



OPEN ACCESS

EDITED BY

Haris M. Khalid,
University of Dubai, United Arab Emirates

REVIEWED BY

Fengtao Guang,
China University of Geosciences Wuhan, China
Krishna Kumar Mohbey,
Central University of Rajasthan, India

*CORRESPONDENCE

Hongmei Cheng,
✉ hmcheng@mail.ustc.edu.cn

RECEIVED 17 July 2024

ACCEPTED 21 November 2024

PUBLISHED 23 January 2026

CITATION

Zhang Z, Cheng H, Wang P and Zhang S (2026)
An approach to model and method for current
pattern mining for electricity consumption.
Front. Energy Res. 12:1464389.
doi: 10.3389/fenrg.2024.1464389

COPYRIGHT

© 2026 Zhang, Cheng, Wang and Zhang. This is
an open-access article distributed under the
terms of the [Creative Commons Attribution
License \(CC BY\)](#). The use, distribution or
reproduction in other forums is permitted,
provided the original author(s) and the
copyright owner(s) are credited and that the
original publication in this journal is cited, in
accordance with accepted academic practice.
No use, distribution or reproduction is
permitted which does not comply with these
terms.

An approach to model and method for current pattern mining for electricity consumption

Zhenya Zhang^{1,2}, Hongmei Cheng^{1,3*}, Ping Wang^{1,2} and Shuguang Zhang⁴

¹Anhui Province Key Laboratory of Intelligent Building and Building Energy Saving, Anhui Jianzhu University, Hefei, China, ²School of Electronic and Information Engineering, Anhui Jianzhu University, Hefei, China, ³School of Economics and Management, Anhui Jianzhu University, Hefei, China, ⁴Department of Statistics and Finance, University of Science and Technology of China, Hefei, China

Many studies of intelligent electricity must ascertain the running state of appliances/devices based on the electricity situation of users. In this article, we present the current pattern-based appliance running state identification model (CPARSIM) to identify the running states of appliances based on current online data. Based on frequent online power situation data, current online data sequences can be segmented into some sequence pieces with different lengths. Within the framework of CPARSIM, variable-length pieces of current-state sequences are represented by their univariate regression features, with each sequence piece being treated as one point within the two-dimensional feature space. Considering the relationship between the appliance running state and current-state patterns, the problem of current-state pattern set mining is modeled as a cluster analysis problem within CPARSIM, and the presented approach employs a DBSCAN algorithm-based technique to mine the current-state pattern set. Experimental results show that the DBSCAN algorithm-based approach for the current-state pattern set mining is more effective than the k-means algorithm and the self-organizing map (SOM) neural network.

KEYWORDS

current sequence piece, current state pattern, univariate regression model, PSO algorithm, clustering, identification

1 Introduction

Electricity is one irreplaceable form of energy in modern production and daily life. Research related to its generation, transmission, storage, and utilization is increasingly the focus of researchers and applications on electric grids, power supply, distribution, renewable energy, and energy efficiency management (Hou et al., 2024; El-Sayed et al., 2024; Ahmed et al., 2024; Naidu et al., 2024; Gennitsaris et al., 2023; Hasanvand et al., 2024; Yang et al., 2024; Chen et al., 2023; Kumar et al., 2023a; Kumar, 2024; Kumar et al., 2023b). With the popularization of Internet of Things (IoT) technology, an increasing number of applications and research have focused on precisely monitoring and managing electricity usage processes at the user end (Ahammed et al., 2021; Abhishek et al., 2021; Fabricio et al., 2020; Ramson et al., 2022; Balakumar et al., 2023; Solatidehkordi et al., 2023; Caldera et al., 2023; Çimen et al., 2021; Chen et al., 2020; Avancini et al., 2021; Chou and Truong, 2019;

Diawuo et al., 2020; Saleem et al., 2021; Sayed et al., 2022; Rago et al., 2021; Himeur et al., 2023; Mohi-Ud-Din et al., 2021; Priyadharshini et al., 2021; Ullah et al., 2022; Zhou et al., 2022; Zhu et al., 2020). Currently, most research on electricity consumption on the user demand side is based on features such as voltage, current, power, and other characteristics of appliances. The steady-state analysis, transient analysis, and a combination of both have received attention. Data used in those research works are electricity consumption situation data acquired through smart meters (Avancini et al., 2021; Priyadharshini et al., 2021; Wang et al., 2013; Swindiaro et al., 2018). Because the time granularity of the observed electricity consumption situation data is relatively large when electricity consumption situations are collected through smart meters, it is insufficient to support the timely identification of the types of appliances in use and their running state. The precise perception and online monitoring of electricity consumption situations on the user side not only can help understand details of the user-side power consumption but can also contribute to enhancing the safety and efficient management of user-side energy (Chou and Truong, 2019; Phangbertha et al., 2019; Wang and Zhang, 2022): On the one hand, the online monitoring and precise perception of electricity consumption situations enable users to promptly monitor the operation state of their appliances and detect any abnormal issues, ensuring the safety of their electricity usage; on the other hand, the online monitoring and precise perception of electricity consumption situations on the user side also contribute to a more accurate understanding of the electricity consumption process, providing detailed evidence for optimizing electricity usage behavior and adjusting electricity consumption habits.

There are two methods for running-state monitoring of electrical appliances. One is intrusive appliance load monitoring (ILM), and the other is nonintrusive appliance load monitoring (NALM). The ILM method generally utilizes devices such as smart plugs to collect data on the electric consumption situation of individual appliances (Chou and Truong, 2019; Phangbertha et al., 2019; Wang and Zhang, 2022; Ridi et al., 2013; Yan et al., 2019). The ILM method typically conducts further research based on the current and power data from the electric consumption situation. By analyzing and modeling those data, the ILM method can reveal the electricity consumption patterns of users, electrical load characteristics, and other potential measures for improving energy efficiency (Chou and Truong, 2019; Wang and Zhang, 2022). Based on the typical ILM system architecture and commonly used feature extraction and machine learning algorithms in ILM systems, methods based on the K-nearest neighbor algorithm and the Gaussian mixture model (GMM) are used to classify appliances based on smart plugs (Ridi et al., 2013). This method extracts dynamic features of current data sequences by utilizing the first and second derivatives of the electric data sequence. A household appliance identification algorithm based on the Bayesian classification model is proposed based on the electric consumption situation data collected through smart plugs (Yan et al., 2019), which can summarize and extract the electricity consumption behavior and energy consumption characteristics of typical household appliances from the load power sequence samples.

The NALM method samples electricity consumption data such as voltage, current, load, or power of the total load at the power

entrance. By decomposing the electricity consumption data, the NALM method can infer the operational state of various electrical appliances. In general, NALM utilizes devices such as electric meters for electricity consumption data collection and can be used to monitor and identify individual electrical appliances within households or buildings in time (Hart, 1992; Kong et al., 2018; Himeur et al., 2020; Chen et al., 2022; Aslan and Nur, 2022; Liu et al., 2019). The NALM method can determine the energy consumption of an appliance when the appliance is working within an electrical load by conducting a detailed analysis of the total load current and voltage measured at the power interface (Hart, 1992).

To offer an improved solution for accurately identifying appliances such as washing machines and dishwashers with diverse modes and varying energy consumption patterns, a novel framework based on hierarchical hidden Markov models (HHMM) is given for modeling the problem of appliance identification (Kong et al., 2018). An appliance identification system based on the NALM method utilizes multi-scale wavelet transform to extract features from data sequences and employs an integrated bagging tree classifier for data sequence classification (Himeur et al., 2020), which presents a novel and reliable event detection scheme. Based on the NALM method, a two-stream convolutional neural network (TSCNN) is used to extract temporal and spectral load characteristics from current sequences and perform classification tasks (Chen et al., 2022). This approach effectively captures and uses the time-domain and frequency-domain features of the load signals for accurate classification. The K-nearest neighbors rule (k-NNR) and support vector machines (SVMs) for load identification are employed to extract ten features on the time domain and the steady state from electrical current sequences (Aslan and Nur, 2022). Additionally, principal component analysis (PCA) is combined with k-NNR and SVM separately to evaluate the effectiveness of the ten features and select the most useful ones. Meanwhile, S-transform extracts device features and employs an SVM classifier to identify each electrical device (Liu et al., 2019).

With high-frequency online monitoring of the electricity consumption situation on the user side, this article proposes the current pattern-based appliance running state identification model (CPARSIM) to promptly identify the running state of electrical appliances on the user side. By analyzing the electrical current state, the model can accurately determine the running state of appliances in real time. This approach can enable efficient monitoring and management of user-side electrical appliances. To extract the current patterns corresponding to different running states of electrical appliances from a massive amount of high-frequency monitoring data of user electricity consumption, with definitions of the current transfer state and current steady state, CPARSIM represents each piece of the current sequence using a univariate regression feature. In CPARSIM, each cluster of the entire set of univariate regression features of all current pieces is treated as one current pattern, and the DBSCAN algorithm is used to cluster the univariate regression feature set. In CPARSIM, each current pattern corresponds to a specific running state of an electrical appliance. According to the current state of the univariate regression features of the current piece, CPARSIM asserts the running state of the appliance when the electricity consumption situation is monitored online.

The rest of this article is organized as follows. The univariate regression features of the current sequence piece and its construction methods are presented in Section 2; Section 3 proposes a current-state pattern mining method based on the DBSCAN algorithm, and experimental results and analysis are provided in Section 4. Finally, Section 5 summarizes the research of the entire article. The main contributions of this article are as follows:

- (1) Introducing CPARSIM. We propose the current pattern-based appliance running state identification model (CPARSIM), which enables the identification of running states of electrical appliances using univariate regression features of current pieces and current patterns.
- (2) Representing the current sequence piece using a univariate regression model. We utilize a univariate regression feature representation to capture the feature of each piece of the current sequence, allowing for practical analysis and clustering of the current patterns. Due to the occasional data missing in current sequence pieces, the particle swarm optimization (PSO) algorithm is applied to construct the univariate regression features of the current sequence pieces.
- (3) Current-state pattern mining based on cluster analysis. To identify the running state of an electrical appliance, a method for acquiring current patterns was designed utilizing the DBSCAN algorithm, which achieved current pattern acquisition through cluster analysis.

2 Univariate regression feature of current sequence piece

The running state of one electrical appliance often changes from one state to another when the appliance is working. Usually, when the running state of an appliance is in a specific stable condition, the electric consumption data on the power line remain relatively stable. However, when the running state of the appliance changes, the electric consumption data on the power line also change.

If an appliance is running stably, the running state of the appliance is called the steady state. Correspondingly, the online monitoring data sequence of the steady state of the appliance is referred to as the steady-state sequence. As to two adjacent steady states in time of one appliance, the running states of an appliance generally cannot instantaneously transfer from one steady state to the next. In general, the transition requires some time from the first to the next. The data sequence collected at the time interval from the ending of the first steady state to the beginning of the following steady state is referred to as the transfer-state sequence, and the running state of the appliance at that time interval is the transfer state. From the perspective of electricity consumption, the running of an electrical appliance is a continuous transfer process between a steady state and a transfer state. During a steady state, the electrical current of the appliance remains relatively stable. However, the electrical current undergoes variations during the transfer, including momentary peaks or sustained fluctuations. This transfer process reflects the appliance energy consumption and dynamic characteristics under different running states.

The timely identification of the running state of an electrical appliance based on high-frequency current data is a common challenge in appliance running state identification. The CPARSIM framework is depicted in Figure 1.

The CPARSIM framework depicted in Figure 1 consists of Parts A and B. Part A utilizes high-frequency collected data from historical records to construct a set of current-state patterns. On the other hand, Part B uses the current pattern set built in Part A to perform an online analysis of the current sequence piece, thereby identifying the running state of the appliance. In Figure 1, Module 1 performs a deep analysis of the current sequence, decomposing it into some individual steady-state sequence pieces. Module 2 extracts all current transfer state sequences from the current sequence with the steady-state sequences outputted by Module 1 and the original current sequence as inputs. All current steady-state sequences outputted by Module 1 and current transfer-state sequences outputted by Module 2 are converted into current-state features by Module 3. Module 4 takes the current-state feature set as input and generates the current-state pattern set.

As shown in Figure 1, Module 5 is responsible for storing, querying, and visualizing all current-state patterns. Module 6 generates feature representations for the current sequence pieces. With the construction of the running state identification model based on the current-state pattern set in Model 5, Module 7 identifies the running state of the electrical appliance on the circuit with the output of Model 6 as input.

Changes in the operational state of electrical loads in a circuit can cause rapid alterations in the current sequence state. These rapid changes in the current sequence state can be effectively captured when monitoring user electricity usage patterns at a fine-grained level. Based on this concept, the CPARSIM model first derives feature representations of all current transitional and steady states from the sample dataset. It can then determine the state of the load corresponding to the observed current sequence state based on these feature datasets, thus achieving recognition of the appliance's operational state. This model has a framework structure, and the design and implementation of each module within this structure are topics of interest in related research. Overall, the construction and identification processes of the CPARSIM model are as follows.

- a) As shown in Part A of Figure 1, when constructing the feature set of all current steady states, the CPARSIM model first uses Module 1 to extract all steady-state sequence segments from the entire sequence data. Then, Module 2 extracts all steady-state sequence segments from the complete sequence data based on the segments obtained by Module 1. Next, Module 3 performs a feature representation of the current sequence segments constructed by Modules 1 and 2, while Module 4 conducts an in-depth analysis of the feature representation of all current sequence segments to obtain current patterns based on the feature representation. Finally, Module 5 stores and manages all current patterns and provides access support.
- b) As shown in Part B of Figure 1, Module 6 first performs a feature representation of the observed current sequence segments when identifying the current state online. Then, Module 7 constructs an identification model for current

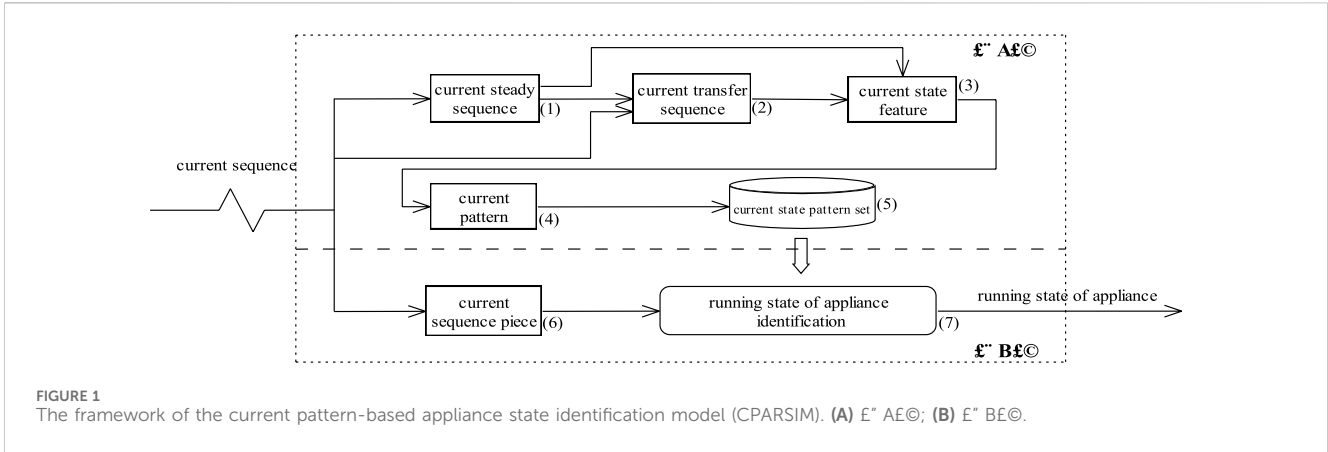


FIGURE 1 The framework of the current pattern-based appliance state identification model (CPARSIM). (A) $E^* AE\odot$; (B) $E^* BE\odot$.

patterns based on the current pattern set stored in Module 5 and uses the constructed identification model to identify the state of the load corresponding to the online current sequence segment features.

2.1 Current steady-state sequence and current transition state sequence

Suppose that the current value in the power line at time interval $(t_{i-1} t_i)$ must be sampled once, where $t_0, t_n \in R, t_0 > 0, t_n > 0, t_0 < t_1 < \dots < t_i < \dots < t_n$, and $T = (t_0 t_n)$ is the time interval for observation. If s_i is the current value in the power line at time interval $(t_{i-1} t_i)$, $S = \langle s_1, s_2, \dots, s_n \rangle$ is the current sequence of electricity consumption. Let $es_k^{(\epsilon)} = \langle s_k, s_{k+1} \dots s_{k+m} \rangle \subseteq S$ be the current sequence of electricity consumption at time interval $\Delta_k^m = (t_k t_{k+m})$ and $\bar{s} = \frac{1}{m+1} \sum_{i=k}^{k+m} s_i$ where $\epsilon > 0$. If inequality system (Equation 1) is satisfied, $es_k^{(\epsilon)}$, the current sequence of electricity consumption at time interval Δ_k^m , is one ϵ -piece of the current sequence of electricity consumption.

The ϵ -piece of the current sequence of electricity consumption defined by Equation 1 below is based on the mean value of the current sequence. In this system, the ϵ -piece of the current sequence is determined based on the mean value of the electricity consumption. It means that the sequence is divided into segments where the variation within each segment is small, and specifically, the maximum deviation from the mean value within a segment does not exceed ϵ . Similarly, Equation 2 illustrates another kind of ϵ -piece of the current sequence of electricity consumption based on jump detection. In this system, jump detection means that if the difference between any two consecutive observed data points is greater than ϵ , those two consecutive observed data points are in different ϵ -pieces of the current sequence. That is, the difference between two successive observed data points is not greater than ϵ in the ϵ -piece of the current sequence of electricity consumption. Furthermore, in the system according to Equation 2, the ϵ -piece is determined based on detecting jumps or significant changes in the consumption data. A jump is defined as a point where the difference between any two consecutive observed data points is greater than ϵ . A jump indicates the end of one ϵ -piece and the beginning of another. It

means that an ϵ -piece is a sequence segment where the difference between any two successive observed data points does not exceed ϵ .

$$\begin{cases} |\bar{s} - s_i| \leq \epsilon \\ k \leq i \leq k + m \\ |\bar{s} - s_{k+m+1}| > \epsilon \\ |\bar{s} - s_{k-1}| > \epsilon \end{cases} \quad (1)$$

$$\begin{cases} |s_i - s_{i+1}| \leq \epsilon \\ k \leq i \leq k + m - 1. \end{cases} \quad (2)$$

Let $es_{i_1}^{(\epsilon)}, es_{i_2}^{(\epsilon)}, \dots, es_{i_u}^{(\epsilon)}$ be all ϵ -pieces of the current sequence of electricity consumption in electricity consumption sequence S , where $i_1 < i_2 < \dots < i_u$, and τs_{i_j} is the time interval of $es_{i_j}^{(\epsilon)}, 1 \leq j \leq u$. If $\tau s_{i_j} \cap \tau s_{i_{j+1}} = \emptyset$ for $1 \leq j \leq u - 1, ES^{(\epsilon)} = \{es_{i_1}^{(\epsilon)}, es_{i_2}^{(\epsilon)}, \dots, es_{i_u}^{(\epsilon)}\}$ is the ϵ -piece set of the current sequence of S .

Suppose $ES^{(\epsilon)} = \{es_{i_1}^{(\epsilon)}, es_{i_2}^{(\epsilon)}, \dots, es_{i_u}^{(\epsilon)}\}$ is the set of ϵ -pieces of the current sequence, and $\tau s_{i_j} = (\tau s_{i_j,begin} \tau s_{i_j,end}]$ is the observation time interval where $i_1 < i_2 < \dots < i_u$. As to $es_{i_1}^{(\epsilon)}$ and $es_{i_2}^{(\epsilon)}$, if $\tau s_{i_1,end} = \tau s_{i_2,begin}$ and $|\bar{es}_{i_1} - \bar{es}_{i_2}| > \Delta > 0$, or $\tau s_{i_1,end} < \tau s_{i_2,begin}, es_{i_1}^{(\epsilon)}$ and $es_{i_2}^{(\epsilon)}$ are current steady-state sequences. If $\tau s_{i_1,end} = \tau s_{i_2,begin}$ and $|\bar{es}_{i_1} - \bar{es}_{i_2}| \leq \Delta > 0, es_{i_1}^{(\epsilon)}$ and $es_{i_2}^{(\epsilon)}$ should be merged as one new current sequence, and the new current sequence is a current steady-state sequence.

Suppose the number of current steady-state sequences in the observation current sequence $S = \langle s_1, s_2, \dots, s_n \rangle$ under $\epsilon > 0, \Delta > 0$ is k . Let es_i be the i th current steady-state sequence, $i = 1 \dots k. ES = \{es_1, es_2, \dots, es_k\}$ is the full current steady-state sequence set of S .

It is easy to prove that the intersection of corresponding time intervals of any two steady-state current sequences is empty.

Suppose $S = \langle s_1, s_2, \dots, s_n \rangle$ is the current sequence observed at period $T = (0 t]$, $ES = \{es_1, es_2, \dots, es_k\}$ is the full current steady-state sequence set of S, es_i is the i th current steady-state sequence, $1 \leq i \leq k, \tau s_i = (\tau s_{i,begin} \tau s_{i,end})$ is the time interval of $es_i, \tau s_{i,begin} \geq 0, \tau s_{i,end} \geq 0$, and $\tau s_{i,begin} < \tau s_{i,end}$. If $\tau s_{0,end} = 0, \tau s_{k+1,begin} = t$ and $move_i$ is the current sequence observed at time interval $(\tau s_{i-1,end} \tau s_{i,begin})$, then $move_i$ is one current transfer-state sequence.

For $es_i \in ES, 1 \leq i \leq k$, if es_i is a current steady-state sequence at time interval $(\tau s_{i,begin} \tau s_{i,end})$, the running state of load at time

interval $(\tau_{s_i,begin} \tau_{s_i,end})$ is also called a steady state. Meanwhile, if es_i is a current transfer-state sequence at time interval $(\tau_{s_i,begin} \tau_{s_i,end})$, the running state of load at time interval $(\tau_{s_i,begin} \tau_{s_i,end})$ is also called a transfer state.

2.2 Mining features of the current sequence piece

Both steady-state current sequences and transfer current sequences are considered current-state sequences. The lengths of each current-state sequence may vary in the current-state sequence. To describe each current-state sequence using fixed features, we use the two constant coefficients of the univariate regression model for the current-state sequence as its characteristics.

If $S = \langle s_1, s_2 \dots s_n \rangle$ is one current-state sequence, and $S' = \langle s'_1, s'_2 \dots s'_n \rangle$ is the estimate of S by the univariate regression model of S , each element of S' can be calculated according to Equation 3. In Equation 3, a and b are the two constant coefficients of the univariate regression model of S . As to one current-state sequence, the optimal values of the coefficients (a , b) of Equation 3 represent the features of the current-state sequence.

$$\begin{cases} s'_{k+1} = s'_k + a, k > 1, a \in R \\ s'_1 = b, k = 1, b \in R \end{cases} \quad (3)$$

As to the current-state sequence $S = \langle s_1, s_2 \dots s_n \rangle$, if $\delta(i)$ is defined as Equation 4 and $E(S, S')$ is defined as Equation 5, the optimal solution to the optimization problem defined by Equation 6 is the univariate regression feature of S .

$$\delta(i) = \begin{cases} 1, s_i \in S, s_i \text{ is not missing} \\ 0, s_i \in S, s_i \text{ is missing} \end{cases} \quad (4)$$

$$E(S, S') = \frac{1}{2 \sum_{i=1}^n \delta(i)} \sum_{i=1}^n \delta(i) (s_i - s'_i)^2 \quad (5)$$

$$E(S, S^*) = \min E(S, S') \quad (6)$$

As to the current-state sequence $S = \langle s_1, s_2 \dots s_n \rangle$, if S has no missing data, that is, if the value of $\delta(i)$ for each element in S is 1, slope a , and intercept b , the univariate regression feature of S can be calculated using the least squares method. When S is missing data, the PSO algorithm is utilized to solve the optimization problem defined by Equation 6 to get the univariate regression feature of S . The optimal solution of the optimization problem defined by Equation 6 is treated as the univariate regression feature of S .

$$v(t+1) = \omega \times v(t) + c_1 \times rand() \times (pBest - x(t)) + c_2 \times rand() \times (gBest - x(t)) \quad (7)$$

$$x(t+1) = x(t) + v(t+1) \quad (8)$$

As to the PSO algorithm, let the velocity of a particle at time t be $v(t)$ and the position of the particle at time t be $x(t)$, $v(t+1)$. The velocity of each particle is updated according to Equation 7. Meanwhile, $x(t+1)$, the position of any particle, is generally updated according to Equation 8. In Equation 7, parameter ω is the inertia factor, c_1 and c_2 are the learning factors, $rand()$ is a

random number, $pBest$ denotes the historical best position of the particle, and $gBest$ is the historical best position of all particles. Because the PSO algorithm is a stochastic optimization algorithm, the solution of PSO for an optimization problem is the approximation of the optimal solution. If an optimization problem is solved using the PSO algorithm, some approximated optimal solutions as candidate solutions may usually be obtained by the PSO algorithm. The solution for the optimization problem is the best solution among those candidate solutions. This approach will significantly increase the computation time required to solve the optimization problem.

3 Methods for current-state pattern mining

3.1 Current-state pattern mining based on DBSCAN algorithm

Input: $FS = \{fs_1, fs_2, \dots, fs_n\}$, univariate regression feature set of current sequence piece; ϵ , neighborhood radius; min_pts , minimum number of samples

Output: PS , current-state pattern set

- 1) $n = \text{length}(FS)$;
- 2) $\text{cluster_labels}(1:n) = 0$;
- 3) $\text{cluster_id} = 1$;
- 4) for $i = 1:n$
- 5) if $\text{cluster_labels}(i) == 0$
- 6) $\text{neighbors} = \text{findNeighbors}(i, FS, \epsilon)$; //Get neighboring data for the i th data point.
- 7) if $\text{numel}(\text{neighbors}) \geq \text{min_pts}$; //The $\text{numel}()$ function counts the total number of neighboring data for the i th data point.
- 8) $\text{cluster_labels}(i) = \text{cluster_id}$
- 9) $\text{cluster_labels} = \text{expandCluster}(i, \text{neighbors}, FS, \epsilon, \text{min_pts}, \text{cluster_id}, \text{cluster_labels})$;
- 10) $\text{cluster_id} = \text{cluster_id} + 1$;
- 11) $PS = \text{cluster_labels}$;

Algorithm 1. current-state pattern mining based on the DBSCAN algorithm.

Although the lengths of each current sequence piece vary in the set of current sequence pieces, each current sequence piece can be represented by a two-dimensional vector if each current sequence piece is represented with the two constant coefficients of its univariate linear regression model. If each current sequence piece is represented with the two constant coefficients of its univariate linear regression model, two analogous current sequence pieces are closely related, and the dissimilarity of two disanalogous current sequence pieces is significant in the two-dimensional feature space for the set of current sequence pieces. The evidence above suggests that the set of current sequence pieces can be clustered into different clusters by a specific clustering analysis algorithm if each current sequence piece is represented with the two constant coefficients of its univariate linear regression model as its

feature. The current-state sequence pieces within each cluster have similar morphologies, while there are significant disparities in the morphology of current-state sequence pieces between different clusters. Given that the morphology of current sequence pieces within each cluster in their respective feature space is similar, the feature representation of a cluster is a current-state pattern. The flow for current-state pattern mining based on the DBSCAN algorithm is given at [Algorithm 1](#). Function *expandCluster()* is used to expand the clustering cluster at step (9) in [Algorithm 1](#). With the use of *expandCluster()*, the neighboring data of the *i*th data point is within the same cluster as the *i*th data point.

Input: *FS*, the feature set of the current sequence piece;
index, the position of the core entity within *FS*;
epsilon, neighborhood radius;
Output: *neighbors*, the indices set of neighboring data for the *i*th core data in *FS*

- 1) *neighbors* = [];
- 2) *distances* = sqrt(sum(*FS* - *FS*(*index*, :).², 2)); // Compute the distance between the specific data point and the other data points.
- 3) for *i* = 1:*numel*(*distances*)
- 4) if *distances*(*i*) ≤ *epsilon*
- 5) *neighbors* = [*neighbors* *i*];

Algorithm 2. *findNeighbors()*, Seeking the neighbors of the core data.

The flow for current-state pattern mining, as presented in [Algorithm 1](#), is based on the fundamental principle of the DBSCAN algorithm. However, it does not label any current-state sequence piece as noise. The key idea for current-state pattern mining based on the DBSCAN algorithm in [Algorithm 1](#) is that any current-state sequence piece with several neighboring sequence pieces not less than *min_pts* is assigned to a cluster, and all neighboring sequences are also within the same cluster. At step(2) of [Algorithm 1](#), each current-state sequence piece is marked with a cluster label of 0, indicating that its cluster membership is undecided. Based on the concept that any current-state sequence piece is the neighbor of the *i*th current-state sequence piece if the distance from the *i*th current-state sequence piece to the current-state sequence piece is not greater than *epsilon*, step(6) of [Algorithm 1](#) uses the function *findNeighbors()* to obtain all neighboring current-state sequence pieces of the *i*th current sequence piece: any current-state sequence piece within a distance *epsilon* from *i*th current sequence piece is considered a neighbor. In step(9) of [Algorithm 1](#), the *expandCluster()* function is used to expand the cluster of the *i*th current sequence piece by performing a depth-first search on its neighboring sequence pieces. [Algorithm 2](#) provides one implementation flow for the *findNeighbors()* function. Meanwhile, [Algorithm 3](#) provides one implementation flow for the *expandCluster()* function. In [Algorithm 2](#) and [Algorithm 3](#), the function *numel()* is used to obtain the number of elements in the current sequence piece set. In [Algorithm 2](#), steps(3)~(5) seek data that are in the neighboring field of data point *FS(index)* where the distance from data point *FS(index)* to the data point is not greater than *epsilon*.

Input: *S*, data sequence; *index*, the indices of core data in *S*; *neighbors*, data buffer for the neighbor of data point *S(index)*; *epsilon*, neighborhood radius; *min_pts*, minimum number of samples; *cluster_id*, the identifier of the cluster where *S(index)* is located; *cluster_labels*, the identifier buffer of each element in *S*, which is the cluster partition of *S*;
Output: *cluster_labels*, Cluster labels after expansion

- 1) for *i* = 1:*numel*(*neighbors*) // traversal each neighboring data point
- 2) *neighbor* = *neighbors*(*i*);
- 3) if *cluster_labels*(*neighbor*) == 0 // no cluster id is assigned for the neighboring data point
- 4) *cluster_labels*(*neighbor*) = *cluster_id*; // Assign cluster id to neighbor
- 5) *neighbor_neighbors* = *findNeighbors*(*neighbor*, *S*, *epsilon*); // Seeking all neighbors of neighbor.
- 6) if *numel*(*neighbor_neighbors*) ≥ *min_pts* *cluster_labels* = *expandCluster*(*neighbor*, *neighbor_neighbors*, *FS*, *epsilon*, *min_pts*, *cluster_id*, *cluster_labels*);

Algorithm 3. *expandCluster()*, Extend the cluster ID of the specified data point to its neighboring data points.

The *findNeighbors()* function, as described in [Algorithm 2](#), computes the distance between the current data point and all other data points in the feature set *FS*. This operation takes $O(n)$ time, where *n* is the number of data points in *FS*. The *expandCluster()* function, as described in [Algorithm 3](#), performs a depth-first search (DFS) to expand the cluster for the neighboring points. In the worst case, it may visit each data point once, leading to a time complexity of $O(n)$ for each call. In [Algorithm 1](#), the main loop iterates over each data point in *FS*, and for each point, it may call *findNeighbors()* and *expandCluster()*, both of which have time complexities of $O(n)$. Therefore, the overall time complexity of [Algorithm 1](#) is $O(n) \times O(n) = O(n^2)$.

3.2 Evaluation of the current-state pattern mining method

When the clustering analysis method extracts current sequence patterns from a collection of current sequence pieces, each current sequence pattern responds to a cluster. For each current-state pattern, every current sequence piece within its corresponding cluster exhibits approximate similarity to the others in the feature space. Furthermore, the sequence pieces corresponding to different current-state modes exhibit dissimilarities with distant distances in the feature space. Those facts, as above, imply that it is possible to partition a set of current sequence pieces into distinct clusters using clustering techniques. Current sequence pieces in the same cluster exhibit similar morphology in the feature space. Meanwhile, the morphology for different current sequence pieces in different clusters is notably disparate. Various metrics, such as SSE (sum of squares for error), silhouette coefficient, and the Calinski–Harabasz index, are commonly used to evaluate the performance of the current-state pattern mining method.

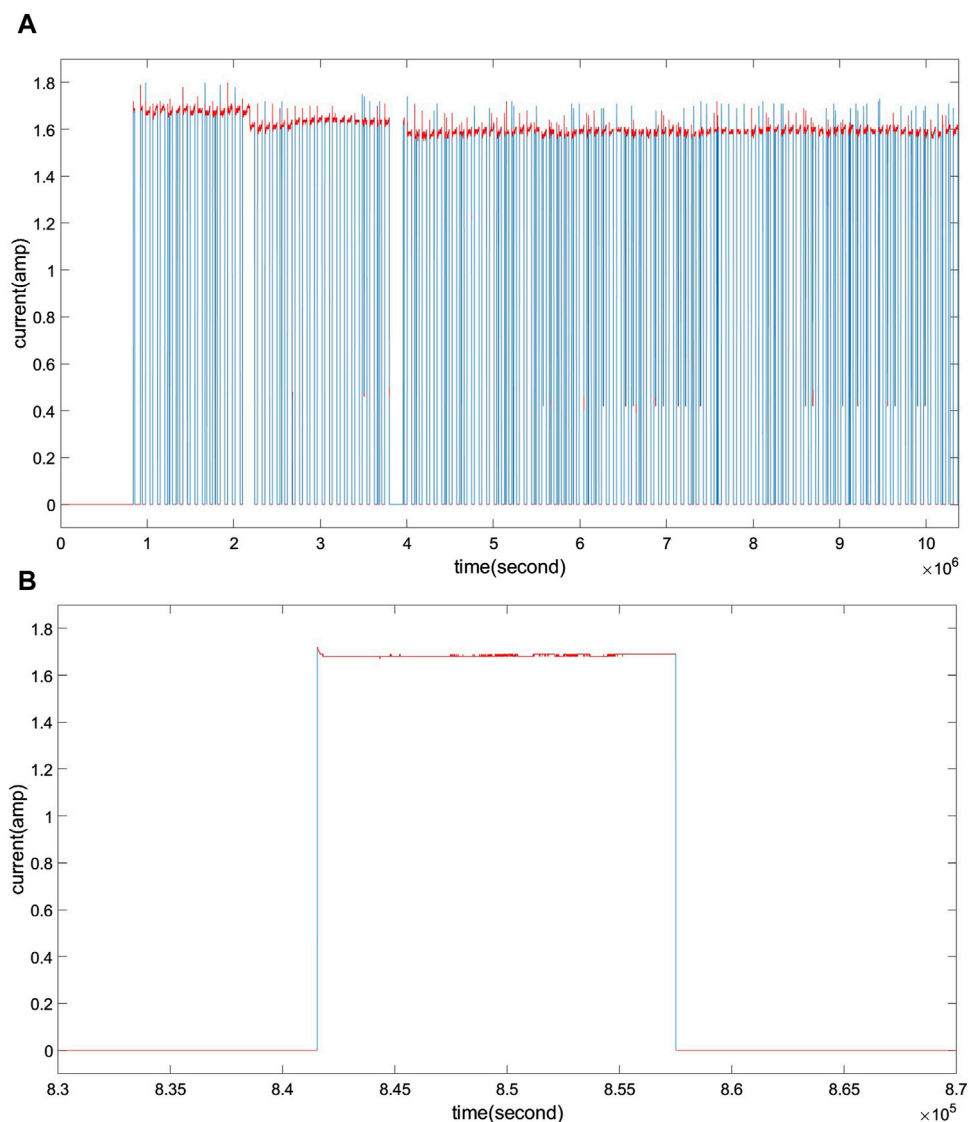


FIGURE 2 Current steady-state and transfer-state sequence pieces. (A) All and (B) Part.

$$precision = 1 - \frac{\|abs(M_{F_1} - M_F)\|}{n \times n}, \tag{9}$$

In the process of clustering the set of current sequence pieces V , where each element in V is the univariate regression features of a current sequence piece, the current sequence pieces within each cluster are similar. At the same time, there are significant variations in the morphology among current sequence pieces across different clusters. Suppose the similarity of feature morphology between any two data points in V is established. In that case, the performance of clustering-based methods for current-state pattern mining can be evaluated by the accuracy of cluster partitioning.

Given a dataset V , let $n = \|V\|$, and suppose F is a clustering partition of V . F can be represented by an $n \times n$ binary matrix M_F . For any arbitrary u and v belonging to V , if F asserts that u and v

are in the same cluster, $M_F(u, v) = 1$; otherwise, $M_F(u, v) = 0$. Specially, if $u = v$, $M_F(u, v) = 1$. With the supposition above, M_F is a representation matrix of the clustering partition. Furthermore, let C be an annotated clustering partition of V , and let F be any clustering partition of V . Then, M_C is the representation matrix of C , and M_F is the representation matrix of F . For $\forall u, v \in V$, if $M_F(u, v) - M_C(u, v) = 0$, the clustering assertion of F for $u, v \in V$ holds. Accordingly, the precision indicator defined in Equation 9 represents the accuracy of cluster partition F .

In Equation 9, the function $abs()$ is used to calculate the absolute value of each component, and the operator $\|\bullet\|$ is used to sum up the matrix components. The value of precision ranges between 0 and 1. As to the clustering partition F , the assertion of whether any two data points in V belong to the same cluster in F is closer to the view of clustering partition C if the value of precision is near 1. The precision indicator defined in Equation 9 can evaluate the

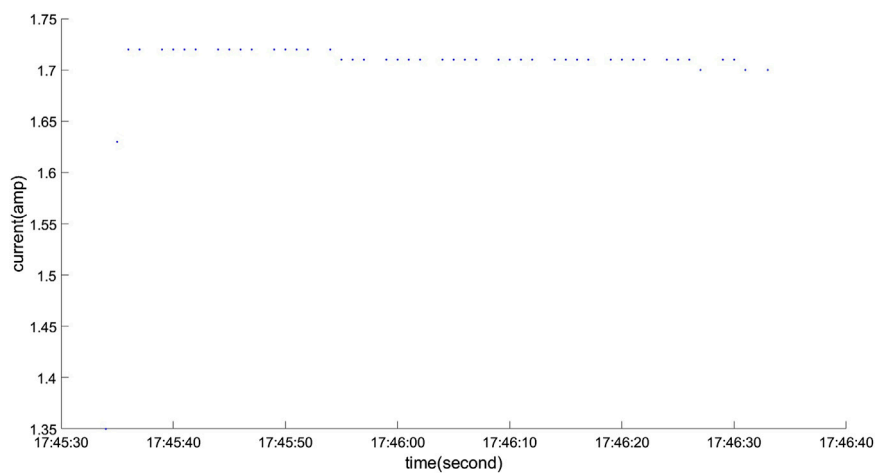


FIGURE 3
The missing data phenomenon in the current sequence.

performance of clustering-based approaches for current-state pattern mining.

It is worth noting that the choice of evaluation methods for clustering partition needs to be determined based on specific problem scenarios and data characteristics in practical applications. Moreover, evaluation metrics are often referenced and should be comprehensively considered with domain knowledge and applicable application requirements.

4 Results and discussion

The electrical data sequence used in our experiment is from the observation data of the lighting line in Anhui Province Key Laboratory of Intelligent Building & Building Energy Saving. The period for the current observation data collected spans from 2023-02-01 00:00:00 to 2023-05-31 23:59:59. During those 10,368,000 s, data were collected 8,098,039 times, and duplicate data collections (where two or more collections occurred in one second) occurred 1,024 times. Moreover, data were missed 2,062,229 times, and power outages occurred 138 times, while the data collection program was terminated once at 206,570 s.

With $\varepsilon = 0.08$, $\Delta = 0.3$, for the current sequence as above, there are 339 current sequence pieces corresponding to the transfer state and 348 current sequence pieces corresponding to the steady state. That is, there are 687 current sequence pieces in our experiment. Figure 2A illustrates the time-domain morphology of the current sequence and the 687 current sequence pieces. Figure 2B presents the time-domain morphology of the current sequence from 2023-02-10 14:33:19 to 2023-02-11 01:39:59 within five current sequence pieces (three steady-state sequence pieces and two transfer sequence pieces). In Figures 2A, B, the curve for the current sequence piece of the steady state is marked in red, while blue marks the curve for the current sequence piece of the transfer state. According to curves in Figures 2A, B, each piece of the current transfer state corresponds to the current sequence piece with relatively rapid ascent or descent. In contrast, the current steady-state pieces fit into the current sequence piece with minimal fluctuations. In Figures 2A, B, the missing

current data within each non-power outage time interval are estimated using the observed value or estimated value (when data are missing) of the previous second's current.

Figure 3 illustrates the time-domain morphology of the current sequence in 1 min intervals from 2023-02-10 17:45:34 to 2023-02-10 17:46:33. In Figure 3, the horizontal axis represents the sampling time, and the vertical axis represents the collected current values. A point in Figure 3 represents a current value collected at a specific time. Theoretically, the points in Figure 3 should have equal horizontal intervals due to the electricity state being sampled per second. However, it can be observed from Figure 3 that the points are not evenly spaced horizontally. This phenomenon indicates that not every moment of current data is collected and implies that the model and method used to construct the feature of the current sequence must be able to handle or tolerate missing data.

The PSO algorithm is used to construct the univariate regression feature of the current sequence piece defined by Equation 3. In our experiment, the particle number of the PSO algorithm is 400, the values of learning factors c_1 and c_2 are 2, and the value of inertia factor ω is 0.8. Because each particle represents a candidate solution to the corresponding optimization problem in the PSO algorithm for solving optimization problems, Table 1 presents the minimum, maximum, mean, median, Q1, Q3, and variance of the feasible solution fitness values corresponding to all 400 particles when the PSO algorithm is used to construct all univariate regression features for 687 current sequence pieces. As the experiment was repeated 20 times, Table 1 contains 20 rows. Across all 20 experiments, the fitness value of the optimal solution had a minimum of $8.11\text{E}-20$, a maximum of $1.71\text{E}-15$, and a variance of $3.9474-16$. These values are all very close to 0. Because the PSO algorithm uses the feasible solution with the smallest fitness value as the estimated optimal solution, and each data point in the minimum column of Table 1 is approximately 0, it can be asserted that the PSO algorithm is suitable for accurately solving the univariate regression features of the current sequence piece. The time used by the PSO algorithm is also given in Table 1. When the PSO algorithm is used to solve the univariate regression features of all 687 current sequence pieces, the minimum time used

TABLE 1 Performance of the PSO algorithm (iterations: 200).

	Minimum	Q1	Median	Q3	Maximum	Mean	Variance	Time (s)
1	2.18E-17	6.14E-06	0.005915	0.046876	0.284161	0.026936	0.035172	3,916.186
2	8.11E-20	6.97E-06	0.004900	0.046876	0.812825	0.028135	0.047289	4,051.608
3	1.72E-17	7.55E-06	0.008788	0.04967	0.556733	0.028838	0.043793	4,037.236
4	1.80E-16	7.20E-06	0.004900	0.04655	0.390373	0.026696	0.035665	3,860.215
5	7.05E-17	5.89E-06	0.008783	0.049599	0.350894	0.028190	0.039172	3,952.556
6	6.24E-16	7.47E-06	0.006978	0.049599	0.367692	0.028093	0.038814	4,055.331
7	4.02E-18	7.69E-06	0.006400	0.049599	0.962228	0.02947	0.053411	4,002.863
8	3.80E-16	7.45E-06	0.007446	0.050069	0.442084	0.028848	0.040611	3,844.893
9	5.13E-17	6.95E-06	0.004225	0.046102	0.394482	0.02644	0.036134	4,071.216
10	1.29E-17	7.25E-06	0.005915	0.047078	0.713795	0.028431	0.047236	3,748.056
11	5.40E-16	5.57E-06	0.002363	0.046008	0.231456	0.026177	0.03434	4,210.883
12	1.83E-16	7.85E-06	0.006400	0.049358	0.322102	0.027581	0.036152	4,028.632
13	1.71E-15	6.37E-06	0.006247	0.047078	0.607395	0.028316	0.044176	3,959.003
14	1.43E-16	7.29E-06	0.006061	0.046755	0.304938	0.027006	0.035239	4,040.110
15	5.28E-17	7.51E-06	0.004900	0.046550	0.231841	0.026547	0.034294	4,308.729
16	1.27E-17	7.29E-06	0.004900	0.046876	0.545821	0.027147	0.038771	4,524.026
17	1.12E-16	7.80E-06	0.004900	0.046102	0.261583	0.026215	0.033314	4,332.294
18	4.52E-17	5.10E-06	0.003803	0.046755	3.118498	0.031507	0.123786	4,312.621
19	4.35E-17	5.21E-06	0.004900	0.046102	0.232323	0.026108	0.033349	4,566.583
20	5.26E-17	4.99E-06	0.004900	0.046755	0.908978	0.028676	0.051158	4,548.110

TABLE 2 Electricity usage state of appliances.

State	Code	Count
Fully closed	1	148
First group open and second group closed	2	30
First group closed and second group open	3	1
Fully open	4	169
From first group open and second group closed to fully closed	5	12
From first group open and second group closed to fully open	6	18
From fully closed to first group open and second group closed	7	1
From fully closed to fully open	8	142
From first group closed and second group open to fully closed	9	1
From fully open to fully closed	10	132
From fully open to first group open and second group closed	11	29
From fully open to first group closed and second group open	12	1
From fully open to fully open via instantaneously fully closed	13	3

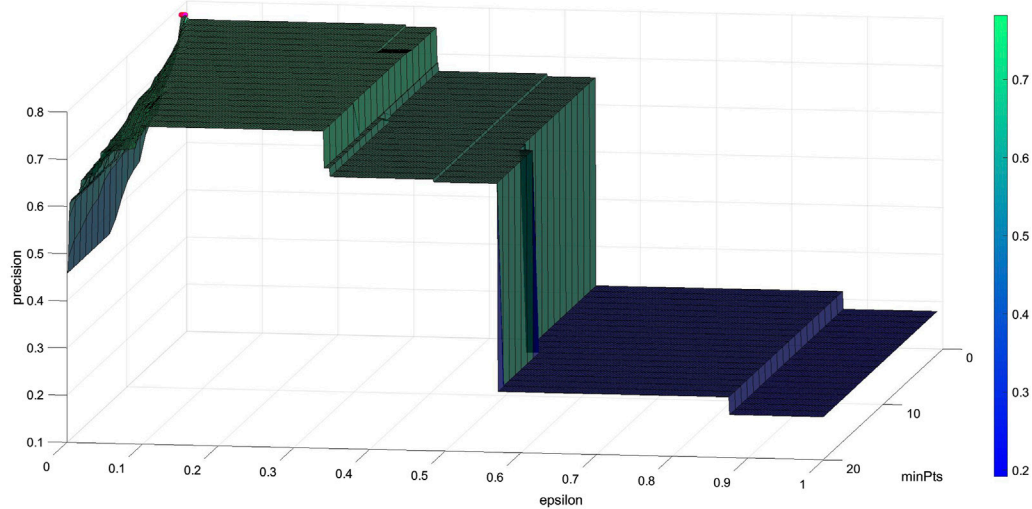


FIGURE 4
The precision of current-state pattern mining by Algorithm 1.

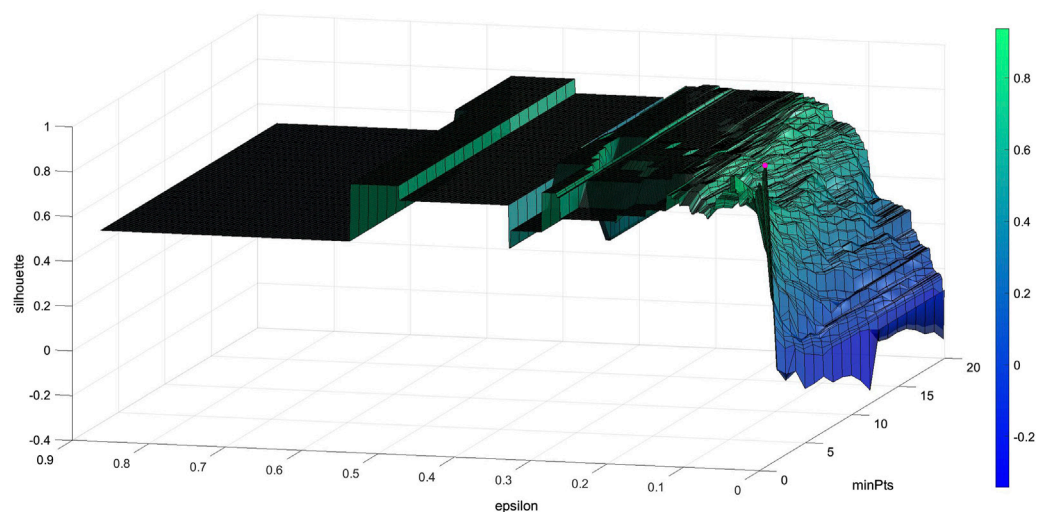


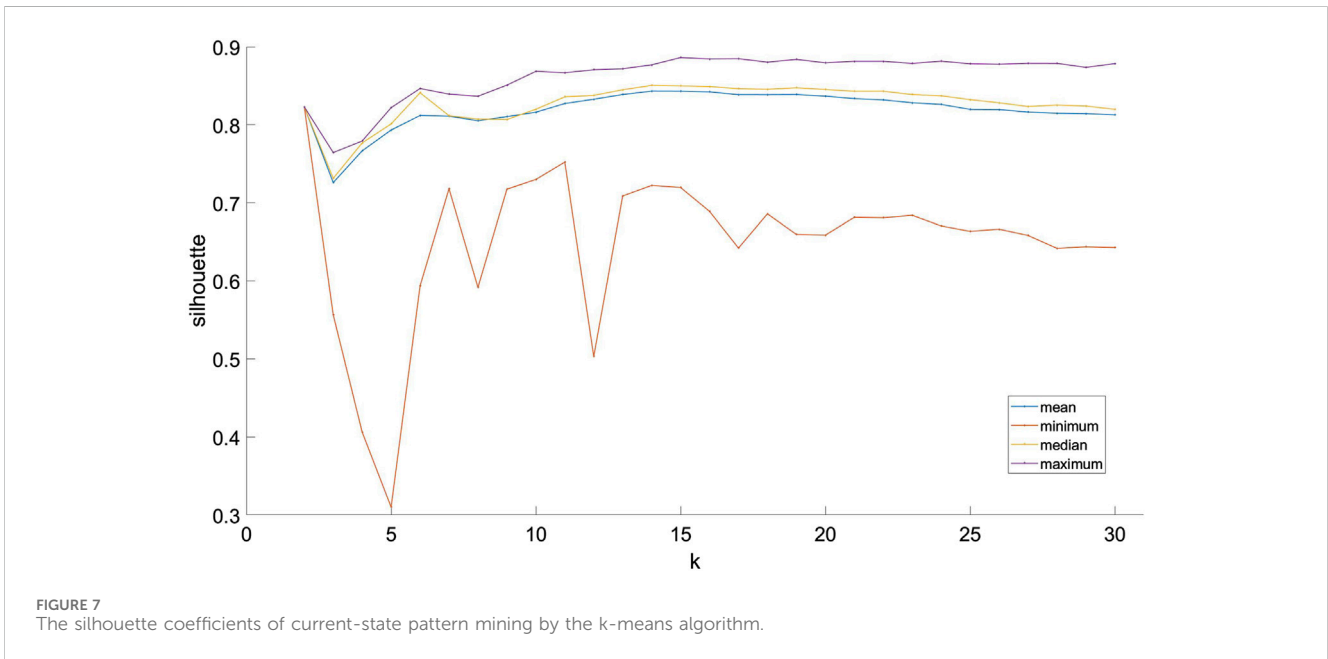
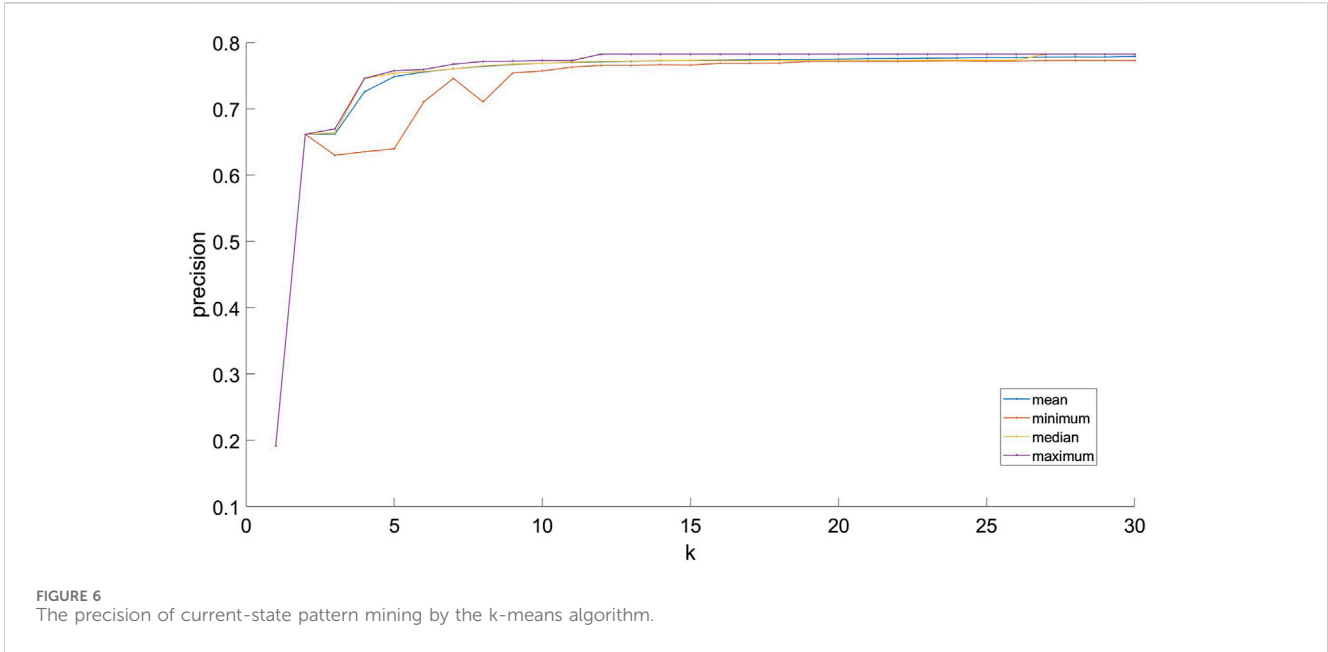
FIGURE 5
The silhouette coefficients of current-state pattern mining by Algorithm 1.

is 3,748.056 s, the maximum is 4,566.583 s, the mean is 4,118.558 s, the variance is 240.122 s, and the total time is 82,371.151 s over 20 repetitions. Given that the average time required to solve the univariate regression feature of each current sequence piece is 5.995 s, attention should be paid to the associated time constraints when the PSO algorithm is used to construct the univariate regression features of the current sequence piece in practice.

Two sets of lighting appliance fixtures are used in the experiment. Manual annotation determined that the 687 current sequence pieces in the experimental current sequence data corresponded to 13 different electricity states for the two lighting appliances. Table 2 describes those 13 electricity states. Because the

missing data phenomena happened randomly during power data acquisition, the PSO algorithm was employed to construct the univariate regression features of the current sequence pieces specified in Equation 3.

First, all 687 current sequence pieces, where each current sequence piece is represented with its univariate regression feature, are partitioned into some clusters by Algorithm 1. Each cluster is treated as one current-state pattern corresponding to one electricity state in Table 2. Figure 4 illustrates the precision of Algorithm 3 for current-state pattern mining, while Figure 5 illustrates the silhouette coefficients. Because the implementation of Algorithm 1 takes reference from the DBScan algorithm, ϵ , the neighborhood radius parameter



in Figure 4 varies from 0.001 to 1, incrementing by 0.001. Meanwhile, *minPts*, the minimum number of sample parameter for the density threshold, ranges from 1 to 20, incrementing by 1. Figure 4 illustrates the fluctuations in precision, as defined by Equation 9, with variations in the parameters *epsilon* and *minPts*. According to Figure 4, the precision of current-state pattern mining by Algorithm 3 exhibits significant variability when *epsilon* and *minPts* vary: the minimum precision is 0.1913, the maximum precision is 0.7824, the median precision is 0.6697, and the average precision is 0.5010 with 0.2561 as the standard deviation. When *epsilon* ranges from 0.001 to 0.006 and *minPts* is set to 1, the precision reaches its maximum. In Figure 4, the

purple point indicates the corresponding data point representing the maximum precision value.

Because Algorithm 1 is implemented based on the DBScan algorithm, the silhouette coefficient can also be used to evaluate the performance of current-state pattern mining by Algorithm 1. Figure 5 illustrates the variation of the average silhouette coefficient concerning changes in the values of *epsilon* and *minPts*. In Figure 5, because Algorithm 1 clusters all the current sequence pieces into one cluster when *epsilon* exceeds 0.875, the neighborhood radius parameter, *epsilon*, varies from 0.001 to 0.875, incrementing by 0.001; meanwhile, *minPts*, varies uniformly from 1 to 20, incrementing by 1. It is evident that the silhouette coefficient of

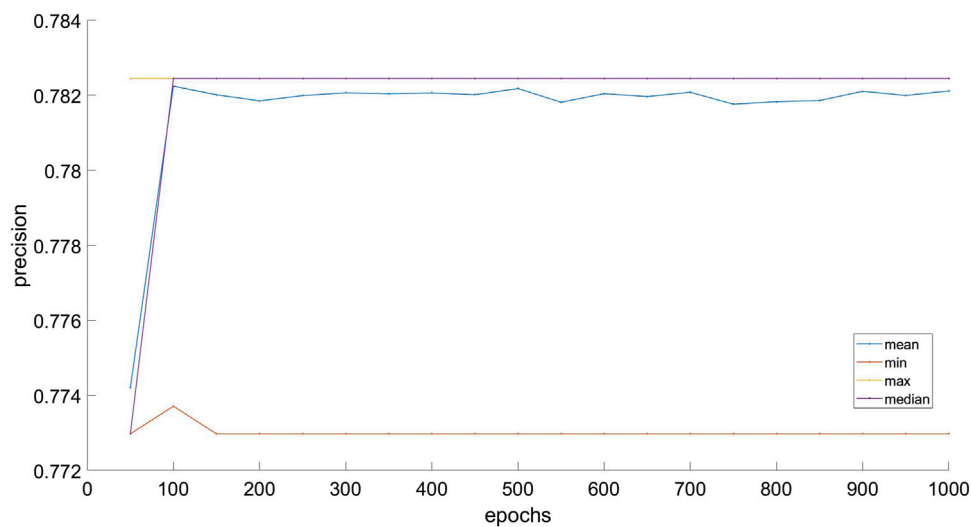


FIGURE 8
The precision of current-state pattern mining by SOM.

current-state pattern mining by Algorithm 1 exhibits significant variability when ϵ and \minPts vary. For the average silhouette coefficient, the minimum is -0.3425 , the maximum is 0.9379 , the mean is 0.6142 , the median is 0.6582 , and the standard deviation is 0.1573 . When ϵ is set to 0.005 , and \minPts is set to 1 , the average silhouette coefficient reaches its maximum of 0.9379 . In Figure 5, the maximum of the average silhouette coefficient is highlighted in purple. According to Figures 4, 5, it is easy to assert that the performance of current state pattern mining by Algorithm 1 is optimal when the value of ϵ is set to 0.005 and \minPts is set to 1 .

The k-means algorithm is a widely used clustering algorithm. For comparison, Figures 6, 7 illustrate the variations in precision and average silhouette coefficient of current-state pattern mining by the k-means algorithm based on the univariate regression feature set of current sequence pieces. When current-state patterns are mined by the k-means algorithm, for each possible value of parameter k , the process is repeatedly executed 400 times, and the precision and silhouette coefficient are recorded each time in our experiment. Figure 6 illustrates the variation of precision with increasing k . In Figure 6, k uniformly increases from 1 to 30 , incrementing by 1 . As k increases, some indicators for the precision of the current-state pattern constructed by the k-means algorithm, such as minimum, maximum, average, or median, tend to stabilize. Moreover, when $k \geq 13$, the minimum, maximum, average, and median values are all approximately 0.77 , and the values of these indicators are gradually approaching consistency. Because the k-means algorithm considers all data as one cluster when $k = 1$, the silhouette coefficient for the clustering cannot be calculated; therefore, in Figure 7, the value of k varies from 2 to 30 , incrementing by 1 . In Figure 7, when $k \geq 13$, the mean and median of the silhouette coefficients for the current-state patterns constructed by the k -means algorithm converge and approach stability. Furthermore, when $k = 14, 15$, and 16 , both the mean and the median reach their highest levels, and the average silhouette coefficient values are $0.8431, 0.8430$, and 0.8422 ; meanwhile, the median values are $0.8505, 0.8499$, and 0.8490 ,

respectively. According to information in Figures 6, 7, the best value of parameter k is $14, 15$, or 16 when the k-means algorithm is used to mine the current-state patterns.

The SOM neural network is one kind of unsupervised learning artificial neural network that is extensively used in domains such as data clustering, feature extraction, and data visualization. The experiment utilized an SOM neural network to perform cluster analysis on the univariate regression feature set of all current sequence pieces to obtain current sequence patterns. In the experiment, the SOM neural network consisted of 30 neurons arranged in five rows and six columns. Parameter epochs , representing the number of training iterations, varied uniformly from 50 to $1,000$, incrementing by 50 . For each value of epochs , the current-state patterns are constructed 400 times, and the precision and silhouette coefficient of the partition of the feature set of current sequence pieces are recorded each time. Figures 8, 9 illustrate the variations of precision and the average silhouette coefficient across different epochs.

Figure 8 illustrates the variation curve of precision of cluster partition of current state pattern mining by SOM with epoch increases. Obviously, the average precision remains relatively stable at approximately 0.78 , and the median of precision aligns with the maximum precision and stabilizes at 0.7824 when $\text{epochs} \geq 100$. Figure 9 illustrates the variation curve of the silhouette coefficient of cluster partition of current state pattern mining by SOM with epochs increases. The average silhouette coefficients remain relatively stable at around 0.82 , and the median silhouette coefficients remain relatively stable at around 0.83 when $\text{epochs} \geq 150$. According to Figures 8, 9, it can be concluded that the performance of cluster partition of current-state pattern mining by SOM is better when $\text{epochs} = 150$ or 200 .

The optimal precision and silhouette coefficient of the current-state pattern constructed by the DBSCAN algorithm, k-means algorithm, and SOM neural network are given in Table 3. In the row for the DBSCAN algorithm of Table 3, the precision value at the precision column is the maximum precision when the current-state

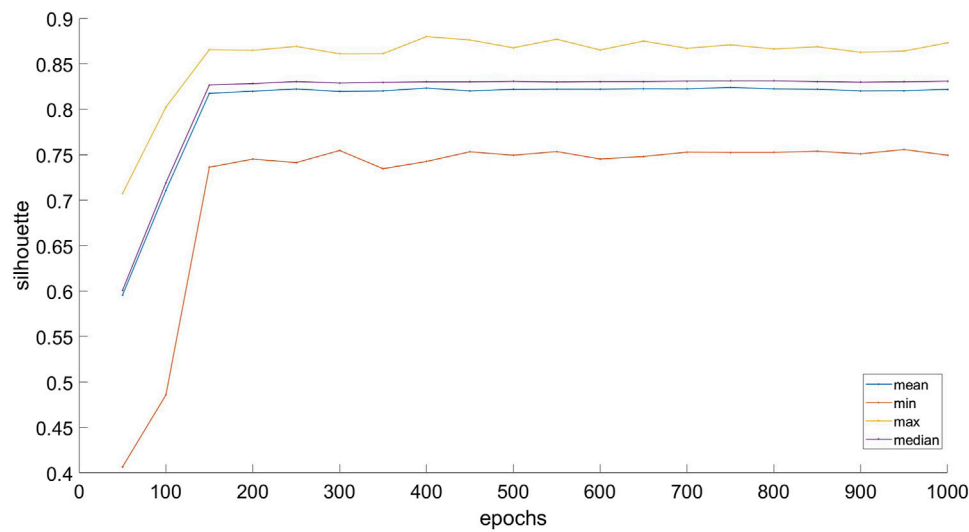


FIGURE 9
The silhouette coefficients of current-state pattern mining by SOM.

TABLE 3 Performance of current-state patterns by different clustering algorithms.

Clustering algorithm	Precision	Silhouette coefficient
DBSCAN	0.7824	0.9379
K-means	0.7789	0.8431
SOM	0.7824	0.8240

pattern is constructed by Algorithm 1 as above, and the silhouette coefficient value at the silhouette coefficient column represents the maximum average silhouette coefficient of the current-state pattern is built by Algorithm 1 as above. In the row for the k-means algorithm of Table 3, the precision value at the precision column is the maximum of average precision when the current-state pattern is constructed by the k-means algorithm 400 times as above, and the silhouette coefficient value at the silhouette coefficient column represents the maximum of average silhouette coefficient of the current-state pattern is constructed by the k-means algorithm 400 times. In the row for the SOM neural network of Table 3, the precision value at the precision column is the maximum of average precision when the current-state pattern is constructed by the SOM algorithm 400 times as above, and the silhouette coefficient value at the silhouette coefficient column represents the maximum of average silhouette coefficient of the current-state pattern is constructed by the SOM 400 times. Based on the information provided in Table 3, it can be concluded that Algorithm 1 demonstrates superior performance in partitioning current-state patterns than the k-means algorithm and SOM neural network. Thus, Algorithm 1 is deemed more suitable for current-state pattern mining.

The computer used for current-state pattern mining in this experiment is a Dawn W760-g20 server with dual Intel® Xeon® CPU E5-2637 v3 processors and 192 GB ECC memory. The operating

system is Windows 10, and the program environment is MatLab R2017a.

5 Conclusion

This study introduces the CPARSIM framework, designed to efficiently identify electrical current-state patterns and the running states of appliances by analyzing current sequence pieces. In the CPARSIM framework, the two coefficients of the univariate regression equation for each current sequence piece are treated as its features, and the PSO algorithm is applied to solve the univariate regression equation for a current sequence piece where some data may be missing. The challenge of mining current-state patterns is approached as a cluster analysis problem, with each cluster representing a distinct current-state pattern. A novel current-state pattern mining method is proposed based on the DBSCAN algorithm. In our experiments, clustering analysis algorithms, such as the k-means algorithm, DBSCAN algorithm, and SOM neural network, are individually used to cluster the feature set of current sequence pieces. Experimental results show that the current-state pattern mining by the DBSCAN algorithm is more suitable for identifying running states of appliances based on online current sequence pieces in real scenarios.

In this article, the tactics of adjacent data differencing are used to identify pieces of the online current sequence. The online data may exhibit more significant fluctuations if there is significant interference on the circuit, and those fluctuations may result in a single current sequence piece being identified as multiple pieces, as well as significant drifting of the features of these pieces. To ensure the completeness and stability of the current sequence pieces for the same event, techniques such as applying sliding windows, fixed time interval sampling, event triggering, and methods based on load switching points are exciting topics. Additionally, the standardization and normalization of data and the analysis and interpretation of

appliances running states corresponding to various current-state patterns based on cluster analysis results are valuable topics for future research.

Furthermore, this article uses univariate regression features to represent variable-length current sequence pieces, and each sequence piece is treated as one point within the two-dimensional feature space. In the future, Fourier transformation can extract the spectral features of the current sequence pieces, or statistical features such as mean, variance, and peak value could be used to represent variable-length current sequence pieces in a unified dimension. Representing variable-length current sequence pieces in a suitable dimension is worth attention in different application scenarios.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

ZZ: Writing–review and editing, Writing–original draft, Software, Resources, Project administration, Methodology, Funding acquisition, Data curation, and Conceptualization. HC: Writing–review and editing, Writing–original draft, Visualization, Validation, Software, Resources, Investigation, Funding acquisition, Data curation, and Conceptualization. PW: Writing–review and editing, Visualization, Validation, Software, Investigation, Formal analysis, and Conceptualization. SZ: Writing–review and editing, Visualization, Supervision, and Methodology.

References

- Ahmed, S., Ali, A., Ciocia, A., and D'Angola, A. (2024). Technological elements behind the renewable energy community: current status, existing gap, necessity, and future perspective—overview. *Energies* 17 (13), 3100. doi:10.3390/en17133100
- Aslan, M., and Nur, Z. E. (2022). An efficient hybrid model for appliances classification based on time series features. *Energy Build.* 266, 112087. doi:10.1016/j.enbuild.2022.112087
- Avancini, D. B., Rodrigues, J. J. P. C., Rabêlo, R. A. L., Das, A. K., Kozlov, S., and Solic, P. (2021). A new IoT-based smart energy meter for smart grids. *Int. J. Energy Res.* 45 (1), 189–202. doi:10.1002/er.5177
- Balakumar, P., Vinopraba, T., and Chandrasekaran, K. (2023). Machine learning based demand response scheme for IoT enabled PV integrated smart building. *Sustain. Cities Soc.* 89, 104260. doi:10.1016/j.scs.2022.104260
- Caldera, M., Hussain, A., Romano, S., and Re, V. (2023). Energy-consumption pattern-detecting technique for household appliances for smart home platform. *Energies* 16 (2), 824. doi:10.3390/en16020824
- Chen, J. F., Wang, X., Zhang, X. T., and Zhang, W. H. (2022). Temporal and spectral feature learning with two-stream convolutional neural networks for appliance recognition in NILM. *IEEE Trans. Smart Grid* 13 (1), 762–772. doi:10.1109/TSG.2021.3112341
- Chen, K., Zhang, Y., Wang, Q., Hu, J., Fan, H., and He, J. (2020). Scale- and context-aware convolutional non-intrusive load monitoring. *IEEE Trans. Power Syst.* 35 (3), 2362–2373. doi:10.1109/TPWRS.2019.2953225
- Chen, X., Dong, H., Zhang, Y., Che, Y., Chen, L., and Huang, A. (2023). Coordinated emergency control strategy of high-voltage direct current transmission and energy storage system based on Pontryagin minimum principle for enhancing power system frequency stability. *IET Renew. Power Gener.* 17 (7), 1680–1698. doi:10.1049/rpg2.12703
- Chou, J. S., and Truong, N. S. (2019). Cloud forecasting system for monitoring and alerting of energy use by home appliances. *Appl. Energy* 249, 166–177. doi:10.1016/j.apenergy.2019.04.063
- Çimen, H., Çetinkaya, N., Vasquez, J. C., and Guerrero, J. M. (2021). A microgrid energy management system based on non-intrusive load monitoring via multitask learning. *IEEE Trans. Smart Grid* 12 (2), 977–987. doi:10.1109/TSG.2020.3027491
- Diawuo, F. A., Sakah, M., de la Rue du Can, S., Baptista, P. C., and Silva, C. A. (2020). Assessment of multiple-based demand response actions for peak residential electricity reduction in Ghana. *Sustain. Cities Soc.* 59, 102235. doi:10.1016/j.scs.2020.102235
- El-Sayed, W. T., Awad, A. S. A., Al-Abri, R., Alawasa, K., Onen, A., and Ahshan, R. (2024). Integrated planning of hydrogen supply chain and reinforcement of power distribution network for accommodating fuel cell electric vehicles. *Int. J. Hydrogen Energy* 81, 865–877. doi:10.1016/j.ijhydene.2024.07.316
- Fabricio, M. A., Behrens, F. H., and Bianchini, D. (2020). Monitoring of industrial electrical equipment using IoT. *IEEE Lat. Am. Trans.* 18 (8), 1425–1432. doi:10.1109/TLA.2020.9111678
- Gennitsaris, S., Oliveira, M. C., Vris, G., Bofilios, A., Ntinou, T., Frutuoso, A. R., et al. (2023). Energy efficiency management in small and medium-sized enterprises: current situation, case studies and best practices. *Sustainability* 15 (4), 3727. doi:10.3390/su15043727
- Hart, G. W. (1992). Nonintrusive appliance load monitoring. *Proc. IEEE* 80 (12), 1870–1891. doi:10.1109/5.192069
- Hasanvand, A., Savaedi, M., Hajivand, A., and Hamrani Salemi, H. (2024). Harnessing tidal energy efficiency: a comprehensive analysis of tandem flapping

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This work is partially supported by the Discipline (Major) Top-notch Talent Academic Funding Project of Anhui Provincial University and College (gxbjZD2021067 and gxyq2022030), the Innovative Leading Talents Project of Anhui Provincial Special Support Program [(2022)21], the Key Project of Natural Science Research of Universities of Anhui Province (2024AH050246), and the director foundation of the Anhui Province Key Laboratory of Intelligent Building & Building Energy Saving (No. IBES2022ZR01).

Conflict of interest

The 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.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

- hydrofoils for maximizing power generation from low-level currents. *Ocean. Eng.* 310, 118673. doi:10.1016/j.oceaneng.2024.118673
- Himeur, Y., Alsalemi, A., Bensaali, F., and Amira, A. (2020). Robust event-based non-intrusive appliance recognition using multi-scale wavelet packet tree and ensemble bagging tree. *Appl. Energy* 267, 114877. doi:10.1016/j.apenergy.2020.114877
- Himeur, Y., Elnour, M., Fadli, F., Meskin, N., Petri, I., Rezgui, Y., et al. (2023). AI-big data analytics for building automation and management systems: a survey, actual challenges and future perspectives. *Artif. Intell. Rev.* 56 (6), 4929–5021. doi:10.1007/s10462-022-10286-2
- Hou, J., Qin, X., Fan, Y., Song, G., Wu, X., Li, P., et al. (2024). A selective and staged active current limiting control strategy for fault identification in MMC HVDC power grids. *Int. J. Electr. Power Energy Syst.* 160, 110143. doi:10.1016/j.ijepes.2024.110143
- Kong, W. C., Dong, Z. Y., Hill, D. J., Ma, J., Zhao, J. H., and Luo, F. J. (2018). A hierarchical hidden Markov model framework for home appliance modeling. *IEEE Trans. Smart Grid* 9 (4), 3079–3090. doi:10.1109/TSG.2016.2626389
- Kumar, N. (2024). EV charging adapter to operate with isolated pillar Top solar panels in remote locations. *IEEE Trans. Energy Convers.* 39 (1), 29–36. doi:10.1109/TEC.2023.3298817
- Kumar, N., Saxena, V., Singh, B., and Panigrahi, B. K. (2023a). Power quality improved grid-interfaced PV-assisted onboard EV charging infrastructure for smart households consumers. *IEEE Trans. Consumer Electron.* 69 (4), 1091–1100. doi:10.1109/TCE.2023.3296480
- Kumar, N., Singh, H. K., and Niwareeba, R. (2023b). Adaptive control technique for portable solar powered EV charging adapter to operate in remote location. *IEEE Open J. Circuits Syst.* 4, 115–125. doi:10.1109/OJCS.2023.3247573
- Liu, H., Wu, H. P., and Yu, C. M. (2019). A hybrid model for appliance classification based on time series features. *Energy Build.* 196, 112–123. doi:10.1016/j.enbuild.2019.05.028
- Mohi-Ud-Din, G., Marnerides, A. K., Shi, Q., Dobbins, C., and Macdermott, A. (2021). Deep cola: a deep COmpetitive learning algorithm for future home energy management systems. *IEEE Trans. Emerg. Top. Comput. Intell.* 5 (6), 1–11. doi:10.1109/TETCI.2020.3027300
- Naidu, S., Chand, A., Pandaram, A., and Vosikata, S. (2024). Electricity consumption, renewable energy production, and current account of organisation for economic Co-operation and development countries: implications for sustainability. *Sustainability* 16 (9), 3722. doi:10.3390/su16093722
- Phangbertha, L. N., Fitri, A., Purnamasari, I., and Muliono, Y. (2019). Smart socket for electricity control in home environment. *Procedia Comput. Sci.* 157, 465–472. doi:10.1016/j.procs.2019.09.002
- Priyadharshini, S. G., Subramani, C., and Preetha Roselyn, J. (2021). Development of intelligent smart metering system through remote monitoring and control under robust conditions. *J. Test. Eval.* 49 (4), 2370–2387. doi:10.1520/JTE20200218
- Rago, A., Piro, G., Boggia, G., and Dini, P. (2021). Anticipatory allocation of communication and computational Resources at the edge using spatio-temporal dynamics of mobile users. *IEEE Trans. Netw. Serv. Manag.* 18 (4), 4548–4562. doi:10.1109/TNSM.2021.3099472
- Ramson, S. R. J., Vishnu, S., Kirubaraj, A. A., Anagnostopoulos, T., and Abu-Mahfouz, A. M. (2022). A LoRaWAN IoT-enabled trash bin level monitoring system. *IEEE Trans. Industrial Inf.* 18 (2), 786–795. doi:10.1109/TII.2021.3078556
- Ridi, A., Gisler, C., and Hennebert, J. (2013). “Automatic identification of electrical appliances using smart plugs,” in 2013 8th International Workshop on Systems, Signal Processing and Their Applications (WoSSPA) (Algiers, Algeria: IEEE), 301–305. doi:10.1109/WoSSPA.2013.6602380
- Saleem, M. U., Usman, M. R., and Shakir, M. (2021). Design, implementation, and deployment of an IoT based smart energy management system. *IEEE Access* 9, 59649–59664. doi:10.1109/ACCESS.2021.3070960
- Sayed, A., Himeur, Y., Alsalemi, A., Bensaali, F., and Amira, A. (2022). Intelligent edge-based recommender system for Internet of energy applications. *IEEE Syst. J.* 16 (3), 5001–5010. doi:10.1109/JSYST.2021.3124793
- Solatidehkordi, Z., Ramesh, J., Al-Ali, A. R., Osman, A., and Shaaban, M. (2023). An IoT deep learning-based home appliances management and classification system. *Energy Rep.* 9 (3), 503–509. doi:10.1016/j.egyr.2023.01.071
- Swindiarto, V. T. P., Sarno, R., and Novitasari, D. C. R. (2018). “Integration of fuzzy C-means clustering and TOPSIS (FCM-TOPSIS) with silhouette analysis for multi criteria parameter data,” in 2018 International Seminar on Application for Technology of Information and Communication: Creative Technology for Human Life, iSemantic (Semarang, Indonesia: IEEE), 463–468. doi:10.1109/ISEMANTIC.2018.8549844
- Ullah, M., Narayanan, A., Wolff, A., and Nardelli, P. H. J. (2022). Industrial energy management system: design of a conceptual framework using IoT and big data. *IEEE Access* 10, 110557–110567. doi:10.1109/ACCESS.2022.3215167
- Wang, J., and Zhang, S. (2022). Anomaly detection for power consumption patterns based on linear discriminant analysis and density peak clustering. *Dianli Xi. Zidonghua/Automation Electr. Power Syst.* 46 (5), 87–95. doi:10.7500/AEPS20210717003
- Wang, L. C., Cho, W. T., Chiu, Y. S., and Lai, C. F. (2013). “A parallel multi-appliance recognition for smart meter,” in Proceedings - 2013 IEEE 11th International Conference on Dependable, Autonomic and Secure Computing (DASC) (Chengdu, Sichuan, China: IEEE), 475–480. doi:10.1109/DASC.2013.110
- Yan, D., Jin, Y., Sun, H., Dong, B., Ye, Z., Li, Z., et al. (2019). Household appliance recognition through a Bayes classification model. *Sustain. Cities Soc.* 46, 101393. doi:10.1016/j.scs.2018.12.021
- Yang, Y., Wei, H., Wang, J., Fan, S., Wu, H., and Qiu, A. (2024). Profile of inner magnetically insulated transmission lines to minimize energy loss under MA/Cm lineal current density for high-current pulsed-power accelerators. *IEEE Trans. Plasma Sci.* 52 (4), 1372–1379. doi:10.1109/TPS.2024.3388712
- Zhou, Y., Wu, Y., Dong, Z., Hu, Y., Xiao, X., and Zhang, S. (2022). Non-intrusive load monitoring based on motif mining and harmonic function based semi-supervised learning. *Dianli Zidonghua Shebei/Electric Power Autom. Equip.* 42 (7), 3–10. doi:10.16081/j.epae.202202013
- Zhu, T., Ai, Q., He, X., Li, Z., Sun, D., and Li, X. (2020). An overview of data-driven electricity consumption behavior analysis method and application. *Dianwang Jishu/Power Syst. Technol.* 44 (9), 3497–3507. doi:10.13335/j.1000-3673.pst.2020.0226a