarity Measure Analysis for V2G Electricity Price Prediction

Tie Hua Zhou [1], Xirao Xun [1], Ling Wang [1,*], Gongliang Hu [2], Wei Ding [3,4] and Lei Kou [5]

[1] Department of Computer Science and Technology, School of Computer Science, Northeast Electric Power University, Jilin 132013, China; thzhou@neepu.edu.cn (T.H.Z.); 2202301039@neepu.edu.cn (X.X.)
[2] Qingdao Integrated Service Group for Talented Individuals and Enterprises, Qingdao 266000, China; hugongliang777@gmail.com
[3] Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250000, China; dingw@sdas.org
[4] Shandong Provincial Key Laboratory of Computer Networks, Shandong Fundamental Research Center for Computer Science, Jinan 250000, China
[5] Institute of Oceanographic Instrumentation, Qilu University of Technology (Shandong Academy of Sciences), Qingdao 266000, China; koulei1991@qlu.edu.cn
* Correspondence: smile2867ling@neepu.edu.cn

Abstract: Data with time attributions such as price, load, and stock, which directly reflect the variation tendency, are the most common type of data character available. However, it is difficult to predict complex and volatile time-series character data. Further, most density cluster methods employ existing data to train the initial radius; however, a certain density radius is hard to be made suitable for continuously generated on-going datasets. Therefore, how to select a suitable timespan according to the time-series character in a way that makes it possible to support an adaptive updated density radius for real-time calculation is a core process. In this paper, a self-adaptive multi-density (SAMD) prediction model is proposed for solving the dynamic density radius selection problem in time-series data so as to improve the accuracy of real-time prediction. This multi-density clustering method can effectively shorten the iteration times and achieve dynamic clustering by the proposed jump sequence, which can optimize the jump points in the electricity price sequence. Moreover, we especially focus on the time interval features and other multi-source influencing factors together to construct the multi-core function with double-layer optimization to calculate the weighted coefficients, which have good adaptability and improve the classification and recognition performance. The experimental results show that the model had higher prediction accuracy and reduced processing time consumption in order to achieve real-time prediction.

Keywords: self-adaptive prediction model; real-time electricity price analysis; smart grid; data mining

check for updates

Academic Editor: Fabio La Foresta

Received: 15 November 2024
Revised: 1 February 2025
Accepted: 12 February 2025
Published: 16 February 2025

Citation: Zhou, T.H.; Xun, X.; Wang, L.; Hu, G.; Ding, W.; Kou, L. Self-Adaptive Clustering Model Based on Variable Time-Series Similarity Measure Analysis for V2G Electricity Price Prediction. *Appl. Sci.* **2025**, *15*, 2069. https://doi.org/10.3390/app15042069

Copyright: © 2025 by the authors. 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/).

1. Introduction

Real-time tariffs are generated in real time and are constantly accumulated and calculated at a fast rate, which means that large, complex time-series data are involved, requiring efficient algorithms and techniques to process and analyze them [1]. Big data presents a complex technical challenge due to datasets with large volumes, diverse categories, and intricate structures, requiring innovative frameworks and techniques to efficiently extract valuable insights [2]. Therefore, real-time electricity price forecasting is not only an important area of big data application but also a key aspect of electricity market analysis. In

recent years, the focus on big data analysis with time attribution has become increasingly popular for analyzing and predicting electricity prices, especially for real-time electricity price forecasting [3] and energy trading for electric vehicles. As electricity price structures become more diverse, an increasing number of power utilities integrate demand charges into their tariffs. Households equipped with photovoltaic (PV) and energy storage systems can respond to this shift by implementing effective energy management strategies [4]. Accurate price forecasting not only brings stable benefits to market participants but also arranges data in making reasonable user demand-side planning decisions [5]. Renewable energy will undoubtedly have a huge impact on electricity price volatility as wind and solar resources are integrated into the electricity system at high penetration rates [6]. Therefore, forecasting data with time attribution is becoming an important part of the research on the power economy market [7].

With changes in climate, festivals, etc., time-series data have dynamic change characteristics with time lapses [8]. Time-series clustering is used to identify feature-driven clusters based on the similarity method [9]. In recent years, clustering methods have usually divided into three categories: Firstly, structural characteristics are mostly considered the shape-shifting of cluster datasets [10]. Secondly, structural characteristics can be constructed by clustering regulation using the basic statistical features, time or frequency feathers. Thirdly, model characteristics are based on the density function that reflects the spatial distribution of data points. Connectivity is a typical feature of the density clustering method; it is suitable for solving real-time processing problems, especially time-series parameters, and for classifying suitable timespan intervals. Furthermore, calculating a dynamic density radius to satisfy the real-time operation is still a big challenge.

Real-time tariff forecasting is an important aspect of electricity market management, especially with the rapid development of new energy sources, home energy trading, and microgrid communities. The popularity of electric vehicles (EVs) brings new challenges and opportunities to the power system. On the one hand, the charging demand of EVs can lead to an increase in the volatility of electric load. On the other hand, EVs can also be used as distributed energy resources to balance the grid load by providing power back during peak grid load hours [11].

Home energy trading refers to the production and sale of electricity by residential customers through distributed energy resources (e.g., photovoltaic power systems). This type of trading not only improves the efficiency of energy utilization but also brings economic benefits to household users [12]. However, the popularity of home energy trading also places greater demands on real-time tariff forecasting in the electricity market, which needs to take into account the fluctuations in distributed energy production and the consumption habits of users [13].

Therefore, we propose a self-adaptive multi-density (SAMD) prediction model to support adaptive changes in the density radius over spatial distribution of time-series data. The main contributions of this study are outlined below:

- An adaptive multi-density (SAMD) prediction model is proposed to solve the dynamic density radius selection problem and improve the accuracy and real-time performance of time-series data prediction.
- One essential step is the calculation of the timespan according to real-time electricity prices, which is able to represent the general electricity tendency by mining the key points of different levels of clusters.
- When forecasting the time series, the effect of the influencing factors is fully considered, so the multi-core function is used to analyze the influencing factors.

- The SAMD model with timespan processing not only deals with the problem of excluding noise points due to the threshold of the set of MITs but also classifies a few of the closest noise points into sparse clusters.

A microgrid community is a regional energy management system based on distributed energy technology that allows the balance of power production and consumption in a localized area [14]. The emergence of microgrids makes energy management more flexible and efficient but also increases the complexity of the power system and the volatility of electricity prices. Therefore, the management of a microgrid community relies on high-precision, real-time electricity price forecasts to ensure a stable and economical power supply.

The SAMD model proposed in this paper can predict the general trend of electricity prices by separating the time-series dataset into high-quality clusters of high, medium, and low levels and representing them as different kinds of real-time electricity price intervals. This approach can reflect the dynamics of the electricity market more accurately and in real time, thus optimizing new energy utilization, improving the economics of home energy transactions, and enhancing the stability and sustainability of microgrid communities.

Figure 1 illustrates energy trading within a microgrid community. The households in this community are assumed to be photovoltaic, have electric vehicles, and are equipped with smart meters and home energy storage batteries. With the popularity of electric vehicles, the provision of services to the grid, homes, and buildings via Vehicle-to-Anything (V2X) technology [15] is also considered in Figure 1.

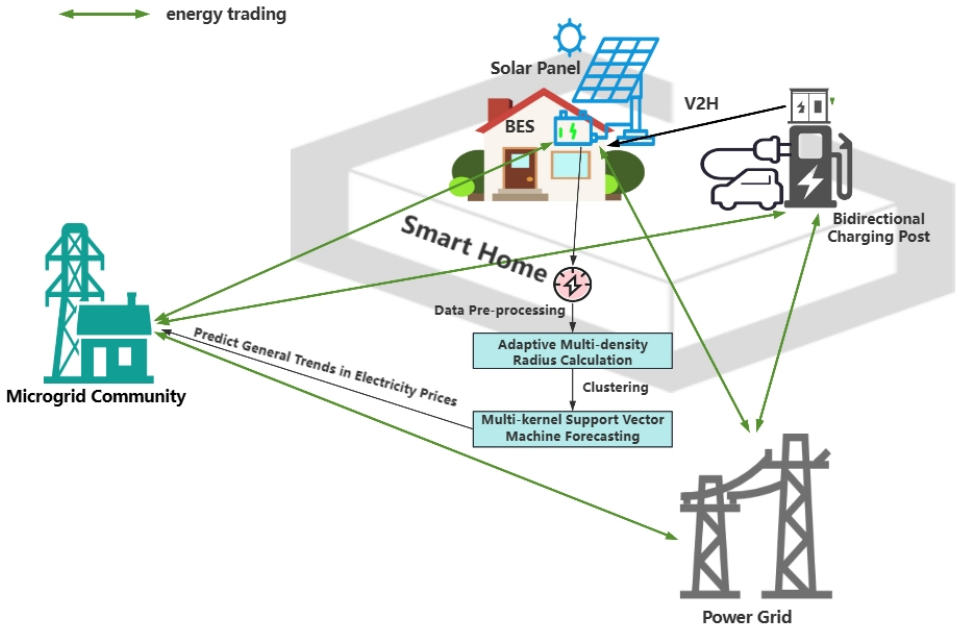


Figure 1. Microgrid community energy trading.

The remainder of this paper is organized as follows: Section 2 describes related work. Section 3 proposes a method for timespan processing. Section 4 describes the SAMD model in detail, including its adaptive multi-density radius calculation and multi-kernel support vector machine forecasting. Section 5 provides a description of the experiment and comparative analysis. Finally, the conclusions of this paper are summarized in Section 6.

2. Related Work

General cluster analysis [16] is based on vectors to divide original time-series data into equal-length sets or overlapping-sequence sets, as based on the user-specified length and time granularity before sorting. The data stream time-series sequence is recorded

in chronological order; however, it still appears to be inefficient in solving the real-time processing problem.

Currently, there are four types of time-series clustering measures, which forecast similarity (or distance), features, models, and segmentation. Similarity measures are commonly used in Euclidean distance, DTW, or some improvement distance measures [17–19]. Similarity measures can contribute to determining the vector space, whereas the length of the time series is usually variable and sensitive to time variation but cannot be efficiently denoted as a point in the field of finite-dimensional space.

Prakaisak, I et al. first proposed a novel unsupervised small-shape-learning and adaptive neighborhood (USLA) model for learning prominent polyvariable quantum sequences. However, the USLA model performed as well with isometric multivariable shapes with different lengths, so a novel multiview USLA (MUSLA) model was developed, which treats the representation of small shapes learned from shapes of different lengths as different views [20].

Li, H et al. proposed a time-series clustering method based on complex network synchronous matching (CSC). The method uses a density peak clustering algorithm to identify the state of each time point and obtains the state sequence according to the time axis of the original time series. By synchronously comparing two state sequences, the length of the state sequence is calculated with the step length and the similarity is given, thus forming a new method for calculating the similarity of time series [21].

Li et al. proposed the ACDTW algorithm to adaptively calculate the similarity distance among multiple points in a certain time series, which can be mapped to related points in other time series [22]. Yu, C et al. proposed a locality-sensitive hashing method by evaluating distance errors for pruning to solve the multivariate time-series problem [23]. Ma, R et al. proposed a distance density clustering method that is medoid-based clustering with time-series data density consideration that provides clustering results in a hierarchical fashion. The DTW Barycenter Averaging (DBA) algorithm is a global averaging method that uses DTW and has proven to be an effective averaging method for time-series data [24].

When the length of time-series dynamic adjustment by external conditions changes or constraint functions occur, better accuracy can be obtained in time-series clustering results [25]. Sotiriadi et al. [26] proposed cluster time-series data based on the Fourier transform. They assumed that the data used had been preprocessed to remove the effects of non-seasonal factors and normalize them to allow a comparison of sales of different items on the same scale. However, although traditional time-series forecasting methods, such as ARIMA [27], SARIMA, etc., can capture the regularity of time-series data to a certain extent, they often appear to be incompetent in the face of the complex and volatile electricity market, especially in the prediction of real-time electricity prices [28].

These methods usually require a large amount of historical data for training and find it difficult to adapt to updated real-time data. Although density clustering methods have advantages in dealing with complex data, the selection of their initial radius often depends on existing data, which makes them inflexible in dealing with continuously generated real-time data [29].

3. SAMD Model for Time-Series Clustering

3.1. Motivation

Traditional density clustering methods find it difficult to cluster precisely if the density of spatial clustering is uneven and the spacing difference in axes is large (e.g., the dimension of the horizontal axis does not match the dimension of the vertical axis). This is because the parameters of the density radius and timespan threshold are difficult to match with each other, which can further cause clustering failure, as shown in Figure 2.

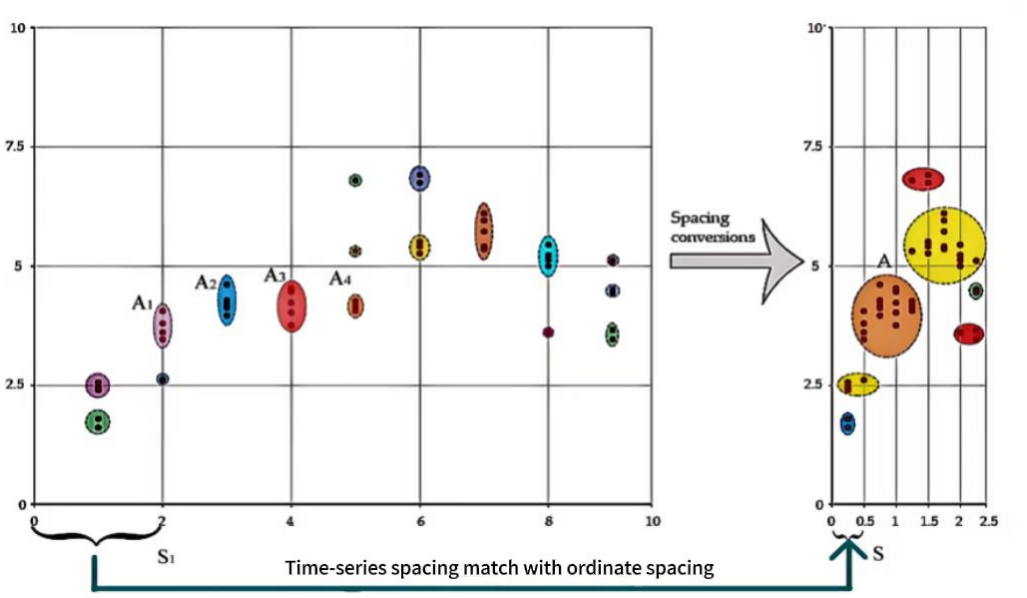


Figure 2. Timespan interval (such as S_1 and S, et al.) selection.

Real-time electricity prices are dynamic data that change over time, and if they are roughly handled in the clustering process, it is easy for some important features with time attributes to be lost, which is not conducive to the accurate clustering of electricity prices and thus affects the prediction accuracy. The randomly generated cluster center is not only complicated and the clustering effect is not as expected, and if the initial cluster center is properly selected, it can effectively reduce the number of iterations and thus reduce the clustering time, so the point density algorithm is used to initialize the clustering center.

3.2. Timespan Processing

The timespan interval (S_1 and S, as shown in Figure 2) is a very important measurable parameter in determining the quality of density clustering because it affects the initialization of the density radius. If the timespan interval is not suitable for the spacing of the y-axis, it will lead to the clustering results being non-ideal. If the timespan interval selected is too large, it will lead to more vertical space clusters. For instance, the x-axis spacing is represented as a time series, such as 5 min or 10 min. If the timespan interval is too large, such as S_1, the results of the density clusters will deviate from the actual cluster results, and some vertical-space, little clusters could be generated. If the suitable timespan interval (S) can be mined, these little clusters, such as A_1, A_2, A_3, and A_4, could be merged into one cluster, A. This is the reason why we need to consider the timespan to suit the dimension of density clustering; it is possible for it to have a great impact on the clustering quality because of proportional variation and the position offset. The time interval transformation is shown in Figure 3.

Definition 1. *Unmapped sample points are indicated as* $\mathrm{SX} = \{(x_1, y_1), (x_2, y_2) \ldots (x_i, y_i) \ldots (x_n, y_n)\}$, *which are distributed among dimensional, non-uniform space. The weighted average distance is selected to adjust the scale of the x-axis, and the sample points are mapped to* $\mathrm{SRX} = \{(x_{r1}, y_1), (x_{r2}, y_2) \ldots, (x_{ri}, y_i) \ldots, (x_{rn}, y_n)\}$ *through the mapping relation P1→P2, after spacing conversion.*

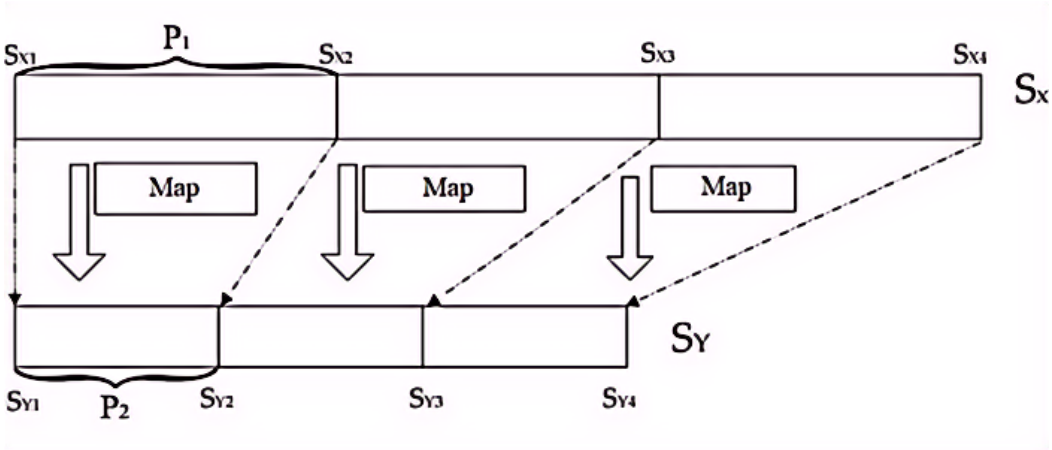


Figure 3. Mapping and transformation process , such as P1 (S_{X1}, S_{X2}) transforms into P2 (S_{Y1}, S_{Y2}) after mapping process.

4. Self-Adaptive Multi-Density Prediction Model

4.1. Data Preprocessing

Efficient and accurate data mining calculation can ensure that a few features reflect the main information of the time series. We present the key points algorithm to combine the angle method with the extreme value method, in order to find the main change characteristics in the data sequence.

Due to the variety and irregularity of the influencing factors of real-time electricity prices, this paper differentiates and fuses them, constructs multiple kernel functions by choosing different kernel functions, and maps each feature to a specific interval to deal with qualitative and quantitative indexes with different quantities by the normalization method.

As the tendency of the demarcation point of the time series changes, the key point can affect the changing patterns. The key points in data represent the data sequence because they find the main variation characteristic. Therefore, we propose combining the extreme value method with the angle method. For X_i, the smaller the angle of X_{i-1}, X_i, and X_{i+1} is, the more likely it is to be the key point. By only scanning the dataset once, we keep the important point in the non-monotonic sequence, and according to the monotone of the three consecutive data points of the triangle length of the trip, we determine whether the intermediate data points need to retain the key points. As Figure 4 shows, X_i in (a) and (b) are more likely to be the key points.

Theorem 1. *X_{i-1}, X_i, and X_{i+1} consist of three sequential data points. The probability that X_i is the turning point is related to $\left| x_i - \frac{x_{i+l}+x_{i-l}}{2} \right|$; the smaller $\left| x_i - \frac{x_{i+l}+x_{i-l}}{2} \right|$ is, the smaller the likelihood that it is the turning point.*

After the key points are selected, the time-series intervals become unequal lengths. Then, we linear fit the key points into a new sequence, which represents the time series. The size of the new sequence is much smaller than that of the original time data sequence, achieving the purpose of data reduction.

4.2. Adaptive Multi-Density Radius Calculation

The clustering center is the most important key point in the dataset; the closer the other points are to the center, the higher the density is. The density value transform is constrained by the bounds of the cluster, and the number of boundary points must be smaller than the maximum point.

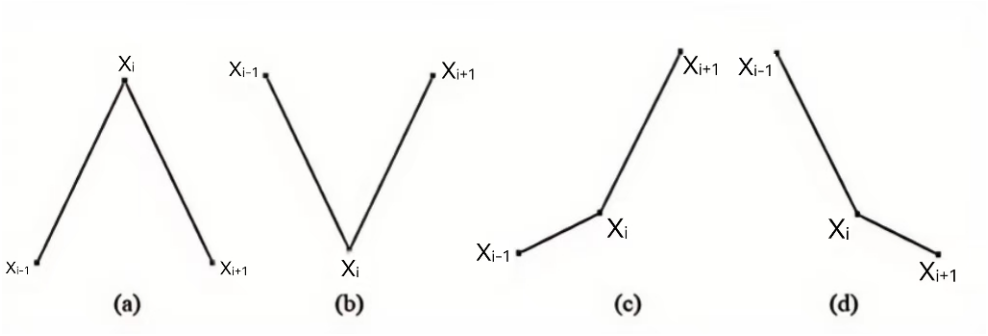


Figure 4. Keypoints (can affect the changing patterns and find the main variation characteristic) detection, such as X_i in (**a**,**b**), and (**c**,**d**).

4.2.1. Density Radius EPS Calculation

The density radius EPS calculation is used to evaluate the cluster density of each key point, where a higher density cluster contains more closely spaced points.

$$r = k \times \Gamma(n) \times \frac{m}{i \times \sqrt{\pi^n}}^{\frac{1}{n}} \times \prod_{i=1}^{n} x_i. \quad (1)$$

In the formula, r represents the radius, k represents the threshold for core objects, $\Gamma(n)$ represents the gamma function, and m represents the dimensions of the points.

4.2.2. Adaptive Radius Selection

The highest-density cluster is designated as an initial cluster; then, we calculate the average distance among all cluster points as the new density radius. Meanwhile, MinPts is denoted as $K - 1$, and the DBSCAN method is then used for clustering.

4.2.3. Assignment

Labeled as clustered points, these points will not be involved in subsequent clusters. Then, we pick the next-highest-density cluster to calculate by the same way, until all clusters are labeled. Algorithm 1 shows the detailed algorithm.

Algorithm 1: SAMD algorithm

Input: The key point of in real time series expressed in time series
Output: Different classes of data
1: **Begin**
2: R = 0; J = 0;
3: **for** i from 1 to Set of Points size **do**
4: **if** distance (Unsigned Point, i) < Reference Radius
5: R = R + k − distance [i];
6: J = J + 1;
7: R = R / J;
8: **end if**
9: **end for**
10: MULTI DENSITY(uobj, radiuses)
11: **for** each obj ∈ uobj
12: Set the label of obj as noisy
13: dis = L1 distance between obj and oi
14: **if** dis < density radius of o
15: mark obj with same label of o
16: **end if**
17: **end for**
18: **End**

4.3. Multi-Kernel Support Vector Machine Forecasting

Many factors can influence time-series fluctuations; therefore, when predicting time series, the effects of the influencing factors should be fully considered, and a multi-kernel function is used to analyze the influencing factors. The multi-kernel function is a kernel function corresponding to different types of data construction and weights and combines according to the importance and the degree of each kernel function to form a multi-kernel function. As shown in Figure 5, multi-kernel functions can combine multiple types of influencing factors and adjust the weights to form an optimal function. Compared to traditional single-core functions, multi-kernel functions are more robust and accurate.

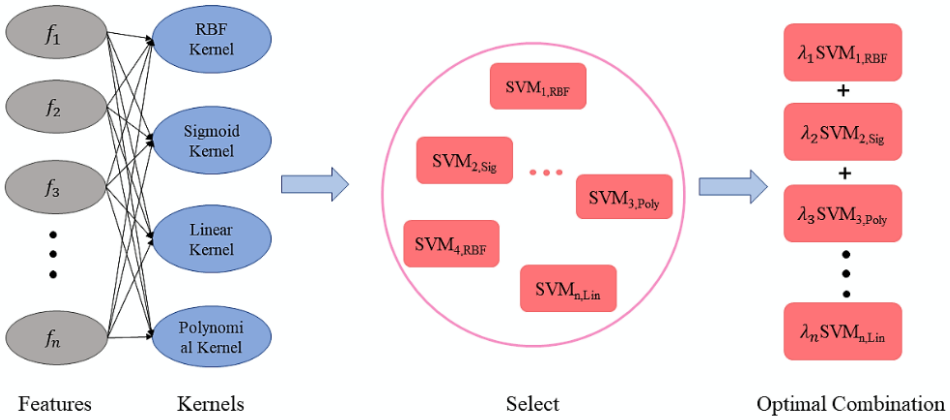


Figure 5. Multi-kernel support vector machine process principle.

Multi-kernel functions have two typical types: linear multi-kernel functions and nonlinear multi-kernel functions. Linear multi-kernel functions can be transformed into quadratic programming problems when solving decision functions. The parameters can be optimized according to the convexity of linear functions. For nonlinear multi-kernel functions, the quadratic programming problem does not have convexity, which makes it difficult to guarantee global stability.

Each influencing factor sample is represented in the corresponding coordinate system, and the distribution characteristics of the factor are judged according to the sample distribution. According to the characteristics of each factor, we choose the multi-kernel function $K_{MKL(x_i,x_j)}$ corresponding to the optimal kernel function $K_{(x_i,x_j)}$:

$$K_{MKL}(x_i, x_j) = \sum_{n=1}^{m} \lambda_n K_n(x_i, x_j) \tag{2}$$

Above, $K_{MKL(x_i,x_j)}$ represents the single kernel function, λ_n represents the kernel function weight, x_i represents the time point, x_j is the j-th influencing factor's sample value at the time point, and m is the number of influencing factors.

4.3.1. Multi-Kernel Support Vector Machine in LP Paradigm

In the multi-kernel support vector machine, the weight coefficients λ_n need to be set. If λ is inappropriate, over-fitting may occur in the optimization process. If we add a rule to the weight coefficient, we can effectively prevent over-fitting. The commonly used rules are the L_1 normal form and L_p normal form. Although the L_1 normal form can limit the scope of the search space, it makes it easy to produce sparse solutions, which makes it ignore the case that μ is 0. Therefore, we add a rule to the weight coefficient; it can effectively prevent

over-fitting. In this paper, we adopt a multi-kernel support vector machine based on the L_p normal form. Its objective function and constraints are shown below:

$$\min_{\tilde{w},b} \frac{\lambda}{2}(\sum_{k} \|\tilde{w}_k\|_2)^2 + \frac{1}{N}\sum_{i=1}^{l}(\xi_i + \hat{\xi}) + \tilde{\mu}\|\mu\|_p^2 \tag{3}$$

$$s.t.\begin{cases} (\langle \tilde{w}, K(x_i)\rangle) + b - y_i \le \varepsilon + \xi_i, \\ y_i - (\langle \tilde{w}, K(x_i)\rangle) + b \le \varepsilon + \hat{\xi}_i, \\ \tilde{\mu} > 0, \\ \mu_1, \mu_2, \ldots, \mu_M \ge 0, \\ \xi_i, \hat{\xi}_i \ge 0, i = 1, 2, \ldots N \end{cases} \tag{4}$$

Above, $\tilde{w}, b$ represents hyperplanes mapped from low dimensions and $\|\mu\|_p^2$ represents a weight coefficient of the L_p normal form. The optimization of the objective function and its constraints are as follows:

$$\min_{\tilde{w},b} \frac{1}{2}\sum_{k} \frac{\|w_k\|_2^2}{\mu_k} + C\sum_{i=1}^{l}(\xi_i + \hat{\xi}) \tag{5}$$

$$s.t.\begin{cases} (\langle \tilde{w}, K(x_i)\rangle) + b - y_i \le \varepsilon + \xi_i, \\ y_i - (\langle \tilde{w}, K(x_i)\rangle) + b \le \varepsilon + \hat{\xi}_i, \\ \tilde{\mu} > 0, \\ \mu_1, \mu_2, \ldots, \mu_M \ge 0, \\ \xi_i, \hat{\xi}_i \ge 0, i = 1, 2, \ldots N \end{cases} \tag{6}$$

Lagrange multipliers are introduced to optimize the function and its constraints:

$$\begin{aligned} \min_{\mu}\max_{\hat{\alpha},\alpha} &\left\{\frac{1}{2}\sum_{i,j=1}^{l} y_i(\hat{\alpha}_i - \alpha_i) - \varepsilon\sum_{i=1}^{l}(\hat{\alpha}_i - \alpha_i)\right\} \\ &- \left\|\frac{1}{2}\sum_{i=1}^{n}\mu_k\sum_{i=1}^{l}\sum_{j=1}^{l}(\hat{\alpha}_i - \alpha_i)\cdot(\hat{\alpha}_j - \alpha_j)K_{\mathrm{MRL}_p}(x_i, x_j)\right\|_p \end{aligned} \tag{7}$$

$$s.t.\begin{cases} \sum_{i=1}^{l}(\hat{\alpha} - \alpha_i) = 0 \\ 0 \le \hat{\alpha}, \alpha_i \le C, \quad i = 1, 2, \ldots, l \\ \mu_n \ge 0, \mu = 1, 2, \ldots, m \end{cases} \tag{8}$$

In the above formula,

$$K_{MKL_p}(x_i, x_j) = \sum_{n=1}^{m}\mu_n k_n(x_i, x_j) \tag{9}$$

$$p^* = \frac{p}{p-1} \tag{10}$$

Finally, the expression of the optimal regression function is shown below:

$$f(x) = \sum_{i=1}^{n}(\alpha_i^* - \alpha_i)K_{MKL_p}(x_i, x_j) + b^* \tag{11}$$

4.3.2. Double-Layer Alternating Optimization Parameters

The selection of the kernel function weight determines the proportion of the influence of each influencing factor. Selecting the appropriate weight coefficient can effectively adjust the role of each kernel function in the regression function and control the influence of heteroscedasticity. Therefore, this section uses the double-layer iterative method to solve the optimization of the kernel function weight, Lagrange multiplier, and parameters alternately. Until the stopping time criterion is satisfied, the optimal regression function is obtained. Compared to other methods, the biggest advantage of this method is that the weight coefficient can be obtained by learning to dynamically adjust the proportion of each single kernel function.

In the first-layer kernel learning, according to the objective function and constraint conditions based on the normal LP form in Equation (4), the Lagrange multiplier is kept unchanged, and the optimal weight coefficient is calculated. The calculation process is shown in Equation (12):

$$\begin{aligned} L = & \sum_{i,j=1}^{l} y_i(\hat{\alpha}_i - \alpha_i) - \varepsilon \sum_{i=1}^{l}(\hat{\alpha}_i - \alpha_i) \\ & - \frac{1}{2}\sum_{i=1}^{n}\mu_k \sum_{i=1}^{l}\sum_{j=1}^{l}(\hat{\alpha}_i - \alpha_i)\cdot(\hat{\alpha}_j - \alpha_j)K_{MKL_p}(x_i, x_j) \\ & + \beta\left(\frac{1}{2}\|\mu\|_p^2 - \frac{1}{2}\right) - \gamma^T\mu \end{aligned} \tag{12}$$

After finding the partial derivative of L, we can obtain Equation (13):

$$\begin{aligned} \mu_k = \frac{1}{2\beta}\Bigg(& \sum_k \sum_{i=1}^{1}\sum_{j=1}^{1}(\widehat{\alpha}_i - \alpha_i)(\widehat{\alpha}_j - \alpha_j)k_{MKL_p}(X_i, X_j)\Bigg)^{\frac{1}{q}-\frac{1}{p}} \\ & \times \left(\sum_{i=1}^{1}\sum_{j=1}^{1}(\widehat{\alpha}_i - \alpha_i)(\widehat{\alpha}_j - \alpha_j)k_{MKL_p}(X_i, X_j)\right) \end{aligned} \tag{13}$$

In the above formula,

$$\frac{1}{p} + \frac{1}{q} = 1, k = 1, 2, \ldots, k \tag{14}$$

In the second layer, with the weight vector unchanged, the Lagrange multipliers are solved by the improved sequential minimum optimization algorithm, and then the gradient descent method is used to optimize alternately, until the termination condition is satisfied. The optimization steps are as follows if $\widehat{\alpha}_i = \widehat{j}_i = \widehat{j}_{hi} = 0$ and $\alpha_i = \widehat{f}_i = f_{hi}\ \forall i = 1, 2, \ldots, N$.

Step 1: initialize parameters; assume that $\mu_k = \sqrt[p]{\frac{1}{m}}$ and L $= s = \infty$; the threshold $\varepsilon \leq \left|1 - \frac{L-S}{L_x - S_x}\right|$.

Step 2: using a sequential minimum optimization algorithm to generate the working set X, $\widehat{\alpha}_X = \widehat{\alpha}_{Xi1}, \widehat{\alpha}_{Xi2}, \ldots, \widehat{\alpha}_{XiX}$; $\alpha_X = \alpha_{Xi1}, \alpha_{Xi1}, \ldots, \alpha_{XiX}$.

Step 3: $\widehat{\alpha}_X^{old} = \widehat{\alpha}$; $\widehat{\alpha}_X^{old} = \alpha$; update the Lagrange multiplier with Formula (2).

Step 4: update the gradient by the gradient descent method.

Step 5: calculate the step size S_k: $S_k = \frac{1}{2}\sum_{i=1}^{N}(f_{k,i} - g_{k,i})(\widehat{\alpha}_i - \alpha_i)$; $\forall k = 1, 2, \ldots, m$.

Step 6: $L_x = \sum_i y_i(\widehat{\alpha}_i - \alpha_i) - \sum_i \varepsilon(\widehat{\alpha}_i + \alpha_i)$, $S_x = \sum_k d_k S_k$.

Step 7: if the threshold termination condition is met, bring two multipliers into (6) and calculate the final Lagrange multipliers α_i^*, α_i and weight vector μ; otherwise, return to Step 3.

Overall, the proposed SAMD prediction model contains two main processing, multi-density clustering, and multi-kernel support vector machine models, which can better meet

the time variability of real-time electricity prices for the dynamic change in demand in the prediction model. Moreover, the multi-kernel function is used to analyze the influencing factors, which effectively solves the problem of unsatisfactory prediction accuracy caused by various characteristics and influencing factors of real-time electricity prices. The overall process is shown in Figure 6.

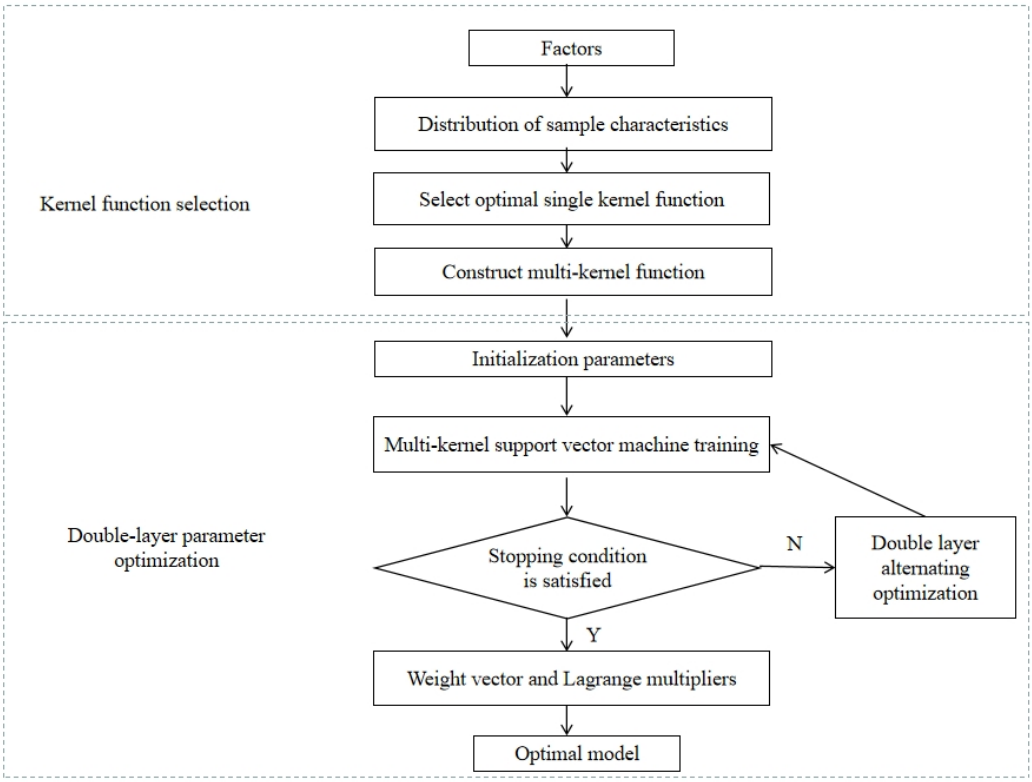


Figure 6. SAMD-SVM forecasting flowchart.

5. Experiments

DBSCAN [30], K-means clustering [31], and the proposed SAMD model were compared. The testing dataset was PJM 2018, which was 1 h for the time interval of the electricity price monitoring time. A full understanding of the characteristics of real-time electricity prices was essential for the selection of the forecasting model, and real-time electricity prices were found to have the following characteristics: periodicity, volatility, mean reversion, and heteroscedasticity. Taking real-time electricity prices in the PJM electricity market for the two adjacent weeks of 1–14 February 2018 as an example, it can be seen from Figure 7 that the electricity prices exhibited strong daily and weekly cycle characteristics. Mean reversion was reflected in the fact that when the load supply exceeded demand, the price of electricity was on an upward trend; when the load supply exceeded demand, the price of electricity was on a downward trend, and the average price of electricity tended to be close to the cost price. In order to better understand the characteristics of real-time tariffs, this paper statistically analyzed the real-time tariff data in terms of the six dimensions in Table 1. The JB was mainly used to test whether the electricity price obeyed a normal distribution, and $Q^2(10)$ was mainly used to test whether the electricity price had heteroskedasticity. From the statistical data in Table 1, it can be seen that real-time tariffs had a skewness of not 0 and a kurtosis of more than 3, which was the phenomenon of "sharp peaks and thick tails", and it can be concluded that the real-time tariffs had heteroskedasticity.

Table 1. Statistical table of real-time electricity price characteristics in electricity market.

Statistic	PJM
Mean	0.018
Standard Deviation	24.143
Skewness	0.512
Peakedness	14.975
JB	2289.879
$Q^2(10)$	8.971 (<0.01)

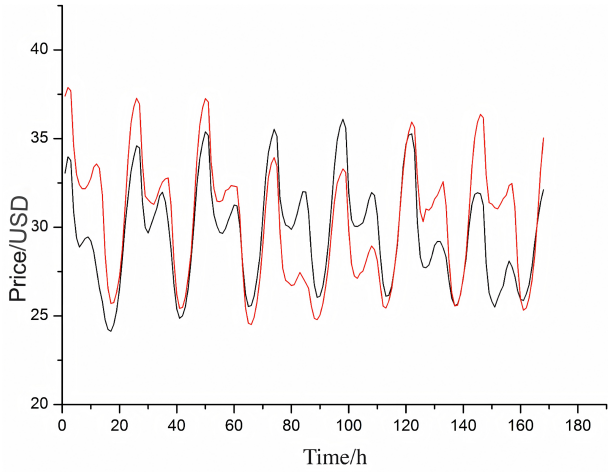


Figure 7. Real-time electricity price fluctuations.

5.1. Multi-Density Clustering

The results show that DBSCAN and K-means clustering did not focus on the factor of noise spots, which included noise points and boundary points. However, the SAMD model with timespan processing not only dealt with the problem of excluding the noise points due to the threshold of the set of MITs but also classified a few of the closest noise points into sparse clusters.

Normally, two widely used measures are chosen to evaluate clustering accuracy. One is normalized mutual information (NMI) [32] and the other is the NCI Cluster. The computation of NMI consists of two main components: mutual information (MI) and entropy. By standardizing the value of MI, it provides an easy-to-understand and comparable measure of how well the clustering results match the true labels. The advantage of NMI is that it combines precision and recall, making it a comprehensive and commonly used metric for clustering evaluation. The value of NMI equaling 1 represents that the clustering result exactly matches the labels, and it being 0 signifies that the result is completely irrelevant. The performance results of the SAMD, DBSCAN, and K-means models are shown in Table 2; the SAMD model was more accurate than the other methods.

Running Time Efficiency: We discuss the changes in the running time regarding an increasing dataset. The general trend was that the running time continued to grow as datasets were added. However, our proposed multi-density method showed a more stable state. Our proposed SAMD model was more efficient than the other DBSCAN and K-means algorithms, as shown in Figure 8.

Table 2. Clustering accuracy evaluation results.

Algorithm	SAMD	DBSCAN	K-Means
NMI	0.925	0.820	0.723

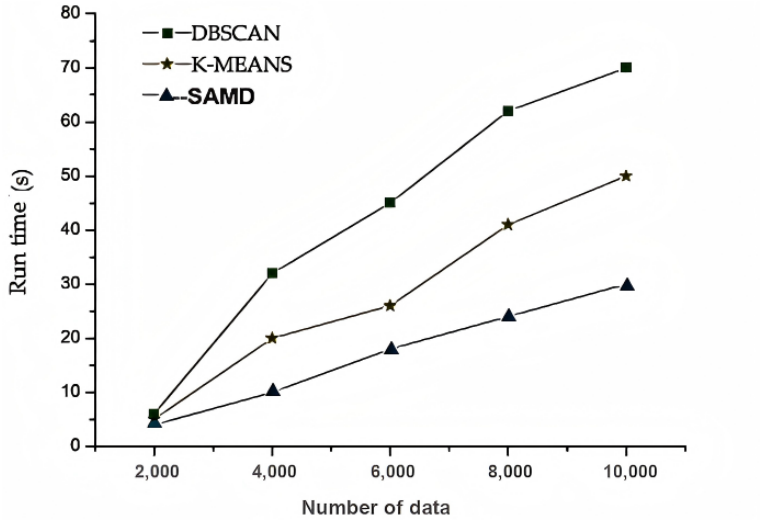


Figure 8. Clustering run-time contrast diagram.

5.2. Multi-Kernel Support Vector Machine Forecasting Evaluation

According to the characteristics of real-time electricity prices, we selected some open and quantifiable influencing factors to establish the multi-kernel support vector machine prediction model. Then, we evaluated the prediction results with conventional neural network prediction algorithms in the same datasets. Finally, the avoided costs were calculated to prove the effectiveness of the proposed algorithm.

Four kinds of kernel function curve graphs are shown in Figure 9. The analysis of the data should be based on the characteristics of the data themselves to choose the appropriate kernel function, so this paper adopted the linear multi-kernel function. In Figure 9, σ is the hyperparameter of the radial basis kernel and d is the number of highest terms in the polynomial kernel function, which defaulted to 3.

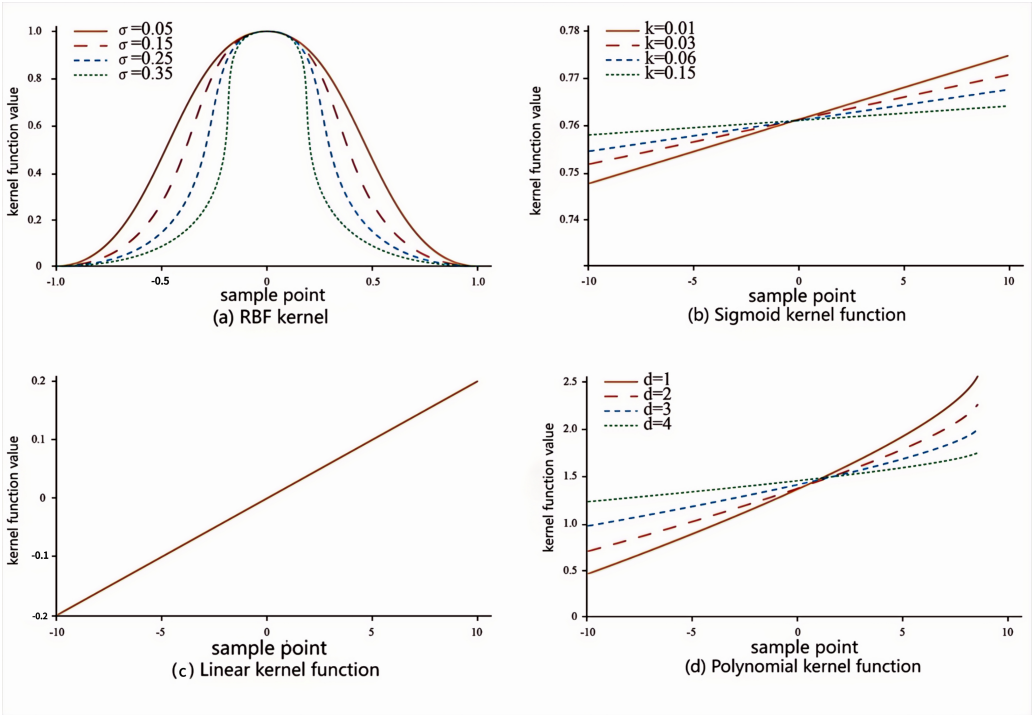


Figure 9. Kernel function curve graph.

5.3. Real-Time Electricity Price Prediction Results

5.3.1. Predication Accuracy

The real-time electricity price datasets of PJM April 2018 [33] were used to test the proposed algorithm; the accuracy results are shown in Table 3. The proposed algorithm was more accurate than the neural network [34]; the results are shown in Figure 10.

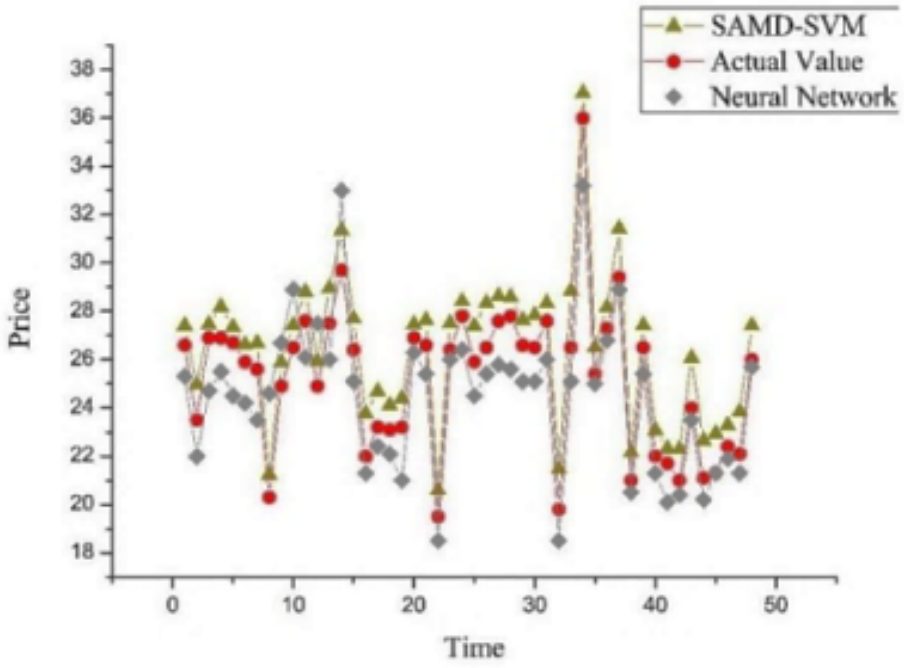


Figure 10. Prediction accuracy contrast diagram.

Table 3. Electricty price prediction comparison.

Time	Actual Value	SAMD-SVM Predicted Value	SAMD-SVM Error (%)	Neural Network Predicted Value	Neural Network Error (%)
01:00	17.506	18.925	3.32	16.716	4.51
02:00	15.42	14.919	3.25	14.695	4.70
03:00	15.142	14.786	2.33	14.486	4.33
04:00	18.632	18.134	2.67	18.025	3.26
05:00	18.642	18.411	1.24	17.892	4.02
06:00	21.098	20.489	2.89	20.410	3.26
07:00	21.162	21.331	0.75	20.836	3.75
08:00	21.062	20.879	0.87	20.482	2.72
09:00	21.263	20.842	1.98	20.756	2.38
10:00	20.280	20.226	0.27	19.827	2.23
11:00	20.534	19.894	3.12	19.827	3.44
12:00	21.2908	20.435	4.02	20.647	3.02
Average error		2.45		3.54	

5.3.2. Avoid Costs

Normally, we use avoided costs to evaluate electricity price strategies and benefits, which express that power suppliers can avoid or partially avoid power supply resources and costs.

$$B_{se} = \frac{C - C_d}{\Delta W} \tag{15}$$

C and C_d represent the original power supply cost and cost after real-time electricity implementation, respectively. ΔW represents power saving, and the results are shown in Table 4.

Table 4. Avoided electricity costs comparison.

Time	SAMD-SVM	Neural Network
01:00	0.323	0.209
02:00	0.294	0.262
03:00	0.452	0.327
04:00	0.486	0.316
05:00	0.414	0.312
06:00	0.512	0.184
07:00	0.394	0.322
08:00	0.792	0.266
09:00	0.242	0.215
10:00	0.196	0.231
11:00	0.438	0.342
12:00	0.493	0.286
Avoid costs	0.419	0.272

6. Conclusions

The setting of electricity prices is closely related to the interests of each participant, and accurate electricity price prediction can help participants understand future trends; however, due to the interference of many influencing factors, prediction accuracy is not ideal, so how to improve the prediction accuracy of real-time electricity prices has become an important direction in current research. The proposed SAMD model can categorize a time-series dataset into quality clusters of high, medium, and low levels, which can be represented as different kinds of real-time electricity price intervals to predict the general trends of the prices. The experimental results were used to analyze and verify the advantageous model to predict the real-time electricity prices, and it is suitable for different types of real datasets.

However, the method in this paper also has limitations: the feature quantization relies on human settings and lacks adaptive learning ability, which may lead to processing bias and then affect prediction. We plan to extend this work to develop a multidimensional data attribute calculation model for mining the influence of indirect factors on real-time electricity prices in our future work.

Author Contributions: Conceptualization, L.W., T.H.Z. and W.D.; formal analysis, T.H.Z., X.X. and G.H.; funding acquisition, L.W., T.H.Z. and L.K.; investigation, X.X. and L.K.; methodology, T.H.Z., X.X. and G.H.; software, G.H. and W.D.; validation, T.H.Z., X.X. and G.H.; resources, L.W., T.H.Z. and L.K.; data curation, G.H. and X.X.; writing—original draft, T.H.Z., X.X. and G.H.; writing—review and editing, T.H.Z., X.X. and L.W.; visualization, X.X. and G.H.; supervision, L.W. and L.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science and Technology Development Plan of Jilin Province, China (Grant No. 20220402033GH), by the National Natural Science Foundation of China (No. 62406156), and by the Shandong Province Natural Science Foundation (No. ZR2023QF036).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: A publicly available dataset was analyzed in this study. The PJM April 2018 dataset can be found here: https://dataminer2.pjm.com/feed/rt_hrl_lmps/definition (accessed on 14 November 2024).

Acknowledgments: The authors would like to thank the reviewers for their essential suggestions in helping improve this manuscript.

Conflicts of Interest: Author Gongliang Hu was employed by Qingdao Integrated Service Group for Talented Individuals and Enterprises. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Kezunovic, M.; Pinson, P.; Obradovic, Z.; Grijalva, S.; Hong, T.; Bessa, R. Big data analytics for future electricity grids. *Electr. Power Syst. Res.* **2020**, *189*, 106788. [CrossRef]
2. Zhang, Y.; Huang, T.; Bompard, E.F. Big data analytics in smart grids: A review. *Energy Inform.* **2018**, *1*, 8. [CrossRef]
3. Tschora, L.; Pierre, E.; Plantevit, M.; Robardet, C. Electricity price forecasting on the day-ahead market using machine learning. *Appl. Energy* **2022**, *313*, 118752. [CrossRef]
4. Liu, Y.; Sun, Y.; Wang, R.; Jiang, Q.; Geng, G.; Wang, X. Forecasting-based electricity tariff selection for resident users with photovoltaic and energy storage considering forecast uncertainties. *Authorea Prepr.* **2023**. [CrossRef]
5. Meng, A.; Wang, P.; Zhai, G.; Zeng, C.; Chen, S.; Yang, X.; Yin, H. Electricity price forecasting with high penetration of renewable energy using attention-based LSTM network trained by crisscross optimization. *Energy* **2022**, *254*, 124212. [CrossRef]
6. Taheri, S.; Jooshaki, M.; Moeini-Aghtaie, M. Long-term planning of integrated local energy systems using deep learning algorithms. *Int. J. Electr. Power Energy Syst.* **2021**, *129*, 106855. [CrossRef]
7. Kuo, P.H.; Huang, C.J. An electricity price forecasting model by hybrid structured deep neural networks. *Sustainability* **2018**, *10*, 1280. [CrossRef]
8. Li, Y.; Pizer, W.A.; Wu, L. Climate change and residential electricity consumption in the Yangtze River Delta, China. *Proc. Natl. Acad. Sci. USA* **2019**, *116*, 472–477. [CrossRef] [PubMed]
9. Holder, C.; Middlehurst, M.; Bagnall, A. A review and evaluation of elastic distance functions for time series clustering. *Knowl. Inf. Syst.* **2024**, *66*, 765–809. [CrossRef]
10. Cook, A.A.; Mısırlı, G.; Fan, Z. Anomaly detection for IoT time-series data: A survey. *IEEE Internet Things J.* **2019**, *7*, 6481–6494. [CrossRef]
11. Jiang, C.; Liebman, A.; Wang, H. Network-Aware Electric Vehicle Coordination for Vehicle-to-Anything Value Stacking Considering Uncertainties. In Proceedings of the 2023 IEEE/IAS 59th Industrial and Commercial Power Systems Technical Conference (I&CPS), Las Vegas, NV, USA, 21–25 May 2023; pp. 1–9.
12. Alfaverh, F.; Denai, M.; Sun, Y. A dynamic peer-to-peer electricity market model for a community microgrid with price-based demand response. *IEEE Trans. Smart Grid* **2023**, *14*, 3976–3991. [CrossRef]
13. Tomin, N.; Shakirov, V.; Kozlov, A.; Sidorov, D.; Kurbatsky, V.; Rehtanz, C.; Lora, E.E. Design and optimal energy management of community microgrids with flexible renewable energy sources. *Renew. Energy* **2022**, *183*, 903–921. [CrossRef]
14. Alzoubi, A. Machine learning for intelligent energy consumption in smart homes. *Int. J. Comput. Inf. Manuf. (IJCIM)* **2022**, *2*. [CrossRef]
15. Rehman, M.A.; Numan, M.; Tahir, H.; Rahman, U.; Khan, M.W.; Iftikhar, M.Z. A comprehensive overview of vehicle to everything (V2X) technology for sustainable EV adoption. *J. Energy Storage* **2023**, *74*, 109304. [CrossRef]
16. Jaeger, A.; Banks, D. Cluster analysis: A modern statistical review. *Wiley Interdiscip. Rev. Comput. Stat.* **2023**, *15*, e1597. [CrossRef]
17. Eminagaoglu, M. A new similarity measure for vector space models in text classification and information retrieval. *J. Inf. Sci.* **2022**, *48*, 463–476. [CrossRef]
18. Zhang, Q.; Zhang, C.; Cui, L.; Han, X.; Jin, Y.; Xiang, G.; Shi, Y. A method for measuring similarity of time series based on series decomposition and dynamic time warping. *Appl. Intell.* **2023**, *53*, 6448–6463. [CrossRef]
19. Al-Barakati, A.; Mishra, A.R.; Mardani, A.; Rani, P. An extended interval-valued Pythagorean fuzzy WASPAS method based on new similarity measures to evaluate the renewable energy sources. *Appl. Soft Comput.* **2022**, *120*, 108689. [CrossRef]
20. Prakaisak, I.; Wongchaisuwat, P. Hydrological time series clustering: A case study of telemetry stations in Thailand. *Water* **2022**, *14*, 2095. [CrossRef]
21. Li, H.; Liu, Z.; Wan, X. Time series clustering based on complex network with synchronous matching states. *Expert Syst. Appl.* **2023**, *211*, 118543. [CrossRef]

22. Li, H.; Liu, J.; Yang, Z.; Liu, R.W.; Wu, K.; Wan, Y. Adaptively constrained dynamic time warping for time series classification and clustering. *Inf. Sci.* **2020**, *534*, 97–116. [CrossRef]
23. Yu, C.; Luo, L.; Chan, L.L.H.; Rakthanmanon, T.; Nutanong, S. A fast LSH-based similarity search method for multivariate time series. *Inf. Sci.* **2019**, *476*, 337–356. [CrossRef]
24. Ma, R.; Angryk, R. Distance and density clustering for time series data. In Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, USA, 18–21 November 2017; pp. 25–32.
25. Du, L.; Gao, R.; Suganthan, P.N.; Wang, D.Z. Bayesian optimization based dynamic ensemble for time series forecasting. *Inf. Sci.* **2022**, *591*, 155–175. [CrossRef]
26. Sotiriadis, M.S.; Tsotsos, R.; Kosmidou, K. Price and volatility interrelationships in the wholesale spot electricity markets of the Central-Western European and Nordic region: A multivariate GARCH approach. *Energy Syst.* **2016**, *7*, 5–32. [CrossRef]
27. Alswaitti, M.; Albughdadi, M.; Isa, N.A.M. Density-based particle swarm optimization algorithm for data clustering. *Expert Syst. Appl.* **2018**, *91*, 170–186. [CrossRef]
28. Sirisha, U.M.; Belavagi, M.C.; Attigeri, G. Profit prediction using ARIMA, SARIMA and LSTM models in time series forecasting: A comparison. *IEEE Access* **2022**, *10*, 124715–124727. [CrossRef]
29. Ezugwu, A.E.; Ikotun, A.M.; Oyelade, O.O.; Abualigah, L.; Agushaka, J.O.; Eke, C.I.; Akinyelu, A.A. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Eng. Appl. Artif. Intell.* **2022**, *110*, 104743. [CrossRef]
30. Deng, D. DBSCAN clustering algorithm based on density. In Proceedings of the 2020 7th International Forum on Electrical Engineering and Automation (IFEEA), Hefei, China, 25–27 September 2020; pp. 949–953.
31. Sinaga, K.P.; Yang, M.S. Unsupervised K-means clustering algorithm. *IEEE Access* **2020**, *8*, 80716–80727. [CrossRef]
32. Kvålseth, T.O. On normalized mutual information: Measure derivations and properties. *Entropy* **2017**, *19*, 631. [CrossRef]
33. Bowring, J.; Monitoring Analytics, L. 2017 State of the Market Report for PJM. *Tech. Rep.* **2018**. Available online: https://www.monitoringanalytics.com (accessed on 1 February 2025).
34. Wang, K.; Yu, M.; Niu, D.; Liang, Y.; Peng, S.; Xu, X. Short-term electricity price forecasting based on similarity day screening, two-layer decomposition technique and Bi-LSTM neural network. *Appl. Soft Comput.* **2023**, *136*, 110018. [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.