EAI Endorsed Transactions
on Energy Web

Research Article EAI.EU

Mining method of power consumption behavior pattern of different subject resource objects in virtual power plant based on graph neural network

DUAN Zhiguo[1], ZHAO Jie[1], CHEN Xi[1], AN Jiakun[2] and FAN Wenyi*[2]

[1]State Grid Hebei Electric Power Co., Ltd., 333 Jianhua South Street, Shijiazhuang, Hebei Province, China

[2]Economic and Technical Research Institute, State Grid Hebei Electric Power Co., Ltd., 27 Fuqiang Street, Yuhua District Shijiazhuang, Hebei Province, China

Abstract

INTRODUCTION: With the advancement of global energy internet and smart grid, massive power big data with complex temporal features is generated by widely deployed smart meters. Accurate identification of power consumption behavior patterns is essential for dispatching a virtual power plant (VPP).
OBJECTIVES: This paper proposes a fine-grained mining framework that combines temporal feature extraction and a graph neural network to support VPP dispatch requirements.
METHODS: Linear interpolation and same-type day mean filling are used for data cleaning and normalization. A user relationship graph is constructed via physical connection and Pearson correlation-based behavioral similarity. A GCN-based unsupervised graph autoencoder is adopted for spatio-temporal feature embedding, and an improved DPC-FCM clustering algorithm is proposed to optimize it.
RESULTS: The method effectively solves the inherent defects of traditional clustering: initial center sensitivity and noise interference.
CONCLUSION: This framework realizes accurate fine-grained mining of power consumption patterns, providing reliable support for VPP dispatch decision-making.

Keywords: Consumption Behavior Pattern Mining, Graph Neural Network (GNN), Density Peak Clustering(DPC), Fuzzy C-Means (FCM) Clustering Algorithm

Received on 23 March 2026, accepted on 12 May 2026, published on 16 June 2026

doi: 10.4108/ew.12319

*Corresponding author. Email: jyy_fanwy@163.com

1. Introduction

With the in-depth development of the global energy internet and smart grid construction, the explosive growth of distributed energy sources and flexible loads has driven the rapid development of Virtual Power Plant (VPP) technology [1-2]. The dispatching optimization mechanism for virtual power plants in the energy internet primarily comprises three aspects: natural aggregation, economic incentives, and operational coordination. In the process of natural aggregation, resource classification is crucial, and accurate identification of power consumption patterns is a prerequisite for virtual power plants to implement economic incentives and operational coordination.

Meanwhile, the widespread deployment of smart meters has completely changed the operational and data-collection modes of power systems. Traditional electricity metering data have been replaced by high-frequency, high-precision power load curve data [3]. These data not only exhibit the characteristics of massive volume and high speed typical of big data, but also show complex periodicity, randomness, and high-dimensional structure in the time-series dimension. Against the background of massive power big data, accurately and efficiently identifying potential and common power consumption behavior patterns from hundreds of millions of user power consumption records has become the key to supporting the transformation of the power system from supply-driven demand to source-grid-load-storage interaction and enabling virtual power plants to better play their aggregation role[4].

At present, most studies on the classification of power users' consumption behavior use clustering methods [5], including K-means [6], Fuzzy C-Means (FCM) [7], Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [8], and hierarchical clustering. Meanwhile, Bayesian classification is also used to analyze power consumption behavior. Ai Xin et al. [6] proposes a load curve clustering method that combines agglomerative hierarchical clustering and K-means clustering, but the number of clusters k lacks adaptive adjustment, and selecting k is relatively harsh. ZHAO T H et al. [7] introduces the power consumption time offset to correct the abnormal data domain on the basis of the spatial density clustering method, but the abnormal data domain of abnormal power consumption patterns relies on the boundary extrema of each load level of normal power consumption patterns, which has limited adaptability to special types of abnormal power consumption patterns. CAI X W et al. [8] proposes a clustering method based on attribute features of different power consumption behaviors, but the clustering attributes are not comprehensive enough, failing to directly incorporate core influencing factors such as family structure and income level, and only indirectly inferring through average daily power consumption. Many scholars have also improved clustering algorithms, including the Self-Organizing Map (SOM) neural network [9], the density-based spatial clustering [10], the curve clustering algorithm based on variance comparison [11], and the C-vine Copula mixture model [12]. Among them, the Fuzzy C-Means (FCM) clustering algorithm has been widely applied to the identification of power consumption behavior due to its simple principle, ease of implementation, and high efficiency..

The emergence of Graph Neural Networks (GNNs) provides a powerful tool for modeling user relationship networks. By representing customers and their relationships as a graph, GNNs can learn graph embedding vectors that capture both users' own load characteristics and their neighborhood structure, laying a foundation for deeper, more accurate pattern mining [13-14]. However, most existing studies focus on load forecasting scenarios, with very limited research on the mining of electricity consumption patterns in virtual power plants (VPPs). Furthermore, these studies fail to achieve in-depth integration of GNN graph embedding with clustering algorithms, and lack a complete closed loop from pattern mining to VPP scheduling support.

This paper aims to combine temporal feature extraction technology with graph neural network models to construct a fine-grained power consumption behavior pattern mining and demand-side response potential assessment framework for power customers. The specific research contents include: temporal feature and graph construction: extracting multi-dimensional temporal load features of customers and constructing a customer relationship graph based on potential customer relationships (such as similarity or geographical association); GNN-based graph embedding: using a graph neural network model to train the user graph and generate low-dimensional dense graph embedding vectors integrating temporal and structural information; improved pattern clustering: proposing an improved clustering algorithm integrating graph embedding vectors to achieve fine-grained classification and pattern profiling of customers' power consumption behavior; quantitative assessment of demand response potential: establishing a demand-side response potential assessment index system and prediction model based on the identified fine-grained behavior patterns to quantitatively assess the adjustable potential of customers with different patterns.

Compared with existing methods for power-user behavior analysis, the framework proposed in this paper offers significant improvements and advantages. Traditional clustering-based methods rely solely on single load curves and exhibit poor robustness to high-dimensional, sparse data. Meanwhile, they ignore the physical topological correlations and behavioral coupling among users within a VPP and suffer from inherent drawbacks, including vulnerability to noise interference and high sensitivity to the initial clustering centers. In this paper, by constructing a user relationship graph and applying GNNs, we deeply integrate physical transformer area connections with behavioral similarity and effectively map high-dimensional time-series data to low-dimensional dense graph embedding vectors, thereby capturing deeper topological correlations and user electricity consumption characteristics. Meanwhile, this paper innovatively introduces the Density Peak Clustering (DPC) algorithm to adaptively select initial centers and perform iterative optimization combined with graph-embedding features, effectively addressing the local optimum problem and significantly improving the robustness and discriminative power of clustering results. Finally, this study is not limited to identifying behavior patterns; it also constructs a complete closed-loop system that covers data cleaning, feature fusion, fine-grained clustering, and the quantitative evaluation of demand response potential. Compared with studies that focus solely on single clustering algorithms, our work can provide more scientific, quantitative decision support for resource aggregation, precision marketing, and power grid planning for VPPs.

2. Power Consumption Pattern Mining Architecture

To effectively mine the power consumption behavior patterns of flexible resources in virtual power plants, this paper proposes a hierarchical mining architecture based on graph neural networks. The architecture aims to extract spatio-temporal features from multi-source heterogeneous data and ultimately serve the regulation strategy of VPPs. The overall system framework consists of four layers in ascending order: the data perception and processing layer, the graph structure construction and feature mining layer, the clustering analysis layer, and the business application layer. Its overall structure can be expressed as the quaternary structure diagram in Fig. 1.

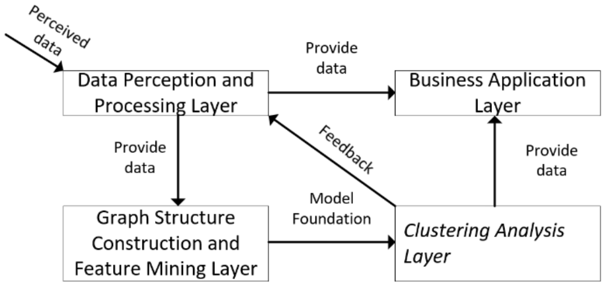

Figure 1. System quaternary structure

2.1. Data Perception and Processing Layer

As the system's physical foundation, this layer is primarily responsible for collecting and cleaning multi-source data. The input of the system mainly includes two types of data: one is flexible resource load data, which collects real-time active and reactive power sequences of resources such as energy storage, air conditioners, and electric vehicles within the virtual power plant's jurisdiction through intelligent measurement devices; the other is external environmental feature data, including real-time temperature, relative humidity, time-of-use electricity price period identification, and holiday type in the region. To address missing values and outliers in the original data, statistical-based preprocessing methods are used to clean the data and ensure its quality.

2.2. Graph Structure Construction and Feature Mining Layer

To address the limitation of traditional methods ignoring the topological associations between resources, this layer first constructs a spatio-temporal association graph $G=(V,E)$ of flexible resources using graph theory methods, where nodes V represent various heterogeneous resource entities, and edges E map the coupling relationships between resources in terms of spatial location or electrical connection. On this basis, a graph neural network model is deployed as the core algorithm to extract implicit high-dimensional spatio-temporal feature vectors by aggregating feature information of neighboring nodes.

2.3. Clustering Analysis Layer

To address the problem that the traditional Fuzzy C-Means clustering algorithm is sensitive to initial values, this paper improves it by selecting initial centers based on density peaks. The low-dimensional embedding vectors output by the graph neural network are clustered into sets with similar features, thereby identifying typical power consumption behavior patterns for different flexible resources in the virtual power plant.

2.4. Business Application Layer

This layer uses the Naive Bayes algorithm for behavior pattern analysis and mining, and converts the results into a specific dispatching decision basis. The system's output mainly includes classification results for resource power consumption patterns, i.e., the probability that each resource belongs to a specific typical pattern at the current moment, to assist the virtual power plant in formulating an optimal aggregation control strategy.

3. Power Big Data Preprocessing and Power Consumption Feature Extraction

Smart meters are key components of smart grids, which can monitor and collect power load data in real time from power users. Smart meter data are usually collected at 15-minute or 1-hour intervals, with extremely high dimensions. Due to communication failures or equipment abnormalities, data often have missing values and noise. At the same time, affected by various factors such as meteorological conditions and holidays, load data exhibit clear non-stationary random process characteristics.

3.1. Power Big Data Preprocessing

For short-term missing values, linear interpolation is used to supplement the data, assuming that power consumption between two known data points changes linearly with time, and estimating the missing values by constructing a linear equation. This method has the advantages of simple calculation and a high degree of temporal trend reduction, and is suitable for scenarios with short missing durations and stable data trends.

Assume that the data x_i at time t_i is missing, and the two nearest moments being t_m and t_k,with corresponding data are x_m and x_k.Then the following formula holds:

$$x_i = x_k + \frac{x_m - x_k}{t_m - t_k} \cdot \left(t_i - t_k\right) \tag{1}$$

For long-term missing values, the same-type day mean filling method is used. Its principle is that users' power consumption behavior has significant periodicity. For example, the power consumption pattern on working days is quite different from that on weekends, and the power consumption curves of the same day of the week each week have high similarity. The core of the same-type day mean filling method is to select the same-type days with the same power consumption characteristics as the missing day, and use the mean value of the corresponding moments of the same-type days as the filling value. It is suitable for scenarios with long missing durations and broken temporal trends.

Since users' power consumption habits do not change significantly in the short term, priority is given to selecting dates of the same day of the week as the missing day within the past three weeks, but holidays and extreme weather days that significantly affect the load should be excluded. The final filling value is calculated by formula (2). Figure 2 is a schematic diagram of missing data filling.

$$x_i = \frac{1}{K} \sum_{s \in S} y_{s,i} \tag{2}$$

Where x_i is the filling value of data at time i; $S = \{s_1, s_2, \cdots, s_k\}$ is the selected set of same-type days; K is the number of same-type days (recommended $K \geq 3$); $y_{s,i}$ is the arithmetic mean of the same-type days at time i.

Affected by signal interference, software failures, equipment performance, and other factors, load data may have incomplete collection or distortion. Based on the characteristic that data have similarity horizontally in a short time, that is, the sample day has a similar shape to the same-type curves in the nearby time period, this paper uses the 3σ principle to judge abnormal data. At the same time, there is noise in the load data during collection and transmission, so it is necessary to reduce noise through smoothing processing. This paper uses the moving average method to smooth the load data.

Assume that the load data of N observation days of a power user are selected, and the average value of the load data at all the i-th observation time points of the N observation days is:

$$\overline{x_i} = \frac{1}{N} \sum_{n=1}^{N} x_{n,i} \tag{3}$$

Where N is the total number of observation days; $x_{n,i}$ is the load data at the i-th observation time point of the n-th observation day.

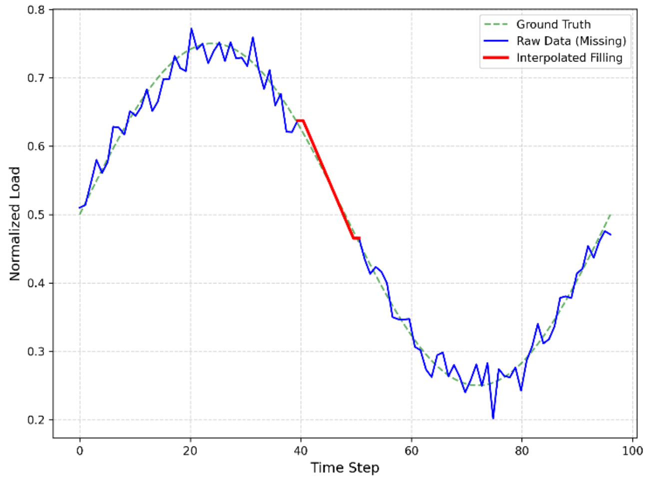

Figure 2. Schematic diagram of missing data filling

The variance of the load data at the i-th observation time point of all observation days of the user is:

$$\sigma_i^2 = \frac{1}{N} \sum_{n=1}^{N} \left(x_{n,i} - \overline{x_i} \right)^2 \tag{4}$$

Based on the 3σ principle, the judgment criterion for abnormal data is:

$$\left| x_{n,i} - \overline{x_i} \right| > 3\sigma_i \varepsilon \tag{5}$$

Where ε is the threshold, usually taking 1.0~1.5.

If formula (5) is satisfied, the data can be judged as abnormal. For these data, formula (6) can be used for correction.

$$\hat{x}_{n,i} = \frac{\alpha}{2}\left(x_{n+1,i} + x_{n-1,i} \right) + \frac{\beta}{2}\left(x_{n,i}^1 + x_{n,i}^2 \right) + \gamma \overline{x_i} \tag{6}$$

Where α, β, γ are weight coefficients satisfying $\alpha+\beta+\gamma=1$; $\hat{x}_{n,i}$ is the corrected data at the i-th observation time point of the n-th day; $x_{n+1,i}$ and $x_{n-1,i}$ are 2 horizontal load points near $x_{n,i}$; $x_{n,i}^1$ and $x_{n,i}^2$ are 2 nearest similar day load points to $x_{n,i}$.

By averaging ζ consecutive data near all load points using formula (7), a smoothed load sequence can be formed, which can remove the highly random burr data during data collection. Figure 3 is a schematic diagram of data smoothing correction.

$$x_{n,i}^{'} = \frac{1}{\zeta} \sum_{j=-2}^{2} x_{n,i+j} \tag{7}$$

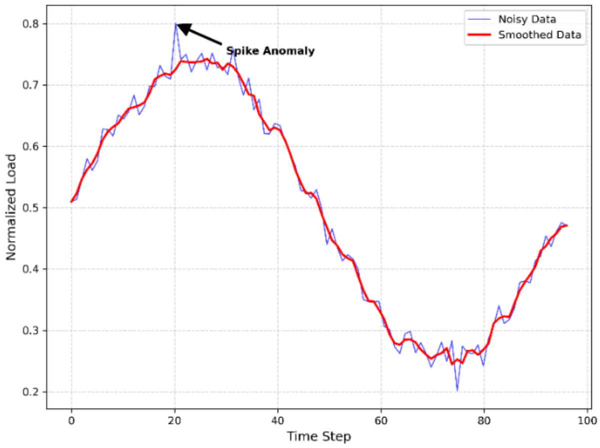

Figure 3. Smooth correction diagram

After completing the abnormal identification and correction of the massive power users' power load data, due to the significant differences in power consumption habits between different users and load characteristics in different time periods, these differences may lead to a decline in model performance during data mining. Therefore, it is necessary to normalize the power users' power load data, and the expression formula is:

$$X = \frac{X^{'} - X_{\min}}{X_{\max} - X_{\min}} \tag{8}$$

Where X' and X represent the power users' power load data before and after processing, respectively; X_{min} and X_{max} represent the minimum and maximum values of the original power users' power load data, respectively. After the above series of collection and preprocessing steps, this paper can obtain a set of high-quality and standardized power users' power load data sets, laying a foundation for subsequent user behavior pattern mining and prediction.

3.2. User Feature Extraction and Relationship Graph Construction

Due to the limitation that a single feature cannot fully describe the complex power consumption rules of users, a high-dimensional feature matrix X is constructed to comprehensively characterize user behavior.

Temporal shape features aim to capture the evolution trend and the instantaneous fluctuations in users' power consumption behavior over time. Given that the power grid typically uses a 15-minute sampling interval, this paper selects the typical daily load curve for users over a specific observation period. The 24 hours of a day are divided into 96 time points to form a feature vector.

Constructing the graph $G=(V,E)$ is a prerequisite for applying GNN. Where $V = \{v_1, v_2, \cdots v_N\}$ represents the set of N user nodes; E represents the set of edges.

The electricity consumption behaviors of users within a virtual power plant are jointly affected by physical electrical coupling and behavioral feature coupling. Users in the same transformer area share the distribution transformer, with highly correlated supply voltage and line loss characteristics, resulting in inherent physical correlations in their power consumption patterns. Meanwhile, users in different transformer areas with highly similar electricity consumption characteristics should also be effectively identified.

This paper adopts a method of fusing physical connections and behavioral similarity to calculate weighted edges.When considering physical connections, based on the power supply transformer district information, if users belong to the same transformer district, there is a connection edge, and the edge weight is set to 1.When considering behavioral similarity, to measure the linear correlation degree between user i and user j in the power consumption time series, the Pearson correlation coefficient of the load sequences between users is calculated:

$$\omega_{ij} = \frac{\operatorname{cov}\left(x_i, x_j\right)}{\sigma_{x_i}\sigma_{x_j}} = \frac{\sum_{t=1}^{T}\left(x_{i,t} - \bar{x}_i\right)\left(x_{j,t} - \bar{x}_j\right)}{\sqrt{\sum_{t=1}^{T}\left(x_{i,t} - \bar{x}_i\right)^2}\sqrt{\sum_{t=1}^{T}\left(x_{j,t} - \bar{x}_j\right)^2}} \tag{9}$$

A threshold ε is set, and a connection is established when $\omega_{ij} > \varepsilon$,otherwise it is set to 0.

Finally, according to the construction rule of the adjacency matrix, the adjacency matrix A is established. If there is a physical connection between user i and user j, then $A_{ij} = 1$; if there is behavioral similarity, then $A_{ij} = \omega_{ij}$;otherwise, it is set to 0. The adjacency matrix A is a symmetric matrix with all diagonal elements set to 0, which integrates physical topology and behavioral similarity, providing rich contextual information for GNN.

4. Graph Neural Network Model Construction and Graph Embedding Representation

This section details the feature learning framework based on Graph Convolutional Network (GCN). Considering that power users lack explicit category labels, this study adopts an unsupervised Graph Autoencoder (GAE) architecture. The architecture uses GCN as the encoder, aiming to learn a low-dimensional, dense vector representation integrating spatio-temporal features, providing high-quality input for subsequent clustering analysis.

This work aims to learn a mapping function $f(A,X) \rightarrow Z$, that embeds high-dimensional sparse graph data into a low-dimensional latent representation matrix $Z \in R_{N \times d}$ (where $d \ll F$), so that Z can maximally preserve both the topological structure and node attributes of the original graph.

4.1. Graph Convolutional Encoder Design

Conventional Convolutional Neural Networks (CNNs) are only applicable to data in Euclidean space, thus failing to directly handle graph data residing in non-Euclidean domains. In this work, we employ a graph convolutional network derived from Spectral Graph Theory.

To capture local and high-order dependencies, this paper designs a two-layer GCN model:

First layer: The original feature X is mapped to the hidden space using the graph convolutional network method. The propagation rule of the graph convolutional layer is based on the hierarchical propagation rule proposed by Kipf&Welling in 2017[15]. GCN updates the representation of the central node by aggregating the information of neighboring nodes.

$$H^{(1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}XW^{(0)}\right) \tag{10}$$

Where $H^{(l)}$ is node feature matrix of the l-th layer. When it is input layer, $H^{(0)}=X$;

$\tilde{A} = A = I_N$ is adjacency matrix with self-loops added. The purpose of introducing self-loops is to ensure that the node's own information can be retained during feature aggregation, rather than only including neighboring information.

$\tilde{D}$ is the degree matrix of $\tilde{A}$,which is a diagonal matrix, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$;

$\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ is crucial, which the porpose is symmetric normalization of the adjacency matrix. It can prevent the feature values of nodes with large degrees from being too large, leading to gradient explosion, and avoid the information of nodes with small degrees being submerged.

$W^{(l)} \in R^{d_l \times d_{l+1}}$ is the trainable weight matrix of the l-th layer.

$\sigma(x)$ is a nonlinear activation function, $RELU(f(x) = \max(0, x))$ is selected to enhance the nonlinear expression ability of the model.

Second layer: The information in the hidden space is mapped to the output layer to generate the final node embedding matrix Z.

$$Z = H^{(2)} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(1)}W^{(1)} \tag{11}$$

Through a two-layer GCN encoder, the finally obtained matrix $Z \in R_{N\times d}$ is the graph embedding representation integrating users' own power consumption behavior and users' social and physical relationships.

4.2. Decoder and Loss Function Construction

Since it is unsupervised learning, we need to construct a reconstruction task to train the network parameters *W(0)* and *W(1)*[16].

Structure Decoder: To ensure that the embedding vector Z can retain the topological structure of the original graph, we use the inner product decoder to reconstruct the adjacency matrix:

$$\hat{A} = sigmoid\left(ZZ^T\right) \tag{12}$$

Where $\hat{A}_{ij}$ represents the predicted probability that there is a connection between user i and user j.

Attribute Decoder: To ensure that Z also retains the users' load features, the fully connected layer can be used to reconstruct the original feature matrix:

$$\hat{X} = MLP(Z) \tag{13}$$

The overall loss function L of the joint optimization objective model consists of reconstruction errors. In this study, the focus is on the reconstruction of the topological structure, and the weighted cross-entropy loss function is adopted:

$$L = -\sum_{i=1}^{N}\sum_{j=1}^{N}\left[\omega \cdot A_{ij}\log\left(\hat{A}_{ij}\right) + \left(1 - A_{ij}\right)\log\left(1 - \hat{A}_{ij}\right)\right] + \lambda\left\|X - \hat{X}\right\|^2 \tag{14}$$

Where ω is the penalty weight for positive samples, which is used to solve the sample imbalance problem caused by the extremely sparse adjacency matrix (more 0s and fewer 1s).

λ is the attribute reconstruction loss parameter.

By using the Adam optimizer to minimize this loss function, GCN network can adaptively learn the optimal weight parameters and iterate until convergence, thereby generating high-quality graph embedding Z, which is used as the input of the clustering algorithm in the next Section.

5. Improved Clustering Algorithm Integrating Graph Embedding

This section proposes an improved fuzzy clustering framework, aiming to solve the problems of initial center sensitivity and noise interference of traditional clustering algorithms in processing power load data. The algorithm takes the high-dimensional embedding vector Z output by GNN as input, adaptively selects initial clustering centers through the Density Peak Clustering (DPC) algorithm, and uses the Fuzzy C-Means (FCM) algorithm for iterative optimization, ultimately obtaining user behavior patterns with soft partitioning characteristics[17].

Although the traditional FCM algorithm can describe the fuzziness of users' power consumption behavior through the membership matrix, it has two main defects: sensitivity to initialization, randomly selecting initial clustering centers is likely to cause the algorithm to fall into local optimal solutions; limitation of Euclidean distance: in high-dimensional space, the simple Euclidean distance is difficult to capture complex data distributions.

To address the above problems, the improvement strategies of this paper are as follows:

Input layer: The graph embedding matrix $Z \in R_{N\times d}$ is used instead of the original load data, and the topological structure information contained in it is used to improve the robustness of clustering.

Initialization layer: The idea of Density Peak Clustering (DPC) is introduced to calculate the local density and delta distance of data points, and determine high-density and mutually separated points as initial centers.

Iteration layer: Perform FCM iterative optimization to output the membership matrix U and final clustering centers V.

5.1.Initial Center Selection Based on Density Peaks

To overcome the sensitivity of FCM to initial values, we first calculate two key indicators of each user node embedding vector z_i: Local Density(ρ_i) and Delta Distance (δ_i)

Calculate local density ρ_i: The Gaussian kernel function is used to calculate the density of data points around node i:

$$\rho_i = \sum_{j\neq i} \exp\left(-\frac{\left\|z_i - z_j\right\|^2}{d_c^2}\right) \tag{15}$$

Where d_c is the Cut-off distance, usually taking the top 1%~2% of all pairwise distances.

Calculate delta distance δ_i: δ_i is defined as the minimum distance from the point to the point with higher local density:

$$\delta_i = \min_{\rho_j > \rho_i}\left(\left\|z_i - z_j\right\|\right) \tag{16}$$

Note: For the point with the maximum density, δ_i takes the maximum distance from it to all points.

Use the decision value to judge initial centers:

$$\gamma_i = \rho_i \times \delta_i \tag{17}$$

Select the top K points with the largest γ as the initial clustering center set $V^{(0)} = \left\{v_1^{(0)}, v_2^{(0)}, \cdots, v_K^{(0)}\right\}$. Figure 4 shows the schematic diagram of the selection of initial cluster centers.

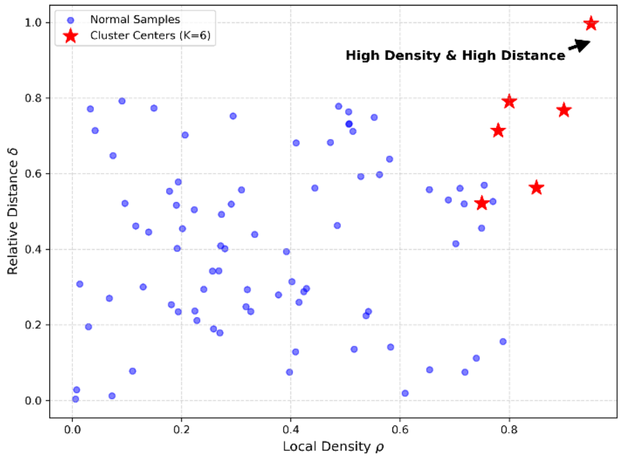

Figure 4. Initial cluster center selection

5.2. Objective Function and Iterative Update of Improved FCM

After determining the high-quality initial centers, enter the fuzzy clustering stage.

Objective Function Construction: Let the data set be $Z = \{z_1, z_2, \cdots, z_N\}$,the clustering centers be $V = \{v_1, v_2, \cdots v_K\}$,and the membership matrix be $U = \left[u_{ij}\right]_{K\times N}$ [18].The objective function J_m is defined as:

$$J_m(U,V) = \sum_{i=1}^{N}\sum_{j=1}^{K}\left(u_{ij}\right)^m \left\|z_i - v_j\right\|^2 \tag{18}$$

Where: m is the fuzzy weight index, usually taking m=2.

Constraint conditions:

$$\sum_{j=1}^{N} u_{ij} = 1 \tag{19}$$

The Lagrange multiplier method is used to solve this problem. It is necessary to construct the Lagrangian function and find the partial derivative to obtain the iterative update formula:

Lagrangian function formula:

$$L = J_m + \sum_i \lambda_i \left(\sum_j u_{ij} - 1\right) \tag{20}$$

Membership update formula:

$$u_{ij} = \frac{1}{\sum_{k=1}^{K}\left(\frac{\left\|z_i - v_j\right\|}{\left\|z_i - v_k\right\|}\right)^{\frac{2}{m-1}}} \tag{21}$$

Physical meaning: The closer user i is to center j, the larger its membership u_{ij} is.

Clustering center update formula:

$$v_j = \frac{\sum_{i=1}^{N} u_{ij}^m z_i}{\sum_{i=1}^{N} u_{ij}^m} \tag{22}$$

Physical meaning: The new clustering center is the weighted average of all data points, and the weight depends on the membership.

The samples are insufficient to preserve the boundary information between classes. At this point, the algorithm degrades into a typical SMOTE data synthesis approach[19-20].

Convergence condition:

$$\left\|V^{(t+1)}-V^{(t)}\right\|<\varepsilon \tag{23}$$

6. Object Behavior Pattern Analysis

After identifying the K typical power consumption behavior patterns of flexible resources in the virtual power plant using graph neural networks and clustering algorithms, this paper further explores the driving factors behind the patterns by introducing the Naive Bayes classifier to construct a probability analysis model. The model aims to quantify the probability of external feature attributes influencing the selection of the resource's power consumption pattern, thereby providing a basis for predicting the day-ahead dispatch of the virtual power plant.

6.1. Bayesian Probability Model Construction

Let $x=\{x_1,x_2,\dots x_m\}$ be the feature attribute vector affecting the power consumption behavior of flexible resources, where x_j represents the j-th feature attribute[21]. According to Bayes' theorem, the posterior probability $P(c_i\mid x)$ that the resource exhibits the i-th power consumption pattern c_i under the known feature vector x can be expressed as:

$$P(c_i\mid x)=\frac{P(x\mid c_i)P(c_i)}{P(x)} \tag{24}$$

Where $P(c_i)$ is the prior probability of pattern c_i that is the frequency of the pattern appearing without considering any environmental factors; $P(x\mid c_i)$ is the conditional probability of the feature attribute x appearing under the condition that pattern c_i occurs.[22]

Assuming that each feature attribute affecting power consumption behavior is independent of the others, the joint conditional probability can be decomposed into the product of the conditional probabilities of each feature attribute.

$$P(x\mid c_i)=\prod_{j=1}^{m}P(x_j\mid c_i) \tag{25}$$

Since the probability $P(x)$ of the feature attribute x appearing in each pattern is not only the same but also a constant for all categories, comparing the size of $P(c_i\mid x)$ is equivalent to comparing the size of the numerator $P(x\mid c_i)$ $P(c_i)$. The most likely power consumption pattern c of the flexible resource in the virtual power plant under a specific environmental feature x is:

$$c=\arg\max_{c_i\in C}P(c_i)P(x\mid c_i) \tag{26}$$

6.2. Probability Calculation Method of Feature Attributes

For discrete features such as working days and holidays, the frequency estimation probability method is adopted. Let the total number of samples belonging to pattern c_i in the training set be N_{c_i} ,and the number of samples where the feature x_j takes the value a_{jk} be $N_{c_i,x_j=a_{jk}}$,then the conditional probability estimation is:

$$P(x_j=a_{jk}\mid c_i)=\frac{N_{c_i,x_j=a_{jk}}}{N_{c_i}} \tag{27}$$

For continuous features such as temperature, it is assumed that they follow a Gaussian distribution under each pattern. Assuming that under pattern c_i ,the mean value of the feature attribute x_j is $\mu_{c_i,j}$,and the variance is $\sigma_{c_i,j}^2$, then its probability density function is:

$$P(x_j\mid c_i)=\frac{1}{\sqrt{2\pi}\sigma_{c_i,j}}\exp\left(-\frac{\left(x_j-\mu_{c_i,j}\right)^2}{2\sigma_{c_i,j}^2}\right) \tag{28}$$

Figure 5 shows the schematic diagram of the power consumption behavior pattern mining method integrated with graph embedding.

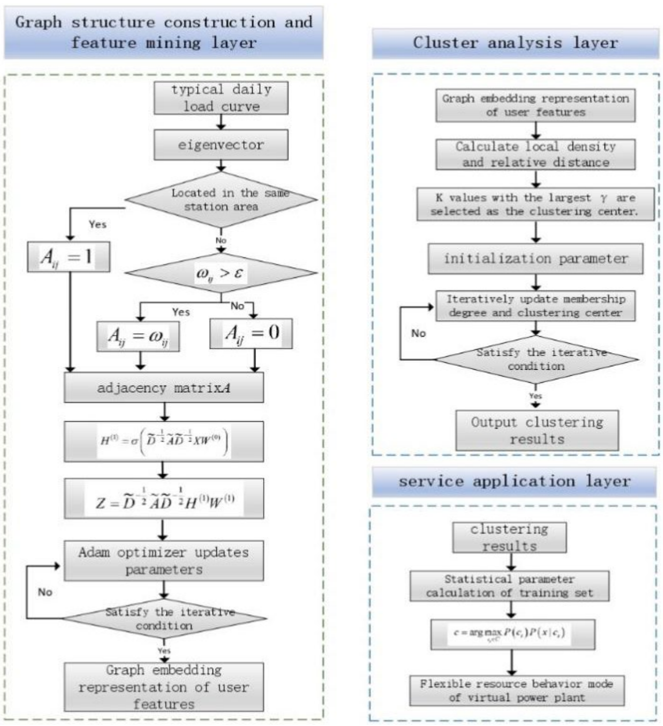

Figure 5. Schematic diagram of the electricity behavior pattern mining method based on graph embedding

7. Case Analysis

7.1. Experimental Data and Parameter Settings

To validate the effectiveness and rationality of the fine-grained power consumption behavior pattern mining of power customers based on graph neural networks in the big data environment, this chapter presents the power user behavior pattern mining method based on decision trees and the machine learning-based method as benchmark approaches, and conducts comparative simulation experiments.

The dataset is derived from actual measurement data from smart meters within the jurisdiction of State Grid Hebei Southern Power Grid, covering 1485 low-voltage residential, commercial, and industrial users in the region. After excluding 3 workdays with abnormal data acquisition and 2 extreme-weather days, the final valid data period is 25 days. The data has a sampling interval of 15 minutes, with 96 load data points collected per user per day, yielding a total of 25×96=2400 valid data points per user. The dataset contains load features, namely the measured active power values for each user at 15-minute intervals, as well as auxiliary features such as the user's affiliated transformer area ID, ambient temperature, relative humidity, and date type. The data have undergone the preprocessing described in Chapter 1, including linear interpolation to fill missing values, same-type-day mean filling, and min-max normalization, resulting in a high-quality load feature matrix.For the graph neural network model, the number of GCN layers is set to 2, the hidden layer dimension is set to 32, the output embedding vector dimension is set to 16, and the learning rate is 0.01. For the clustering d_c takes the top 2% of the total pairwise distances of data points.

7.2. Determination of Optimal Number of Clusters K

The selection of the number of clusters K directly affects the fineness of pattern mining. To avoid the subjectivity of specifying the K value based on experience, in this paper a combination of Sum of Squared Errors (SSE) and Silhouette Coefficient (SC) is adopted for determining the optimal cluster count.

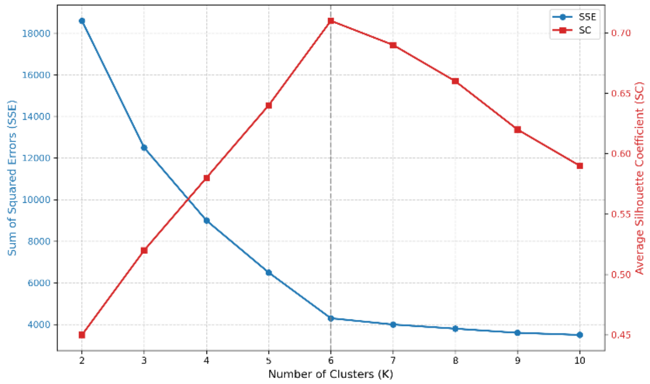

Figure 6. Cluster number selection graph based on SSE and SC

As shown in Figure. 6, with an increase in the number of clusters K, the SSE shows an obvious downward trend. When K increases from 2 to 5, the slope of the SSE curve is large, indicating that increasing the number of clusters can significantly improve the intra-class compactness; when K>6, the decline range of SSE tends to be flat, and the curve shows a typical elbow inflection point. This indicates that K=6 is the balance point between intra-class deviation and model complexity. The value range of the silhouette coefficient is [-1,1], and the closer its absolute value is to 1, the higher the rationality of the category to which the sample belongs. The experimental results show that the SC value reaches the highest point of 0.71 when K=6. When K continues to increase, due to some similar users being forced to be divided into different categories, the inter-class distance decreases, and the SC value begins to decline instead.

7.3. Comparison of Clustering Algorithm Performance

By analyzing the intra-class similarity and the independence of different classes, the effectiveness index can assess the quality of clustering results. The intra-class similarity can be measured by the average cluster center distance, which is the average distance from each sample to its corresponding cluster centroid. The independence of different classes can be measured by the average inter-class distance. Then, the effectiveness index is defined as the ratio of the inter-class independence to the average cluster center distance. According to the definition, the larger the value of the effectiveness index E, the more independent the classes and the more compact the intra-class. Meanwhile, clustering accuracy (ACC), Silhouette Coefficient (SC), and Davies–Bouldin Index (DBI) are introduced to quantitatively evaluate the clustering performance.

The clustering results of the decision tree-based power user behavior pattern mining method and the machine learning-based power user behavior pattern mining method are denoted as Method 1 and Method 2, respectively. The number of clusters is set to 6 for all algorithms, and the corresponding results are presented in Table 1.

The proposed algorithm has the largest average inter-class distance and the smallest average cluster center distance, so the clustering results have a better geometric structure, indicating better clustering results.The final clustering results are shown in Figure 7.

Table.1 Comparison of Clustering Performance

Method	Validity	SC	DBI	ACC	Time(s)
K-means	3.565	0.58	1.92	0.61	1.12
FCM	3.822	0.63	1.56	0.66	1.85
Method I	4.291	0.67	1.32	0.73	2.43
Method II	5.602	0.68	1.28	0.71	2.76
Proposed Method	7.292	0.71	0.94	0.79	2.52

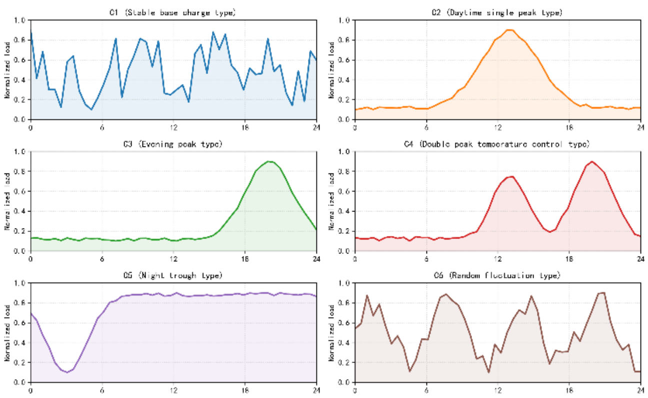

Figure 7. The final clustering effect

7.4. Scalability Validation Experiment

To further verify the scalability of the proposed method, supplementary experiments are conducted on datasets with increasing node scales. Specifically, five datasets with node quantities of N=500, 1000, 2000, 5000, and 10000 are constructed by sampling from the original dataset. The statistical distribution characteristics of load features and graph structures are strictly retained during the sampling process. For each dataset, the following indicators are evaluated: the single-epoch training time of the GCN autoencoder, the clustering time of the DPC-FCM algorithm, and the Silhouette Coefficient.

Table 2. Training duration under different numbers of nodes

Number of nodes	GCN single-round training time (s)	Clustering time (s)	Total time (s)	SC
500	0.42	0.18	0.60	0.72
1000	0.85	0.36	1.21	0.71
2000	1.73	0.79	2.52	0.71
5000	4.38	2.15	6.53	0.70
10000	8.92	4.87	13.79	0.69

As shown in Table 2, the training time of the GCN autoencoder increases approximately linearly with the number of nodes. When the dataset scale expands from 500 to 10000 (a 20-fold increase), the training time rises from 0.42 s to 8.92 s (approximately 21 times). The clustering time also presents a near-linear growth trend. In terms of clustering performance, the silhouette coefficient remains stable as the dataset size increases, demonstrating that the proposed method can maintain robust clustering results even in large-scale scenarios.

In conclusion, the proposed method exhibits excellent scalability with nearly linearly rising computational consumption and stable clustering performance. It is well applicable to large-scale data analysis scenarios for smart grids.

7.5. The Impact of Different Graph Structure Models on Clustering Results

Conduct graph structure ablation experiments to quantitatively analyze the influence of different graph structures on model performance. The graph constructed solely by physical connections, the graph built only based on the Pearson correlation coefficient, and the proposed method are compared to analyze the effectiveness of graph structures.

Table.3 Influence of different graph structures

Method	DBI	Validity	SC
Method I	1.55	5.821	0.62
Method II	1.24	6.433	0.66
Proposed Method	0.94	7.292	0.71

As shown in Table 3, the fused graph structure constructed in this paper outperforms other graph construction strategies across all evaluation indicators. Specifically, it achieves the highest Silhouette Coefficient, the lowest Davies–Bouldin Index, and the maximum validity index E, which demonstrates superior clustering compactness and category separability.

7.6 Analysis of the FCM Fuzzy Weight Exponent *m* Parameter

To investigate the influence of the fuzzy weighting index *m* on clustering performance, five values m=1.5,2,2.5,3,3.5 are tested.

Table.4 The impact of m on clustering performance

m	DBI	Validity	SC
1.5	1.23	6.381	0.66
2.0	0.94	7.292	0.71
2.5	1.15	6.957	0.69
3.0	1.38	6.538	0.67
3.5	1.46	6.119	0.64

As shown in Table 4, when *m* is small (e.g., 1.5), the clustering tends to behave like hard clustering, reducing the flexibility of membership representation and making the model sensitive to noise. When *m*=2, the model achieves the best balance between fuzziness and discrimination, resulting in optimal clustering performance across all metrics. As m increases, the membership distribution becomes overly smooth, leading to blurred cluster boundaries and reduced discriminative power. The fuzzy weighting index m controls the degree of fuzziness in cluster assignments. A moderate value allows the model to capture uncertainty while preserving cluster separability, which is crucial for complex power consumption patterns.Therefore, *m*=2 is selected as the optimal parameter in this study, ensuring both stability and clustering accuracy.

7.7. Behavior Pattern Analysis

Based on the identified 6 typical power consumption behavior patterns, the patterns can be defined as follows: C1 (Stable Base Load Type), C2 (Daytime Single-Peak Type), C3 (Evening Peak Type), C4 (Double-Peak Temperature Control Type), C5 (Night Low Valley Type), C6 (Random Fluctuation Type).

The behavior patterns of resources are greatly affected by climate conditions and date types. For these two influencing factors, this paper analyzes the typical behavior patterns, as shown in Fig. 8 and Fig. 9. For the behavior patterns of resources, the load size of resources can be estimated according to climate conditions and working day changes, which fully verifies the effectiveness of the proposed method in analyzing users' power consumption patterns and provides a reasonable and effective means for the regulation of virtual power plants.

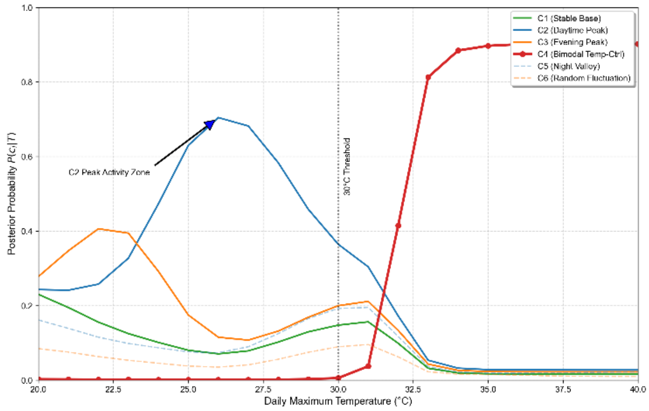

Figure 8. The posterior probability of the typical power consumption mode varies with climatic conditions

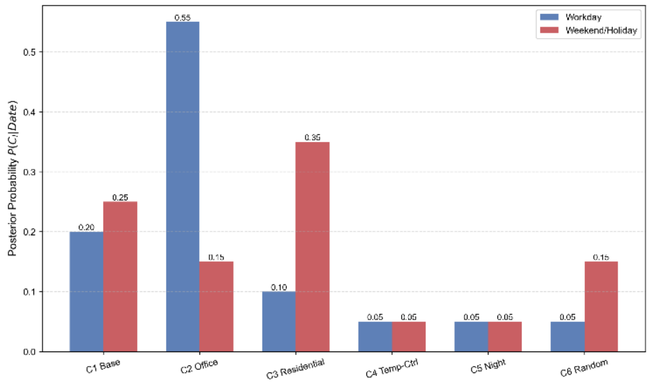

Figure 9 The posterior probability of the typical power consumption mode varies with date type

Conclusion

In this paper, GCN is used to realize the deep fusion and dimension reduction of space-time features of power load data. Experiments show that the algorithm in this paper effectively overcomes the sensitivity of traditional clustering to initial values, and the Validity Function Value reaches 7.292. It is obviously better than the comparison algorithm in terms of intra-class aggregation and inter-class identification, and realizes the accurate portrait of the resource subject of the virtual power plant.

However, the deficiency of this paper is that the computational complexity of the construction and training process of the graph structure is high, and the real-time performance in the large-scale node scenario needs to be optimized. At the same time, the model 's consideration of dynamic random factors such as electricity market prices and policy interventions is not sufficient.

In the future, the lightweight graph neural network architecture will be explored to improve the processing performance, and the multi-task learning framework will be introduced to deeply couple the behavior pattern mining with real-time load forecasting and price elasticity analysis, so as to provide more complete decision support for the intelligent scheduling of virtual power plants.

Acknowledgments

This work was supported by the project "Refined Potential Mining and Evaluation Strategy for Adjustable Resources of Virtual Power Plants under the Market Mechanism in Hebei Province" (Grant No. B704JY230086).

References

[1] KONG X Y, ZHENG F, E Z J, et al. Short-term load forecasting based on deep belief network[J]. Automation of Electric Power Systems, 2018, 42(5): 133-139.

[2] Tambunan H B, Barus D H, Hartono J, et al. Electrical peak load clustering analysis using K-means algorithm and silhouette coefficient[C]//2020 International Conference on Technology and Policy in Energy and Electric Power (ICT-PEP). IEEE, 2020: 258-262.

[3] Rajabi A, Eskandari M, Ghadi M J, et al. A comparative study of clustering techniques for electrical load pattern segmentation[J]. Renewable and Sustainable Energy Reviews, 2020, 120: 109628.

[4] LIU J, LUO F, LIU R J, et al. Study on application strategies of demand-side management under big data background[J]. Power Demand Side Management, 2016, 18(2): 5-10.

[5] SUN M, KONSTANTELOS I, STRBAC G. C-vine copula mixture model for clustering of residential electrical load pattern data[J]. IEEE Transactions on Power Systems, 2017, 32(3): 2382-2393.

[6] AI X, YANG Z H, HU H Y, et al. A load curve clustering method based on improved K-means algorithm for virtual power plant and its application[J]. Electric Power Construction, 2020, 41(5): 28-36.

[7] ZHAO T H, ZHANG Y, WANG J X. Identification method of load outlier based on density-based spatial clustering and outlier boundaries[J]. Automation of Electric Power Systems, 2021, 45(10): 97-105.

[8] CAI X W, YANG J S. Residents mid-term electricity demand forecasting model based on fuzzy C-means clustering[J]. Power Demand Side Management, 2016, 18(3): 23-26.

[9] Pérez-Ortega J, Roblero-Aguilar S S, Almanza-Ortega N N, et al. Hybrid fuzzy C-means clustering algorithm oriented to big data realms[J]. Axioms, 2022, 11(8): 377.

[10] XU W Y. Research and analysis of customer demand response based on artificial intelligence[J]. Power Demand Side Management, 2019, 21(3): 17-20, 31.

[11] WANG X, LIU Z, ZHANG H, et al. A review on virtual power plant concept, application and challenges[C]//2019 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia). 2019: 4328-4333.

[12] LI Z Y, WU J Y, WU W L, et al. Power customers load profile clustering using the SOM neural network[J]. Automation of Electric Power Systems, 2008(15): 66-70, 78.

[13] NAUGHTON J, WANG H, CANTONI M, et al. Co-optimizing virtual power plant services under uncertainty: A robust scheduling and receding horizon dispatch approach[J]. IEEE Transactions on Power Systems, 2021, 36(5): 3960-3972.

[14] E. Mokaramian, H. Shayeghi, F. Sedaghati, A. Safari, H.H. Alhelou, An optimal energy hub management integrating EVs and RBS based on three-stage model considering various uncertainties, IEEE Access 10 (2022) 17349–17365.

[15] Wu Z, Lin X, Lin Z, et al. Interpretable graph convolutional network for multi-view semi-supervised learning[J]. IEEE Transactions on Multimedia, 2023, 25: 8593-8606.

[16] Sierla S, Pourakbari-Kasmaei M, Vyatkin V. A taxonomy of machine learning applications for virtual power plants and home/building energy management systems[J]. Automation in Construction, 2022, 136: 104174.

[17] Ikotun A M, Ezugwu A E, Abualigah L, et al. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data[J]. Information Sciences, 2023, 622: 178-210.

[18] Hashemi S E, Gholian-Jouybari F, Hajiaghaei-Keshteli M. A fuzzy C-means algorithm for optimizing data clustering[J]. Expert Systems with Applications, 2023, 227: 120377.

[19] Zhu J, Ma X, Martínez L, et al. A probabilistic linguistic three-way decision method with regret theory via fuzzy c-means clustering algorithm[J]. IEEE Transactions on Fuzzy Systems, 2023, 31(8): 2821-2835.

[20] Ezugwu A E, Ikotun A M, Oyelade O O, et al. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects[J]. Engineering Applications of Artificial Intelligence, 2022, 110: 104743.

[21] Wang L, Pan Q, Wang S. Data-driven predictions of shield attitudes using Bayesian machine learning[J]. Computers and Geotechnics, 2024, 166: 106002.

[22] Mat S R T, Ab Razak M F, Kahar M N M, et al. A Bayesian probability model for Android malware detection[J]. ICT Express, 2022, 8(3): 424-431.