Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 19(11): 11595–11627.
DOI: 10.3934/mbe.2022540
Received: 04 July 2022
Revised: 26 July 2022
Accepted: 03 August 2022
Published: 12 August 2022

Research article

Reinforced MCTS for non-intrusive online load identification based on cognitive green computing in smart grid

Yanmei Jiang[1,4], Mingsheng Liu[1,*], Jianhua Li[2] and Jingyi Zhang[3]

[1] State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, No. 5340 Xiping Road, Beichen District, Tianjin 300401, China

[2] School of Information Science and Technology, Shijiazhuang TieDao University, No. 17 East North Second Ring Road, Changan District, Shijiazhuang 050043, China

[3] School of Cyber Science and Technology, Beihang University, No. 37 Xue Yuan Road, Haidian District, Beijing 100191, China

[4] Department of Rail Transportation, Hebei Jiaotong Vocational and Technical College, No. 219 The pearl river avenue, Shijiazhuang 050035, China

* **Correspondence:** Email: liums601001@sina.com.

Abstract: Cognitive green computing (CGC) is widely used in the Internet of Things (IoT) for the smart city. As the power system of the smart city, the smart grid has benefited from CGC, which can achieve the dynamic regulation of the electric energy and resource integration optimization. However, it is still challenging for improving the identification accuracy and the performance of the load model in the smart grid. In this paper, we present a novel algorithm framework based on reinforcement learning (RL) to improve the performance of non-invasive load monitoring and identification (NILMI). In this model, a knowledge base of load power facilities (LPF-KB) architecture is designed to facilitate the load data-shared collection and storage; utilizing deep convolutional neural networks (DNNs) structure based on the attentional mechanism to enhance the representations learning of load features; using RL-based Monte-Carlo tree search (MCTS) method to construct an optimal strategy network, and to realize the online combined load prediction without relying on the prior knowledge. We use the massive experiment on the real-world datasets of household appliances to evaluate the performance of our method. The experimental results show that our approach has remarkable performance in reducing the load online identification error rate. Our model is a generic model, and it can be widely used in practical load monitoring identification and the power prediction system.

Keywords: cognitive green computing; deep learning; load identification; Monte-Carlo search tree; reinforcement learning; non-invasive load monitoring and identification

1. Introduction

Nowadays, as the Industrial Internet of Things (IIoT) is racing ahead in smart cities, various industries make full use of CGC to get more economic value and greater profits [1–3]. Meanwhile, green computing advocates producing minimal carbon footprint, and energy optimization for the long-term sustainable development of the power enterprise. As the driving force of the IIoT system in the smart city, the energy industry utilizes the CGC to improve the utilization of the limited electrical energy resources and alleviate the energy crisis. As the core part of the IIoT system, the smart grid utilizes NILMI technology to deeply mine the power consumption information of the various electrical equipment and to provide efficient technical support for optimizing the net-side and intelligent management of the load-side [4–6]. Utilizing the concept of CGC, NILMI is an efficient way to improve the system performance of the smart grid by monitoring and identifying the large masses of electrical load. Meanwhile, the rapid development of machine learning, using deep learning (DL) technology to address the bottleneck problems of NILMI, has gradually become a hot topic, which can improve the performance and stability of the NILMI model [7, 8]. However, the accuracy and the reliability of the NILMI model are still the challenges, which have hindered the development of CGC in the smart grid. The existing methods have low precision of the load recognition for the multiple sources load sampled data, which leads to the poor performance of NILMI in the energy IoT system [9, 10]. There are several reasons: the accuracy of the online load identification is affected by the incomplete load feature representation; the stability of the model is poor in the load dynamics consumption, especially in the superimposed scenario of the multiple electrical devices [11, 12]. To this end, we construct an algorithm framework of MCTS RL-based, and this model framework is a heuristic strategy optimization way to realize the identification and prediction of load power consumption in the online scenario, which can improve the performance of CGC in the smart grid.

By contrast, the non-intrusive approach is more competitive in the energy IoT system based on CGC [13, 14]. They only rely on the measured value of the load waveform (current, voltage, and power signal) as the available critical for load identification. Further enhancement of the discriminability of the load features can improve the efficiency of decision optimization of the identification model [15, 16]. So, plenty of literature and research have taken a lot of studies for extracting the discriminative load features of these load waveform signals; the majority of existing methodologies include the statistical optimization method, artificial neural networks (ANN) method, and DL algorithms [17–19].

At present, DL-based methods are divided into supervised learning and unsupervised learning. Supervised learning used a lot of training data set to obtain an optimal model, and to map all the input to the corresponding output [20–22]. The several common approaches include k-Nearest Neighbor (kNN) [23, 24], artificial neural networks (ANN) [25], and support vector machines (SVM) [26]. In [23], a hybrid structure based on an improved weighted kNN and the overshoot multiples is employed to train the sampled data sets, and this method enhances the identification accuracy of the load sampled data sets on a small scale. A similar method is mentioned in the other literature, the combination of kNN and the similar time window (STW) algorithm is proposed, but this method still has a limited scope of application. In [27], the author utilizes the SVM model with Dempster-Shafer evidence theory for the load identification, but the influence of the computational complexity on model performance is not considered. In [28], the author exploits an improved ANN algorithm to design the parallel computing model and optimize the model performance of NILMI; finally, the result shows that the method can reduce the heavy calculation in the initial stage of the training model.

Considering the advantages of the convolutional neural networks (CNNs) for image recognition, plenty of literature have proved that the CNN model has demonstrated their advancement in load identification and energy prediction [29, 30], especially the various recursive neural network (RNN) [31,32] algorithms. However, supervised learning methods still have the following problems: 1) a large number of annotated data sets are needed for training the model, which will lead to the computational burden; 2) there is the risk of over-reliance on prior knowledge, and it is also easier for a final model performance limitation; 3) samples imbalance result in the poor identification effect and the over-fitting; 4) there are the local optimization problems in online load identification.

By comparison, unsupervised learning is not limited to modeling the prior knowledge, but utilizing the directly unlabelled data sets modeling. The most commonly-used algorithms include clustering analysis (CA) [33, 34] and Hidden Markov Model (HMM) [35] algorithms. These methods are attractive in training the general-purpose system on the small data sets and do not require annotation; they take advantage of the deployment capabilities of the environment to simplify the operations process, which is a very effective way of improving the performance of CGC. In [36], a modified fuzzy clustering algorithm is employed to achieve the load monitoring in NILMI, and the simulation results show that the method demonstrates excellent monitoring performance. In [37], a Markov-based model is constructed to achieve the energy consumption identification in the household; the method utilizes the cost of minimizing the computational power to improve the algorithm performance. However, unsupervised learning methods still have their limitations: when the load features space is large, the performance of recognition and prediction algorithms is not very good; the model performance is unstable, and the better the clustering effect, the lower the identification accuracy, especially in the complex combined load signal; the limited capacity of feature learning restricts the improvement of load recognition accuracy.

To address the mentioned problems, we present a novel algorithm framework based on RL, namely MCTS-CI. MCTS-CI aims to improve the stability and performance of the combined load online identification model in a self-learning way. This framework is an online optimization algorithm and can realize independent learning by using various related strategies. In the MCTS-CI model, we design a fusion DNN model to extract the load feature and utilize the recurrent neural network based on time sequence to complete the load spatio-temporal representation in the frequency domain, which can enhance the representation learning ability of the online combined load. To improve the accuracy of the online combined load identification in the self-learning way, we employ MCTS to build an optimized strategy network. An attentional mechanism is introduced into the structure of DNN, namely ATM-DNN, as the initial policy network of MCTS to replace the random selection, which can only focus on the specific feature of load and ignore the irrelevant feature information.

The highlight contributions of our research include the following aspects:

- Discriminative load features learning. Building ATM-DNN model to enhance the learning ability of the discrimination load feature, which mainly focuses on the specific information.
- Comprehensive load-shared knowledge base. Designing the knowledge base of load power facilities (LPF-KB) to facilitate the load collection and features storage, which can function as an auxiliary means to help the formation of the load feature mapping.
- Optimal decision mechanism. MCTS model is proposed to establish a strategic network and to

obtain the optimal decision by self-learning way for improving the accuracy of the multi-load identification.

The rest of the four sections are arranged as follows: the related work and the systematical arrangement for the paper are presented in Section II. The algorithm framework of MCTS-CI is introduced in Section III. In Section IV, the experimental details, the criterion of performance evaluation, and the discussion of the results that compared with other methods are described. The conclusion of the article is demonstrated in Section V.

2. Related work and systematical arrangement

This chapter mainly reviews the related studies of online load identification and then presents the applications of RL-based MCTS.

2.1. Online load identification in NILMI

NILMI is a very effective method to improve the efficiency of load-side management in the smart grid. Especially, the applications of DL algorithms have brought more competitive effects to NILMI.

In [38], Hilbert Transform long short-term memory (HT-LSTM) model is used to improve the accuracy of load identification, and to reduce the fluctuations of the transient signal in the identification results. However, this method is only suitable for a small load sample space, and it will weaken the generalization of the model. To this end, a model based on data augmentation is presented to produce the synthetic load data for the target device, which is aim at the status switching of the appliance [39]; the results prove that using synthetic data can enhance the generalization performance of the model. Some studies concentrate on the high-quality load features learning, which determines the accuracy of load identification. In [31], a two-layer DL model based on the attention mechanism is employed to address the matter of focus from the load feature learning; the results indicate that the attention mechanisms are more advantageous for enhancing the model robustness, which ensures the stability of the model in the load combination scenarios. Multi-load combination identification and power forecasting are more challenging compared to one-class load. In [40], an improved hidden Markov model based on a Viterbi algorithm (VA) is used to achieve the multi-load identification, and the method can effectively preserve the temporal correlation of the electrical load.

Recently, the relevant literature shows that online load power prediction has gradually become a challenging research topic. In [41], the author utilizes a hybrid algorithm architecture of Fisher information theory and the online support vector regression (OSVR) to improve the load prediction accuracy. However, It is incredible to accurately capture the dynamic load features in the online operating mode by using the trained model on offline load data sets. In [42], the author uses the hidden Markov model to present an adaptive online learning method, and the method is used to update the model parameters recursively; the optimal parameters are applied to the load energy consumption prediction; the experimental result shows that the method can improve the reliability of load prediction performance. The same as a Markov decision process, another article proposes a deep RL framework to realize the energy fine-tuning and forecasting [43]. Experiments prove that the method can overcome the uncertainty of the load energy forecasting effectively, and optimize the design of the continuous-control model parameter in high-dimensional variables. As an improved

Markov decision process, RL is gradually used for the online load identification and power consumption prediction, and to improve the efficiency of energy management [44–46].

2.2. *Monte-Carlo tree search RL-based*

Commonly, the RL model is used to describe and solve the maximum returns between an agent and the environment. RL is also applied to solve a balance between the exploration and the exploitation in the online learning of load feature [47–49]. Compared to supervised learning and unsupervised learning, RL does not rely on the pre-trained model, but it is the sequential decision-making to require a continuous selection of actions, and then to achieve the maximum benefit from completing these actions as the best outcome. In this process, the learning strategies are built, and then the parameters are updated [50]. RL has been widely used in various fields and has produced remarkable results for NILMI.

As a heuristic search algorithm in the decision-making process, MCTS has been used as an assessment strategy for RL. For instance, AlphaGo is the classic application of MCTS [51]. In recent years, much literature present that MCTS can make the optimal decision-making in dealing with online dynamically changing data. In [58], the author utilizes a model-based RL method to realize the online scheduling of the residential microgrid without the preset prediction model, and then MCTS is used to find the optimal strategy network. In [59], MCTS is employed to solve the high dependence on the prior model of the hierarchical task network (HTN) planning; in the model, MCTS can help HTN to choose the optimal decomposition approach. Although MCTS has made excellent contributions in many fields, the application is limited in NILMI, especially in the online combined load scenes. In this paper, a novelty algorithm framework RL-based MCTS is proposed to realize the online combined load identification and prediction.

3. Designed of MCTS-CI framework

In this section, we introduce the algorithm framework of MCTS-CI model. This model aims to realize the online combined load identification and prediction. The schematic diagram of the overall design for the proposed model is shown in Figure 1. Our design inspiration comes from AlphaGo Zero [51], which has the self-directed learning ability, and this means the improvement of learning ability by playing against itself. We introduce the actual process of NILMI, the design principle of the knowledge base of load power facilities, the detailed structure of ATM-DNN, and the load recognition mechanism based on MCTS-CI.

3.1. *NILMI*

In the smart grid IoT system, by measuring and analyzing the real-time power consumption of the electrical equipment, NILMI can realize the actual load consumption prediction of household electricity in the short and medium term [52, 53]. In the current study of NILMI, the load signals are divided into transient and steady-state signals [54]. However, the load transient signal has some limitations: the short load monitoring time will result in the incomplete acquisition of the load characteristic information; more expensive monitoring equipment is needed to maintain the load monitoring accuracy; the accuracy is not satisfactory in online load combination identification scenarios. So, in this paper, we monitor and collect the steady-state power signals of the electrical load for the online load identification.

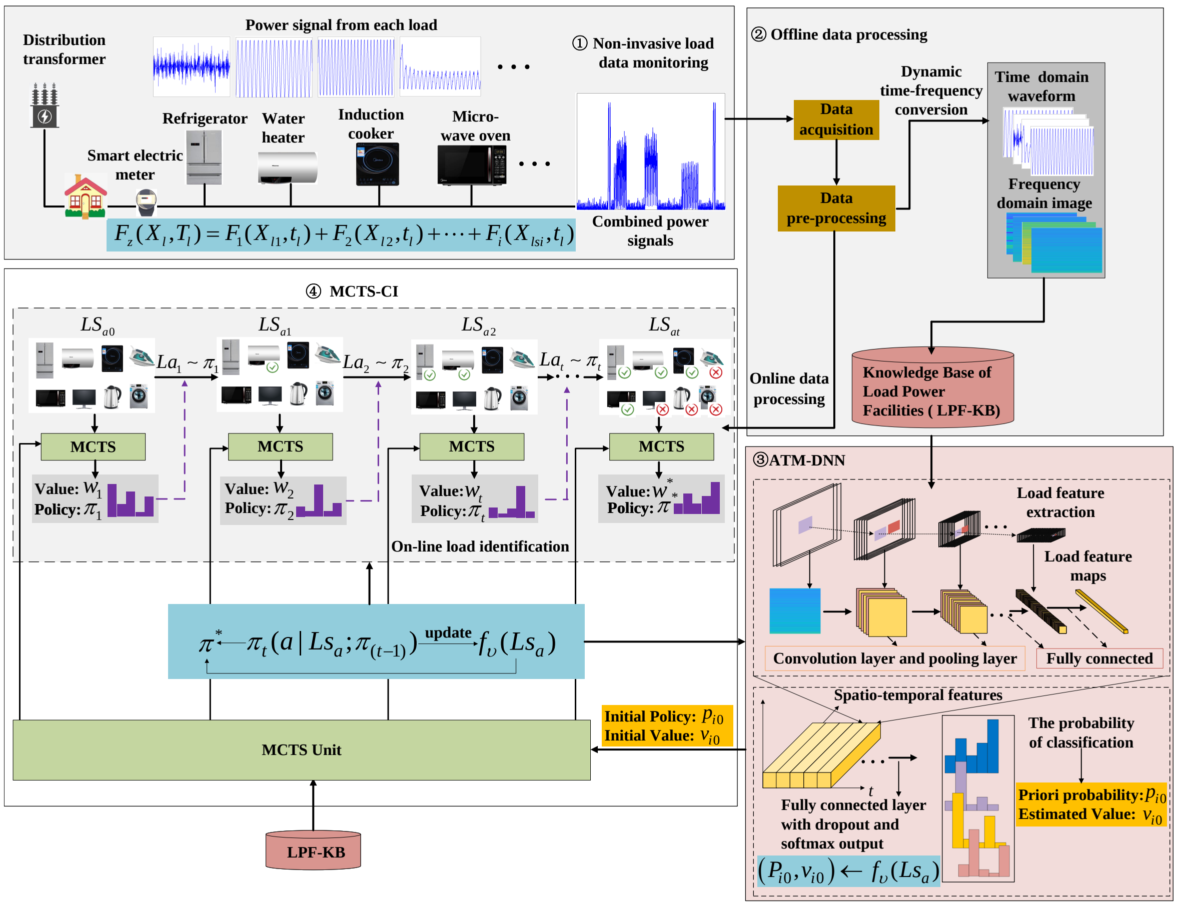
① Non-invasive load data monitoring
Power signal from each load
Distribution transformer
Smart electric meter
Refrigerator
Water heater
Induction cooker
Micro-wave oven
Combined power signals
$F_z(X_l,T_l)=F_1(X_{l1},t_l)+F_2(X_{l2},t_l)+\cdots+F_i(X_{lsi},t_l)$
② Offline data processing
Data acquisition
Data pre-processing
Dynamic time-frequency conversion
Time domain waveform
Frequency domain image
Online data processing
Knowledge Base of Load Power Facilities (LPF-KB)
④ MCTS-CI
MCTS
Value: W_1 Policy: π_1
On-line load identification
$\pi^* \leftarrow \pi_t(a\mid Ls_a;\pi_{(t-1)}) \xrightarrow{update} f_\upsilon(Ls_a)$
MCTS Unit
Initial Policy: p_{i0} Initial Value: v_{i0}
LPF-KB
③ATM-DNN
Load feature extraction
Load feature maps
Convolution layer and pooling layer
Fully connected
Spatio-temporal features
Fully connected layer with dropout and softmax output
The probability of classification
Priori probability: p_{i0} Estimated Value: v_{i0}
$(P_{i0},v_{i0}) \leftarrow f_\upsilon(Ls_a)$

Figure 1. The overall design structure of the proposed model. The model structure consists of four main parts: ① non-invasive load data monitoring; ② offline load data processing and building the knowledge base of load power facilities; ③ ATM-DNN model for forming the spatio-temporal features of load and achieving the initial policy of MCTS; ④ structuring the MCTS-CI model to realize the online load combination identification.

In general, when the supply voltage meets the national standard, the steady-state current of the electrical equipment has a certain statistical law [55]. We suppose the steady-state load power also satisfies the load additivity criterion; the combined load power at time t_l is described as the sum of load power from the individual electrical equipment:

$$P_z(X_l,t_l) = P_1(X_{lsi},t_l) + P_2(X_{lsi},t_l) + P_3(X_{lsi},t_l) + \cdots + P_n(X_{lsi},t_l) + \sigma(t_l), \qquad (3.1)$$

where $P_z(X_l,t_l) \in R$ (R is the real set) represents the total power of the load at time t_l; X_l represents the status of the electrical equipment, $si \in [0,1,2,3,\cdots,M]$, $si = 0$ indicates that the device has stopped and $si = 1$ indicates that the device is running in the status X_1; $P_n(X_{lsi},t_l)$ represents the household appliance n is running in the state X_{lsi} at time t_l, where $n \in N$ and N is the number of the household appliances; $\sigma(t_l)$ represents the measurement error and noise. The expression of the load power is as

$$P(X_l, t_l) = \sum_{r=0}^{N} P_r = \sum_{r=0}^{N} V_r I_r \cos(\phi_r), \tag{3.2}$$

where V represents the magnitude of the voltage, I represents the magnitude of the current, ϕ is the phase angle, and r is the initial phase angle of the first harmonic component. Figure 2 reveals the load power waveform of the different household appliances and the combined load power waveform from a day of a real house.

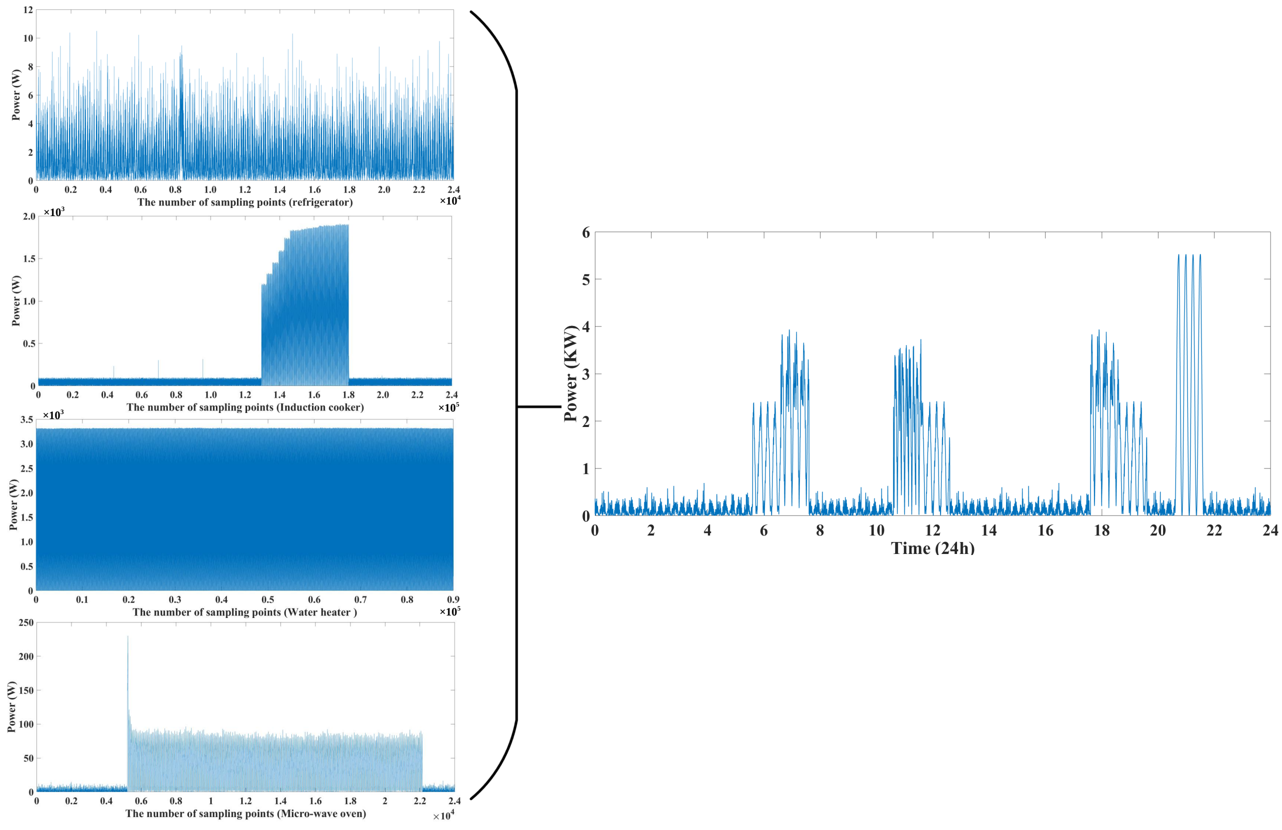

Figure 2. Load power waveform of the different household appliances in a day from a real house.

3.2. *Design and implementation method of LPF-KB*

In the paper, we first introduce the knowledge base of load power facilities (LPF-KB), and it is used to realize the collection and storage of the load sampled data sets. As an important part of the load identification model, LPF-KB is formed by the time-domain waveform of the load and the distribution of the frequency-domain image features at different times from the various household appliances. LPF-KB is the offline data sets of our model, and mainly realizes the data sharing and the critical load features management base on time series. The detailed structure design of LPF-KB is shown in Figure 3. In our paper, considering that the load type of households' electrical equipment and electricity usage are relatively stable, and then the steady-state load power signals are selected as the load signature (LS) to represent the load characteristics. We collect the power signals of the electricity load in real-time and to utilize the dynamic time-frequency conversion technology obtain the frequency-domain image of the load power signals.

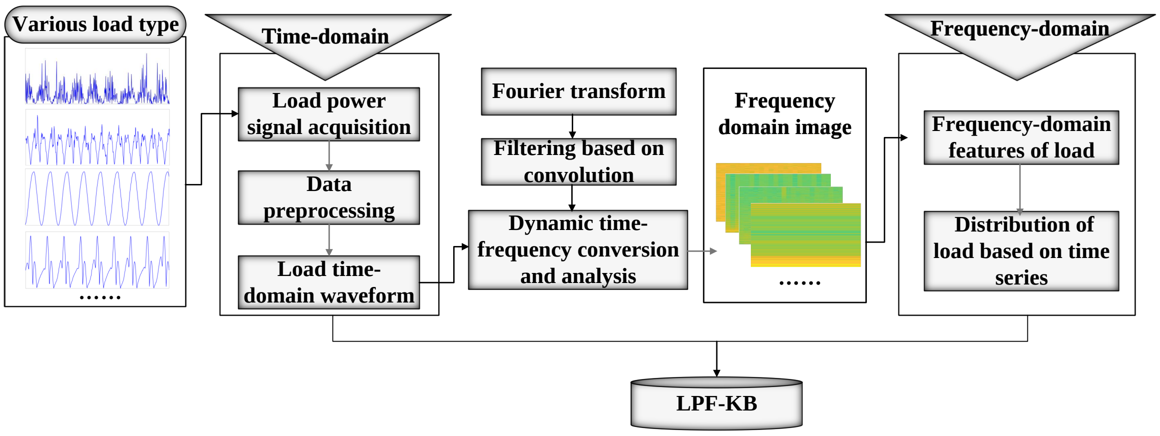
Various load type
……
Time-domain
Load power signal acquisition
Data preprocessing
Load time-domain waveform
Fourier transform
Filtering based on convolution
Dynamic time-frequency conversion and analysis
Frequency domain image
……
Frequency-domain
Frequency-domain features of load
Distribution of load based on time series
LPF-KB

Figure 3. Detailed structural design of LPF-KB.

In LPF-KB structure, we suppose that the known household appliances database $F(t)$, where, $F(t) = [F_1, F_2, \cdots, F_i]^T$. Each of $F_i = (f_{i,1}, f_{i,2}, \cdots, f_{i,N})$, where $F(t)$ represents the knowledge base of load power facilities at time t, $f_{i,N}$ represents the operating state of the i^{th} load power at time t, N is the number of the facility operating state.

To improve the effectiveness of the load sampled data and to reduce the parameters complexity of the input of the neural network, we use the convolutional filtering (CF) method [56]. CF can enhance the load signal by eliminating the specific spatial frequencies. It is divided into low-pass filtering, band-pass filtering, and high-pass filtering according to the enhancement types (low, medium and, high frequency). CF method can efficiently realize the simultaneous filtering of the dynamic load signals in different frequency bands, and its filtering results are uncompressed in each frequency band, which can avoid the error caused by frequency segmentation [57].

From mathematical, the CF method is defined as

$$y_m = \sum_{\lambda=-n}^{n} P_\lambda x_{m-\lambda}, \tag{3.3}$$

where $2n + 1$ is the maximum value of the CF weight coefficient function. P_λ is CF weight coefficient. The ideal frequency response function of the band-pass filter is as follow:

$$H(f) = \begin{cases} 1, \& f_1 \le f \le f_2 \\ 0, \& else \end{cases}, \tag{3.4}$$

CF weight coefficient is described as

$$p_\lambda = \frac{sin(\pi\beta\lambda)}{\pi\beta\lambda} - \frac{sin(\pi\alpha\lambda)}{\pi\alpha\lambda}, \tag{3.5}$$

$$p_0 = \beta - \alpha, \tag{3.6}$$

where, $\beta = 2f_2/SF$, $\alpha = 2f_1/SF$, and SF is the sample frequency. From the above formulas, we can see that the frequency truncation errors can be avoided when both sides of x_m exist m sampling points. The value of m will affect the accuracy and the speed of filtering. From the formula Eq. (3.5), the weights are inversely proportional to λ, which shows the weights far from the center point are negligible.

3.3. MCTS-CI model

To improve the accuracy of the online combination load identification, we propose a novel algorithm architecture RL-based, namely MCTS-CI. MCTS-CI includes a DNN structure based on an attention mechanism, namely ATM-DNN, and a self-learning module based on MCTS. MCTS-CI utilizes the ATM-DNN model to capture the combined load features information in the online scenario, which can focus on the specific features in the combined load while ignoring the unimportant ones. ATM-DNN can realize the fine-grained feature extraction of the load frequency domain and enhance the discrimination of the combined loads. After this operation, the RL agent can acquire the best strategies based on the experience gathered during the different training stages. Finally, this model generates actions La_t and adjust its parameters $\{Ls_t, La_t, \varphi\}$, which is based on the feedback from the environment. The process of the agent interacting with the environment can be seen in Figure 4. The greatest advantage of MCTS-CI is that it can help the RL agents understand the more complex environments and map their states to the actions [60, 61].

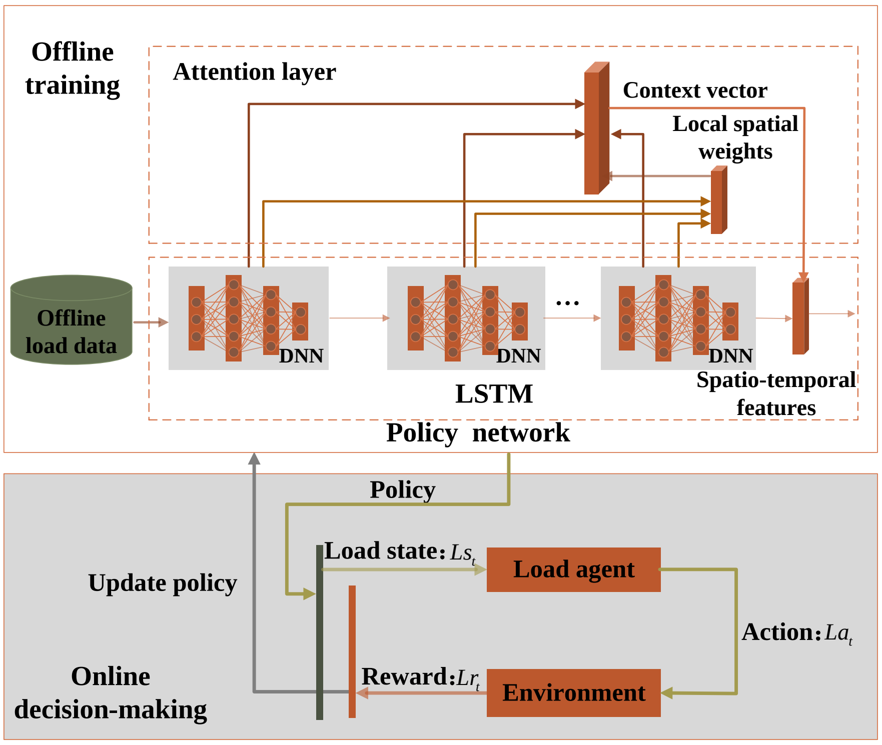

Figure 4. The flowchart of the agent interacts with the environment in RL-MCTS.

In this paper, we propose MCTS-CI model M_s, the initial strategy π_s, and SP is the all sequence of the current state Ls_t : $\{Ls_t, La_t^k, Lr_{t+1}^k, Ls_{t+1}^k, La_{t+1}^k, \cdots, Ls_T^k\}_{k=1}^K \backsim M_s, \pi_s$, where, La_t^k represent the current action, Lr_{t+1}^k represent the reward, K is the number of iteration. First, MCTS is constructed based on the sampling results, and then to compute the action value function $Q(Ls_t, La_s)$, where $La_s \in A$, A is the action set. The calculation expression of the action value function and its corresponding action is shown as follows:

$$Q(Ls_t, a_s) = \frac{1}{N(Ls_t, a_s)} \sum_{k=1}^{K} \sum_{u=t}^{T} (S_{uk} = Ls_t, A_{uk} = a_s) G_u, \tag{3.7}$$

$$La_t = argmaxQ(LS_t, a_s), \tag{3.8}$$

where, $Q(Ls_t, a_s)$ is the state-action and G_u represents its gains value in all complete sequences SP, γ represents the attenuation factor. G_u is described as

$$G_u = Lr_{u+1} + \gamma^1 Lr_{u+2} + \gamma^2 Lr_{u+3} + \cdots + \gamma^{T-u-1} Lr_T. \tag{3.9}$$

MCTS-CI is a model-independent method and it can prevent the solution of the dynamic programming from being too complicated. MCTS-CI is often used to find the best decision in a given domain by randomly sampling from the decision space and reconstructing a search tree according to the results. We employ the MCTS algorithm to realize the optimum search action for identifying the combination status of the online load [62]. The executing processes: according to the maximum probability value p_s^t and the reward v_s^t of the current node, which as the judgment basis of the optimal action a_s, the MCTS-CI determines the next execution action and state; the method is called each time of the search action for iteration until the threshold time or the preset number of cycles; from the starting, MCTS uses the initialization parameters $\varphi_\lambda(p_s^t, v_s^t)$, to perform the tree search, and then when the end of this search, the updated parameters $\varphi_\lambda^*(\pi_s^t, v_s^t)$ of MCTS is outputted, where z_s^t indicates that the action with the highest probability of action a_s. This process can be described as $MCTS_\varphi\{\varphi|(f_s^t, p_s^t, v_s^t)\} \rightarrow MCTS_{\varphi^*}\{\varphi^*|(f_s^t, \pi_s^t, z_s^t)\}$, and the MCTS algorithm is presented in Algorithm 2. Algorithm 2 will be described later (§3.4).

In ATM-DNN, the adjustment parameter λ is to minimize the value of $z_s^t - v_s^t$, and to minimize the value of p_s^t and π_s^t, where p_s^t is the choosing probability of ATM-DNN and π_s^t is the search probability of MCTS for the next state. In RL-MCTS, the loss function L_loss_s includes the mean squared error, the cross-entropy loss, and attention loss (§3.4). The cross entropy loss is used for the load classification loss. The final loss function L_loss_s can be described as follows:

$$L\_loss_s = (z_s^t - v_s^t)^2 - \pi_s^T \log p_s^{\mathrm{t}} + \mathrm{c}||\lambda||^2 + \eta \mathrm{loss\_A}. \tag{3.10}$$

This expression includes the mean squared error, the cross entropy loss [51], and attention loss $loss\_A$ (§3.4). The meaning of the parameter c is to control the extent of the L2 weight regularization, which can effectively prevent over-fitting. η is the weighting coefficient. In each iteration of MCTS, there are four steps to build a search tree: selection, expansion, simulation, and backtracking [62]. The overall design idea of MCTS is shown in Figure 5. In MCTS, each state node Ls contains four parameters: W_s, N_s, Q_s, and P_s. W_s is the sum of values in the next state; N_s is the number of times visited; Q_s is the average value of the next state; P_s indicates the prior probability of selected action.

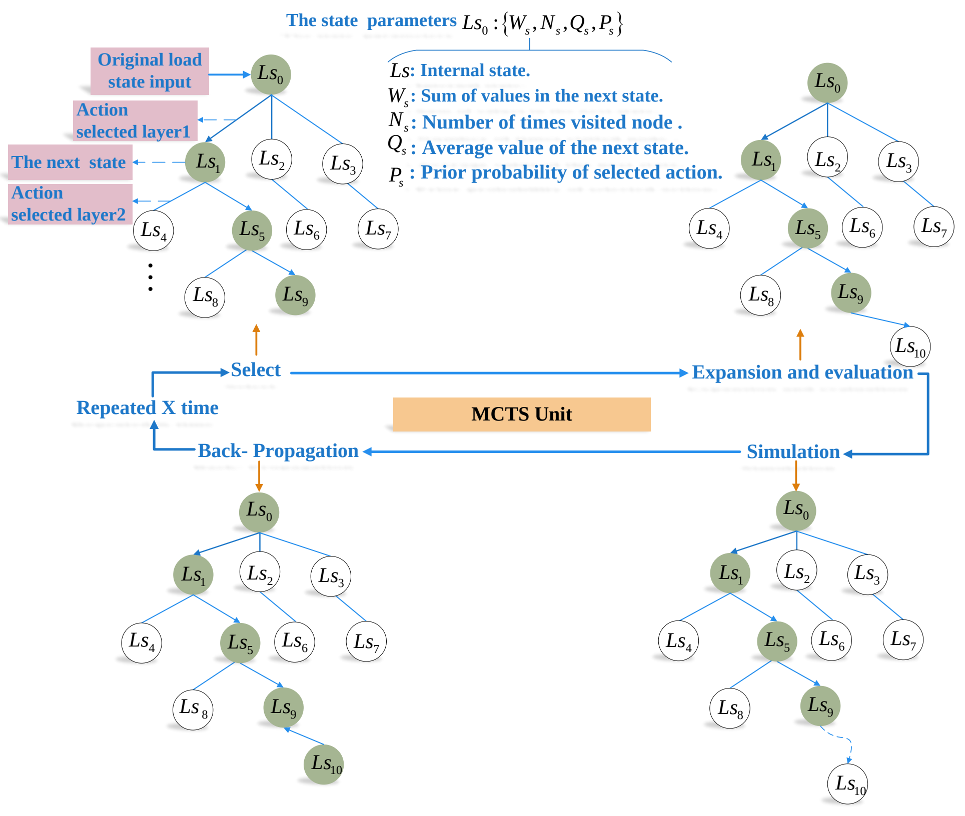
The state parameters $Ls_0:\{W_s,N_s,Q_s,P_s\}$
Ls: Internal state.
W_s: Sum of values in the next state.
N_s: Number of times visited node .
Q_s: Average value of the next state.
P_s: Prior probability of selected action.
Original load state input
Action selected layer1
The next state
Action selected layer2
Select
Expansion and evaluation
Repeated X time
MCTS Unit
Back- Propagation
Simulation

Figure 5. The overall design framework of MCTS in our model.

Step 1: selection. At time t the load power features based on the frequency-domain $F_s^T = \sum_{t=1}^{N} f_s^t$ are used as the current state of the root node LS_0^t, and $LS_0^t = \{W_0^t(f_s^t, a_s), N_0^t(f_s^t, a_s), P_0^t(f_s^t, a_s), Q_0^t(f_s^t, a_s)\}$, where $W_0^t(f_s^t, a_s)$ represents the total value of the next state, $N_0^t(f_s^t, a_s)$ represents the number of the visit count, $P_0^t(f_s^t, a_s)$ represents the prior probability of the selected action a_s, and $Q_0^t(f_s^t, a_s)$ represents the mean of the next state LS_0^t. In our paper, Upper Confidence Bound Applied to Trees (UCT) is employed to select the node with the highest score until a node that un-extended child nodes. UCT is a function and can obtain evaluation values at different depths [63]. UCT algorithm is a special Monte Carlo search algorithm, in mathematics, UCT is defined as follows:

$$UCT(s_m, s_0) = \frac{Q(s_m)}{N(s_m)} + c\sqrt{\frac{log(N(s_0))}{N(s_m)}}, \tag{3.11}$$

$$s_0(f_s^t, a_s) \rightarrow s_m, \tag{3.12}$$

$$a_s^* = \text{argmax} Q[(f_s^t, a_s) + U(f_s^t, a_s)], \tag{3.13}$$

where s_m is a child node of s_0, and we assume that M steps are required to reach the leaf node, and $m < M$. $\frac{Q(s_m)}{N(s_m)}$ indicates the estimated probability of the child node s_m; c is an explore parameters (the theoretically value is $\sqrt{2}$).

It is important to select the node with the maximum of $Q + U$ in the sub-branch of m, which is the next state until all the nodes have been traversed or the leaf node that does not reach the final MCTS. The detailed algorithm is described in Algorithm 1.

Step 2: expansion. At the end of the selection phase, the node s_m needs to be expanded, and its unexpanded action a_s is determined by using the tree selection strategy. Then a new node $s_m + 1$ is created as a new node in the search tree. It gives the flow diagram of algorithm of the steps 1 and 2 in Figure 6. The UCT aims to find the least visited node to explore each time of the tree search.

Algorithm 1: Detailed algorithm Tree_UCT

```
Input: The load state $LS_i$; The root node $RN\_s0$; The tree node $RN\_s$; The reward $re\_v$; Time
       threshold $TN$; Number of visited node $N_s$; The estimated value $Q_s$.
Output: Tree_UCT
//Create Tree_UCT sequence $LS_i$ and create the root node.
$Tree\_UCT\_sequence[LS\_i] = 0$;
$RN\_s0 = LS_0$;
//Within the calculated time limit.
for $tn = 1$ to $TN$ do
    while $RN\_s$ is not a leaf do
        if $RN\_s$ is an extendable node then
            if The action $a \in A(LS(RN\_s))$ is not selected in Action $A$ then
                //Adding the child node $RN\_s.child$. $RN\_s.child$=Best_Child($RN\_s$);
                $LS(RN\_s.child) = f(LS(RN\_s), a)$, $a(RN\_s.child) = a$;
            else
                //Select based on the upper bound index value of UCB information.
                $RN\_s.child \leftarrow$ Eq (3.11);
                $a \leftarrow$ Eq (3.13);
            while $LS$ is not a terminated state do
                Select action $a \in A(RN\_s.child)$;
                State $LS_0$ rewards;
                return State $LS_0$ rewards;
                while $RN\_s \neq NULL$ do
                    $N_s(RN\_s) \leftarrow (RN\_s) + 1$;
                    $Q_s(RN\_s) \leftarrow Q_s(RN\_s) + re\_v$;
                    $re\_v \leftarrow -re\_v$;
                    $RN\_s \leftarrow RN\_s.parent$;
```

Step 3: simulation. According to the original policy, from the extended position $s_m + 1$, the simulation is run with the new start node until the stop condition of the simulation is reached. The process will get an initial score for the new node $s_m + 1$, and the value is expressed as 1 or 0, which indicates true or false.

Step 4: back-propagation. The results of the simulation step are updated by the back-propagation operation, which is from the current iteration path to all the selected nodes. This process is that when the leaf nodes obtain the new observed return value V_s and the visit times of the parent node T_s through the simulation way, and then the UCT algorithm will update all internal node values on the search path by returning the results. This process can be described as

$$T_s^i = \sum_i T_s^i, \tag{3.14}$$

$$v_s = \frac{\sum_i v_s^i T_s^i}{T_s^i}, \tag{3.15}$$

where T_s indicates the visited times of the parent node and is the sum of the visits T_s^i of all the children nodes. The observed return value V_s is the weighted square of the observed return value V_s^i of all child nodes. The back-propagation starts from the leaf node and progresses up the search path to the root node. Each iteration will expand the search tree, and the depth of the tree will increase by the sum of iterations.

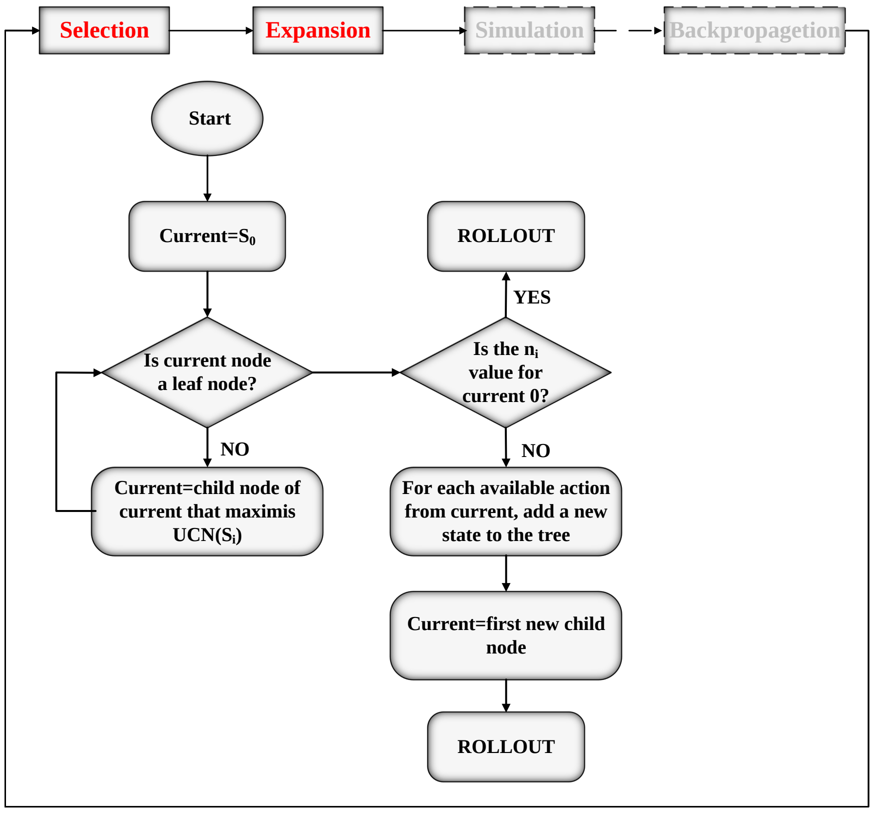
Selection
Expansion
Simulation
Backpropagetion
Start
Current=S0
ROLLOUT
YES
Is current node a leaf node?
Is the ni value for current 0?
NO
NO
Current=child node of current that maximis UCN(Si)
For each available action from current, add a new state to the tree
Current=first new child node
ROLLOUT

Figure 6. Flow diagram of algorithm in selection and expansion.

3.4. *DNN-based strategy network model*

In the paper, the strategy network model includes the internal policy and the external policy for building the MCTS-CI model. The internal strategy utilizes the UCT algorithm to optimize the performance of MCTS; the other one is a fusion DNN model with an attention mechanism, namely ATM-DNN, to replace the random strategy for achieving the sampling of the state sequence when the new state is not in the tree. The ATM-DNN model can better focus on the important feature of load. The overall structure of the model for the external policy can be shown in Figure 7.

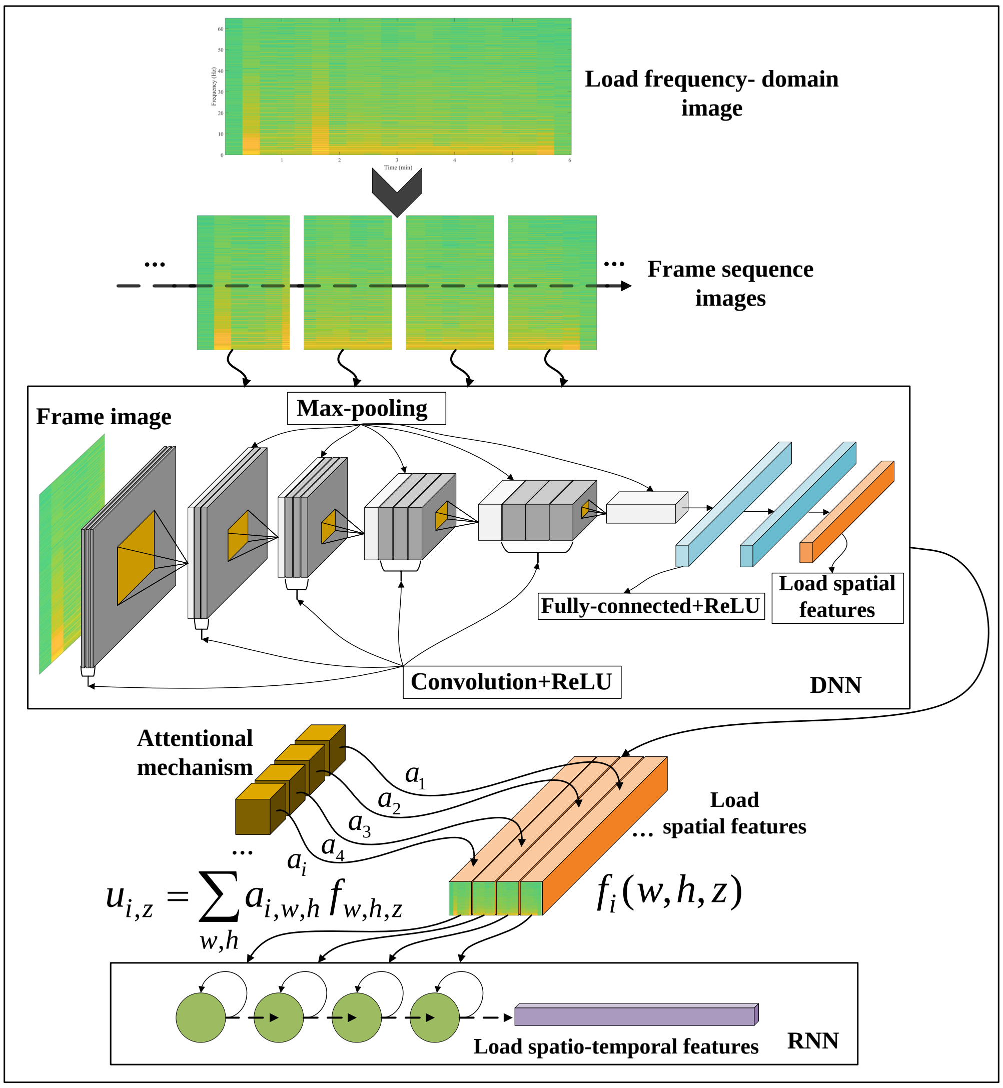

Figure 7. The overall algorithm structure of ATM-DNN.

From Figure 7, firstly, the time-frequency conversion technology is used to obtain the load frequency-domain image of the effective power. Secondly, the image segmentation technology based on the time series is used for the load frequency-domain image segmentation, which can form the load frame sequence. Then each frame image of the load frame sequence is the input of DNN for the load feature extraction. The attentional mechanism (ATM) is proposed to realize the fine-grained discrimination of the frequency-domain image in the salient regions of the load. In this process, ATM-DNN is presented as the extractor of the load feature to finish the load spatial feature map. In Figure 7, $f_i(w,h)$ represent the load spatial feature map, and i represent the i^{th} feature cube, and w,h

are used to find the specific eigenvalues. The sequence $a_i = \{a_1, a_2, a_3, \cdots, i \in N\}$ represent the attentional weights. The spatial feature of load can be presented as u_i, and its expression is as follow:

$$u_i = a_i \odot f_i = \sum_{w,h} a_{i,w,h} f_{i,w,h}. \tag{3.16}$$

Referencing the former studies for the image recognition with the attentional mechanism, in the mathematical, the calculation process of a_i includes the following formulas:

$$Sim_{a_{i,w,h}} = V_a^I tanh(W_l l_i + W_f f_{w,h,:}), \tag{3.17}$$

where l_i denotes the state of the hidden layer in RNN at time I, V_a^I represents the vector, $W_f f_{w,h,:}$ is the product of the load space features and weights, $W_l l_i$ is the dynamic weight, which can highlight the focusing performance of the load frame image.

And then $Sim_{a_{i,w,h}}$ is normalized, the expression is as follow:

$$a_i = softmax_{w,h}(a_i) = \frac{e^{a_i}}{\sum_{j=1} Z e^{a_j}}, \tag{3.18}$$

where Z is the number of the feature map, the introduction of $softmax$ can give more weight to important elements. The input and output of RNN are as follow:

$$R_{x_i} = W_z Z_{i-1} + W_{u_1} u_{(i-1)}, \tag{3.19}$$

$$(o_i, l_i) = G\_rnn(R_{x_i}, l_{i-1}), \tag{3.20}$$

$$O_i^{'} = softmax(W_0 O_i + W_{u_2} u_i). \tag{3.21}$$

The attention scoring criteria are used to measure the performance of the ATM-DNN model, and the attention loss function is described as

$$loss\_A = \sqrt{\sum_p \sum_m \sum_{wh} (1 - \|xs_{m,wh}^p - softmax(ys^q)_{m,wh}\|_{p,m,wh}^2)}, \tag{3.22}$$

where xs^p indicates the scores of the attention in the spatio-temporal dimension $M \times W \times H$; ys^q indicates the score of the q^{th} dimension, and $m \in [1, \ldots, M]$, $wh \in [1, \ldots, WH]$.

The overall design of MCTS-CI algorithm for NILMI is presented in Algorithm 2.

Algorithm 2: MCTS-CI Algorithm

```
Input: Load power series $[P_1, P_2, \ldots, P_{(t-1)}]$; Preprocessed data series $[P'_1, P'_2, \ldots, P'_{(t-1)}]$;
       Time threshold $Tleft$; Create the root node $RS\_node_0$ of the state $Ls_0$; Initial state $Ls_0$
. Output: The RL-MCTS model RM.
//Training RL-MCTS model.
//Construct MCTSsearch $Ls_0$.
if $t \leq Tleft$ then
    $RS\_node_0 = Ls_0$; //The root node $RS\_node_0$.
    if $R\_node_0$ is an expansile node. then
        //Node selection.
        $max\_UCT \leftarrow$ Eq (3.11); //$max\_UCT$ is the tree policy.
        $s_m \leftarrow$ Eq (3.12);
        $a_s^* \leftarrow$ Eq (3.13);
        for $r = RS\_node_0, \ldots, RS\_node.children$ do
            if $RS\_node.UCT > max\_UCT$ then
                $max\_UCT = RS\_node.UCT$;
                $RS\_node.sel = RS\_node.new$; //update node sequence.
        return $RS\_node.sel$;
        while $node.children.expanNode$ do
            $RS\_node.expanNode = node.rand\_children$; //Expansion and random Generate
              Children.
            //Simulation
            $RS\_node = new\_node.copynode$;
            while $!Terminate$ do
                $RS\_code = node.rand\_children$;
                //Backpropagation
                while $new.node.parent!=null$ do
                    $n\_node.n+ = 1$;
                    $n\_node.Q+ = result$;
                    $n\_node = n\_node.parent$;
//Training the external policy network: ATM-DNN.
$u_i \leftarrow$ Eq (3.16);
$O'_i \leftarrow$ Eq (3.21);
//RL processing.
$Q(Ls_t, a_s) \leftarrow$ Eq (3.7); //Action value.
$La_t \leftarrow$ Eq (3.8); // Action.
$L\_loss_s \leftarrow$ Eq (3.10);
return Select the action $a$ corresponding to the best child node (RS_node);
```

4. Our experiment

In this section, we mainly present the specific experimental settings for our proposed model. It includes the acquisition of the experimental data and data preprocessing, the experimental evaluation, and then the results discussion. The following is the purpose of our experiment: 1) the performance verification of the representational capacity of the ATM-DNN model; 2) the performance evaluation of the MCTS-CI model for the online combination load identification.

4.1. Datasets

The experimental datasets consist of two parts: the offline datasets and the online datasets. We use the offline datasets to train the external policy network ATM-DNN, and the online datasets are used to identify and predict the combined load. The experimental datasets mainly sample the high_frequency steady-state load power signals, from the six different houses. The data collection period is three years (2018/05/01-2020/04/30). The sampling period of the load from household appliances is 20 ns. In each dataset, there are four typical household appliances: refrigerator (REF), micro-wave oven (MWO), induction cooker (IC), and water heater (WH). The specific quantity of electrical equipment is shown in Table 1.

Table 1. The number of the sampling load datasets in the different house.

Load type	No. 1 house	No. 2 house	No. 3 house	No. 4 house	No. 5 house	No. 6 house
REF	10^7	10^6	10^7	10^7	10^7	10^7
MWO	10^6	10^6	10^7	10^5	10^7	10^6
IC	10^5	10^5	10^7	10^5	10^5	10^5
WH	10^7	10^7	10^7	10^6	10^7	10^6

4.2. Baseline models

We select the most representative models as the baseline models to verify the stability and robustness of MCTS-CI. These models include the Factorial Hidden Markov model (FHMM) [65], KNN [66], CNN [67], and the bi-directional long short-term memory neural network (Bi-LSTM) [68].

FHMM: This method is a multi-chain model extension structure. It is widely used as a statistical model to deal with time series and state series problems. The method divides the state space into many layers, and each layer is equivalent to a complete Hidden Markov model. At any given moment, the observed probability depends on the current state of all layers. The method can overcome the overfitting of the model training with the increased sampling space and the incomplete classification of a model in the small sampling space.

kNN: This method determines the classification of the sampling data according to the category of the nearest one or several samples. The improved the kNN models generally include adding weight coefficients and using the averages distance k-nearest neighbor algorithm to realize the estimate of the multi-local mean vectors in the specific patterns of each group. In the combined load identification scenario, the kNN method is more suitable.

CNN: This method mainly realizes the load features of that local are extracted by building the connection of the input of each neuron and the local acceptance domain of the previous layer. It can achieve the network parallel operation. By integrating the DNN architecture, the segmentation of the input features is formed in load signals to achieve the load event, classification, and sample regression in NILM.

Bi-LSTM: This method is used to solve the bidirectional time dependence of the load signal. LSTM realizes time memory function through the cell door switch and prevents gradient from disappearing.

4.3. Experimental environment and parameter settings

To meet our experimental requirements, our experimental environment includes the hardware and software. The Pytorch 3.7 is used to achieve the design and implementation of the algorithms; the desktop PC with Windows 10 system and a NVIDA GeForce RTX 3080 Ti GPU, 128GB of RAM, and CUDA v10.2 are employed.

The core of our proposed model includes the ATM-DNN model and the MCTS algorithm. In ATM-DNN, the specific parameters set as follows: there are six blocks in the hidden layer, six filters in each block, the size and stride of the pooling layer are two; our activation function is ReLu; the dropout probability set at 0.45; the number of mini-batch is 512; Adam is as the optimizer; the learning rate set to 0.0006; the number of epochs set to 300. In the MCTS model, the self-play process will last for 20 h, implemented 10^5 decision-makings, and 750 simulation operations are performed in each MCTS, which will present in the following subparagraph.

4.4. Experimental data preparation and preprocessing

We select four representative household appliances for our experiment. Their work patterns range from simple to complex, especially the bidirectional dependence of the device can be reflected. Relatively diverse working modes are beneficial to verifying the performance indicators of our model. We sample the experimental datasets from the smart meter installed at the power entrance to the house, which can realize the acquisition and practical storage of the real-time data (current, voltage, and efficient power) in the various household appliances. The steady-state power signal of load in high-frequency contains the rich load characteristic information. The information can provide the discriminative load features for the combined load identification in various online scenarios.

The CF-based signal enhancement method is used to realize the data preprocessing. In the load data sampling, signal noises are caused by equipment operation errors, the overshoot, or abnormal peak of voltage or current of the signal. The data preprocessing method can solve the signal acquisition error and instability problems effectively; this method can correct the abnormal power values and ensure the top quality of the signal.

We have prepared a lot of sampled datasets to validate the performance of our model, and these datasets are from four houses (No. 1 house, No. 2 house, No. 3 house, and No. 4 house). We divide each dataset into three parts: the training dataset, the validation dataset, and the testing dataset. The first 70% of the dataset sequence for the training model, the subsequent dataset sequence 15% for validating the model, and the rest 15% of the data sets for the testing model. In the online experimental stage, we use the collected data sets of No. 5 house and No. 6 house in real-time to further verify the model performance's competitiveness.

4.5. Experimental evaluation criteria

We employed the most common evaluation criteria to verify the performance and stability of our model, such as *Micro-*F_1, *Macro-*F_1, Hamming loss (H-loss), and the mean absolute error (MAE). These methods have been widely used in the field of deep learning. The criteria can be defined as follow:

Micro-F_1 and *Macro-*F_1. *Micro-*F_1 and *Macro-*F_1 are the two strategies for calculating F_1*-score* of the multi-classification. It is assumed that LTP_i^t, LFP_i^t, LTN_i^t, and LFN_i^t denote the true positive at time t, false positive, true negative, false negative of classify i. The precision, $Lpre_i^t$, and the recall, $Lrec_i^t$, of the class i can be described as

$$Lpre_i^t = \frac{LTP_i^t}{LTP_i^t + LFP_i^t}, \tag{4.1}$$

$$Lrec_i^t = \frac{LTP_i^t}{LTP_i^t + LFN_i^t}. \tag{4.2}$$

So, the total precision and recall of all categories are presented as follows:

$$Lpre_{(i)micro}^t = \frac{\sum_{i=1}^n LTP_i^t}{\sum_{i=1}^n LTP_i^t + \sum_{i=1}^n LFP_i^t}, \tag{4.3}$$

$$Lrec_{(i)micro}^t = \frac{\sum_{i=1}^n LTP_i^t}{\sum_{i=1}^n LTP_i^t + \sum_{i=1}^n LFN_i^t}, \tag{4.4}$$

$$micro - F_1 = 2 \cdot \frac{Lpre_{(i)micro}^t \cdot Lrec_{micro}^t}{Lpre_{(i)micro}^t + Lrec_{(i)micro}^t}, \tag{4.5}$$

Macro-F_1 is described as follows:

$$Macro - F_1 = 2 \cdot \frac{\overline{Lpre_i} \cdot \overline{Lrec_i}}{\overline{Lpre_i} + \overline{Lrec_i}}, \tag{4.6}$$

$\overline{Lpre_i}$ indicates as the mean precision of the i – th class, $\overline{Lrec_i}$ indicates as the mean recall of the i – th class.

Hamming-loss: $Hamming - loss$ is employed to measure the times in the misclassification samples; namely the labels belonging to a sample are not predicted, but the labels that do not belong to the sample are predicted to belong to the sample. The smaller value, the better the model performance. The mathematical formula is described as

$$H - loss(x_i, y_i) = \frac{1}{|M|} \sum_{i=1} |M| \frac{1}{|L|} xor(x_i, y_i), \tag{4.7}$$

where $|M|$ represents the size of samples, $|L|$ represents the size of labels, x_i and y_i indicate the predicted value and real value, respectively.

MAE: we suppose that y_i^t represents the real data, $y_i^{'t}$ is the estimated data of the appliance i at time t. The mathematical expression is as follows:

$$MAE_i = \frac{1}{T}\sum_{t=1}^{T}|y_i^t - y_i'^t|. \tag{4.8}$$

4.6. Experimental results and discussion

In this section, the model's learning ability and overall accuracy are verified, respectively. The various performance scores of the models are presented to evaluate the superiority of our model.

To improve the MCTS model's decision-making efficiency to ensure the decision-making process's reliability, we utilize the ATM-DNN structure to obtain the initial tree policy p_s^t and the action values v_s^t, which instead of the randomly selected actions at the beginning of the program. The next time, the program of MCTS will self-learning for 20 h without any additional intervention, and then the complete search process will be completed. The outputted parameters of each iteration in MCTS are compared with the output results of ATM-DNN and updated ATM-DNN parameters as the policy improvement. We use the performance index MSE and $Loss$ as the critical evaluation indicators to present the performance of the search process in model training. Then this convergence trend can be seen in Figure 8. MSE indicates the degree of error between the actual output of MCTS and the predicted value of ATM-DNN, which can be used as the basis to adjust the parameters of the ATM-DNN and minimize the predicted value. $Loss$ is the crossentropy losses, and the values represent the similarity of ATM-DNN selection strategy p_s^t to the search probabilities π_s.

The training process of MCTS-CI includes two steps: the offline datasets of four different houses are used to train the ATM-DNN model, and then the results of the model training are the initial strategies for completing the initial selection in MCTS; the MCTS starts the search by self-learning and goes through exploration, simulation and, feedback. The process lasts about 20 h and implements about 10^5 decision-makings. In each MCTS, 750 simulation operations are performed, meaning that making the new decisions is nearly 0.5 s. After about 7000 self-decisions, the ATM-DNN can efficiently guide the MCTS search to select stable actions. Finally, the process can realize the policy evaluation and improvement in the MCTS search.

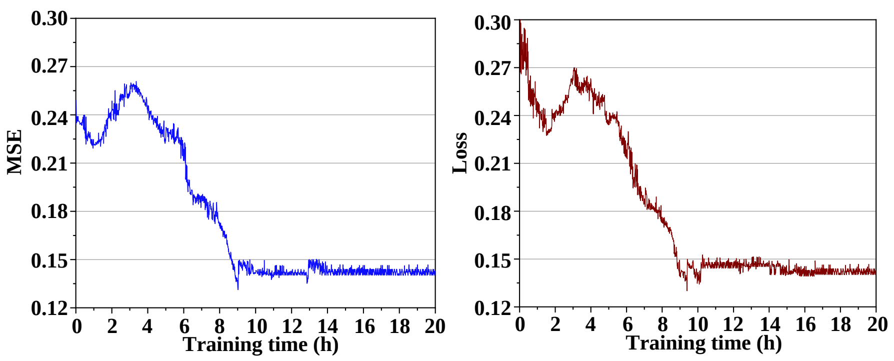

Figure 8. The changing curve of the MSE and $Loss$ on the training process.

We use the mean reward of RL in the training process to further validate the performance of the MCTS-CI. The results are shown in Figure 9. We employ the two datasets of No. 1 house and No.

2 house, respectively. The two datasets provide a large amount of the sampled load data resources in four household appliances. The validation results are shown in Figure 9. From the two curves, the intelligent agent goes through the initial selection and exploration. Then they eventually converge to the global optimum, which can prove the convergence of our method. From convergence result, the average reward is approximately 0.697 and 0.676, respectively, on the two datasets.

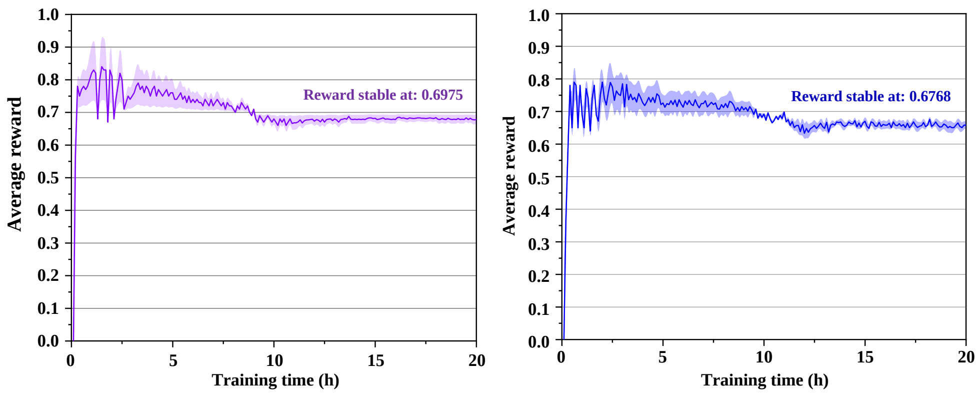

Figure 9. Mean reward of RL in the training process of model on the datasets from No. 1 house and No. 2 house.

To explain the influence of the simulation times on model performance, we try to use the different simulation times to experiment with our model; the result of the experiment can be presented in Figure 10. We performed 50, 150, 350, 750 simulation cycles, respectively, and as the number of cycles increases, so does the rate of reward. It is because the increase of the simulation numbers can make more simulation numbers of the Monte Carlo tree, which can improve the accuracy of the evaluation of the action. So, it can ensure the optimal action and improve the effectiveness of the Monte Carlo tree search. From Figure 10, as the reward reaches a certain point, the performance will tend to stabilize.

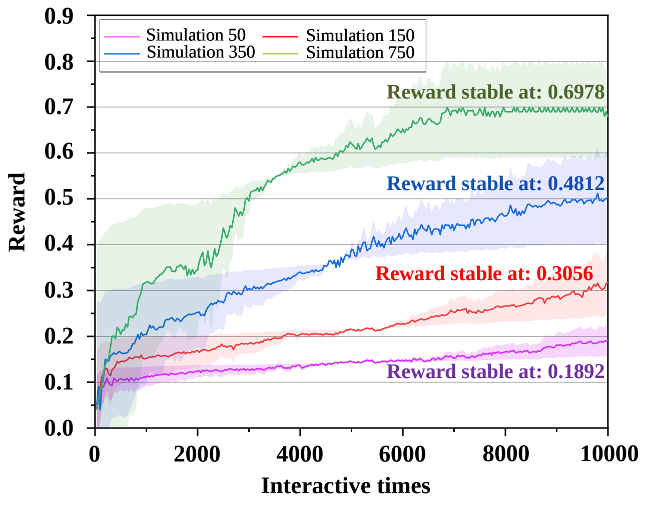

Figure 10. Reward changes during the training of the models.

In the model training process, we use the training time to test the computing power of the model, and the comparison of the training time in five models is shown in Figure 11. The sample size and the performances of the computing devices are the main factors in computing time. From Figure 11, our proposed model has the shortest model training time on the different sizes of the sample training sets.

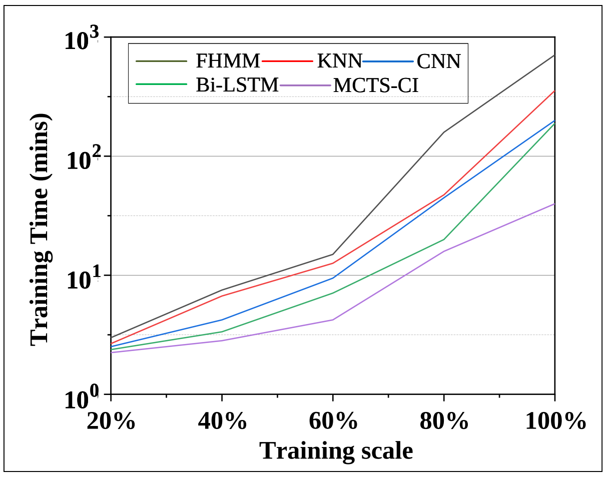

Figure 11. Comparison of the training time in five models.

4.7. Validation of the model accuracy and performance

In this subsection, four datasets from the different houses are employed to verify the performance of FHMM, KNN, CNN, Bi-LSTM, and MCTS-CI. The validation process includes the following: 1) the comparison of the average accuracy with the five methods in training datasets; 2) the overall performance score of five different methods; 3) the comparison of the $Micro-F_1$ and $Macro-F_1$ scores on the specific household appliances datasets. We first present the overall accuracy of benchmark models and our model. The five-fold cross-validation on each dataset is used to ensure the reliability of our experiment results. Table 2 shows the accuracy and standard deviations of the results, which are presented in bold for the best results.

Table 2. Average accuracy for load online identification.

Methods	Datasets			
	No. 1 house	No. 2 house	No. 3 house	No. 4 house
FHMM	67.13 ± 0.36	69.83 ± 1.71	54.56 ± 0.98	63.68 ± 4.32
KNN	70.34 ± 1.23	73.56 ± 0.89	62.78 ± 0.65	74.69 ± 2.31
CNN	78.24 ± 1.12	75.43 ± 0.62	77.23 ± 4.12	80.68 ± 0.89
Bi-LSTM	87.78 ± 4.13	79.56 ± 2.92	85.57 ± 2.54	85.97 ± 5.18
MCTS-CI	95.56 ± 0.19	89.91 ± 0.47	90.95 ± 0.42	92.22 ± 0.34

From Table 2, the accuracy of MCTS-CI is higher than other methods on different datasets. Significantly, the average accuracy of the MCTS-CI reaches 95.56%, which is higher than the second-ranked method 7.78% on No. 1 house datasets. On the other datasets, the average accuracy of MCTS-CI is 89.91%, 90.95%, and 92.22%, which are superior to others 10.35%, 5.38%, and 6.25%, respectively. In comparison with CNN, in load combination identification, MCTS-CI is more competitive. The reason that different from image recognition, the load has the time-series feature, and CNN does not play to its strengths. It can illustrate that our model has advantages in online identification of the loads. In a word, this method will have a good application prospect for NILMI.

To go a step further and validate the performance of our model, we carry out the experimental tests from the overall and detailed aspects of the model, respectively. The scores from *Micro-*F_1, *Macro-*F_1, and $H - loss$ are employed to describe the experimental results.

The overall case: the experimental results of four datasets from the different houses are shown and discussed. Table 3 demonstrates the overall performance indicators of the benchmark models and our proposed model in different datasets from four houses, respectively. The numbers in the bold represent the performance is better than the others. Especially, from Table 3 *Micro-*F_1 score of our method can come up to 0.964, which is higher than the second-ranked model, 0.107. In addition, the score is 0.947, 0.946, 0.935 are higher than the second-ranked model Bi-LSTM 0.084, 0.092, 0.058, respectively. The same effect of model performance appears on *Macro-*F_1 score. From H-loss, the value of MCTS-CI is relatively minimal, indicating that the number of misclassification samples is low. The second-lowest is Bi-LSTM, indicating that the LSTM model has a significant advantage for load sampling data based on time series characteristics. CNN model can complete the feature extraction of load well, but it is not sensitive to the load combination identification. From MAE, the values of MCTS-CI are smaller than the other benchmark models, and it presents a high identification accuracy in different datasets.

Table 3. Overall performance score on four different datasets from different houses.

Criteria	Methods	Datasets			
		No. 1 house	No. 2 house	No. 3 house	No. 4 house
$H - loss$	FHMM	0.021 ± 0.002	0.019 ± 0.001	0.017 ± 0.002	0.020 ± 0.001
	KNN	0.018 ± 0.001	0.016 ± 0.002	0.014 ± 0.001	0.015 ± 0.004
	CNN	0.014 ± 0.003	0.012 ± 0.004	0.011 ± 0.003	0.013 ± 0.002
	Bi-LSTM	0.009 ± 0.002	0.008 ± 0.001	0.006 ± 0.001	0.009 ± 0.003
	MCTS-CI	0.003 ± 0.001	0.003 ± 0.001	0.002 ± 0.001	0.004 ± 0.002
Micro-F_1	FHMM	0.691 ± 0.019	0.636 ± 0.012	0.627 ± 0.021	0.624 ± 0.017
	KNN	0.724 ± 0.003	0.652 ± 0.014	0.719 ± 0.016	0.698 ± 0.030
	CNN	0.821 ± 0.013	0.794 ± 0.017	0.814 ± 0.008	0.755 ± 0.023
	Bi-LSTM	0.857 ± 0.019	0.873 ± 0.040	0.854 ± 0.013	0.877 ± 0.017
	MCTS-CI	0.964 ± 0.004	0.947 ± 0.008	0.946 ± 0.011	0.935 ± 0.006
Macro-F_1	FHMM	0.562 ± 0.029	0.637 ± 0.010	0.587 ± 0.014	0.613 ± 0.011
	KNN	0.695 ± 0.019	0.642 ± 0.013	0.693 ± 0.012	0.661 ± 0.008
	CNN	0.795 ± 0.015	0.788 ± 0.015	0.758 ± 0.005	0.714 ± 0.003
	Bi-LSTM	0.828 ± 0.012	0.854 ± 0.010	0.831 ± 0.004	0.856 ± 0.012
	MCTS-CI	0.924 ± 0.008	0.937 ± 0.012	0.912 ± 0.010	0.928 ± 0.014
MAE	FHMM	0.321 ± 0.018	0.351 ± 0.015	0.371 ± 0.002	0.251 ± 0.008
	KNN	0.275 ± 0.011	0.247 ± 0.012	0.284 ± 0.003	0.212 ± 0.013
	CNN	0.172 ± 0.009	0.201 ± 0.022	0.171 ± 0.017	0.149 ± 0.008
	Bi-LSTM	0.154 ± 0.007	0.132 ± 0.011	0.154 ± 0.012	0.135 ± 0.014
	MCTS-CI	0.081 ± 0.013	0.062 ± 0.008	0.091 ± 0.012	0.061 ± 0.006

The detailed case: to better demonstrate our approach's significant advantages, we use our model and four benchmark models on the various appliances datasets from the four houses to verify the models performance. Finally, *Micro-F_1* and *Macro-F_1* scores of the various appliances used in the four houses are obtained, which can be found in Figures 12 and 13. From Figure 12, *Micro-F_1* scores of each appliance are shown from subgraph a to subgraph d; the scores of MCTS-CI in each of the appliances are the highest-scoring, and Bi-LSTM is the second-highest score, and then the FHMM is the lowest-scoring. The results similar to that presented in Figure 13, *Macro-F_1* scores of MCTS-CI are best. Our proposed method excels the other methods in the score of each appliance. Especially, in subgraph Figure 12(a), *Micro-F_1* score of the refrigerator, water heater, microwave oven, and induction cooker is up to 96.19%, 95.52%, 97.57%, and 94.37%, which is far superior to the other methods. In subgraph Figure 12(d), the highest score of *Micro-F_1* is 97.96%, and it is found in the water heater with our method; the worst score appears in subgraph Figure 13(d), *Macro-F_1*, 53.23%, which is the score with FHMM in the refrigerator. The performance of FHMM is not good in our experiment, and the same situation also appears in KNN and CNN models. The Bi-LSTM is better than CNN and KNN because some memory gates mode characterizes it, and it can better represent the time-dependent relationships of sampling data.

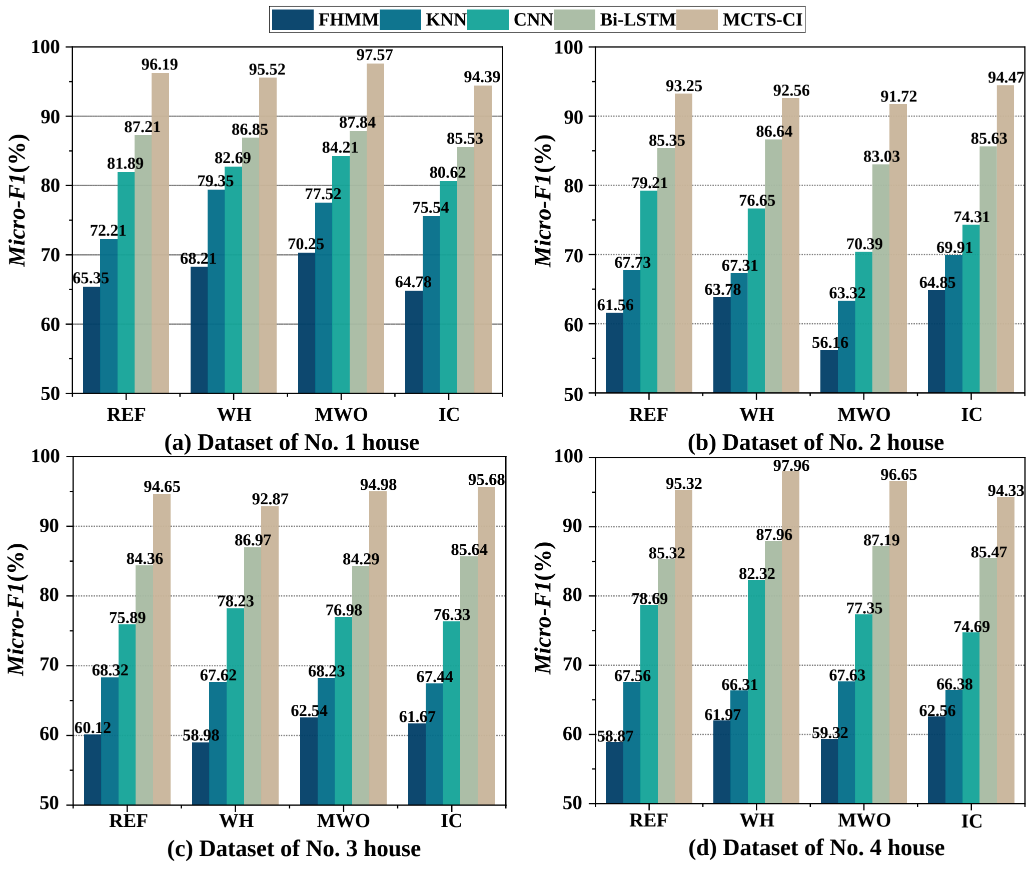

Figure 12. The *Micro-F_1* scores of the various appliances used in the four houses.

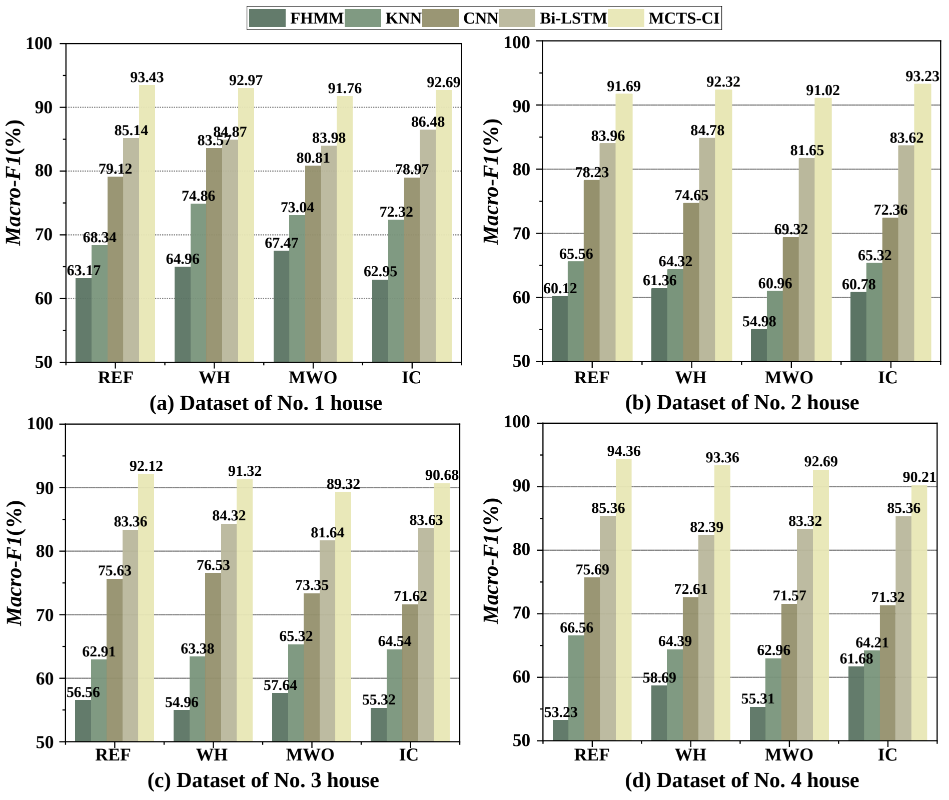

Figure 13. The *Macro-F_1* scores of the various appliances used in the four houses.

Actual load identification and prediction test: In this subsection, we collect the online data from No. 5 house and No. 6 house as the validation datasets to verify our model performance in real-world applications. We observe the electricity consumption of two households and sample the load power signals from various household appliances on three random days. Figure 14 shows the power curve of the time-domain from the two different houses in three days. From Figure 14, we can identify four periods of electricity consumption, which mainly include 5:30 a.m. to 7:30 a.m, 11:00 a.m. to 1:00 p.m., 5:30 p.m. to 8:00 p.m., and 9:00 p.m. to 10:30 p.m.

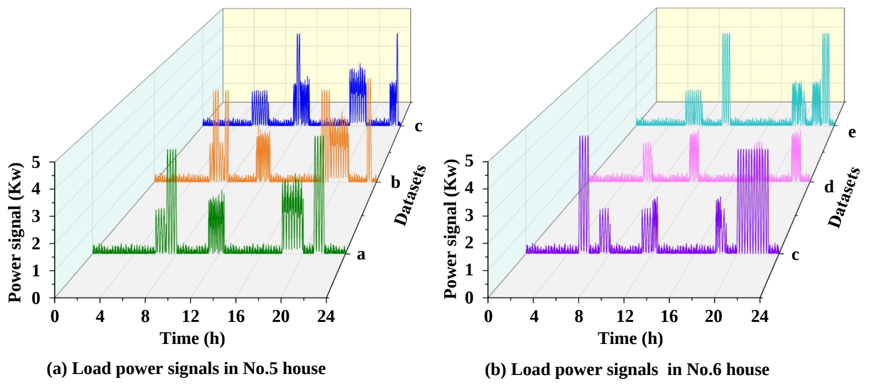

Figure 14. Power signal of different houses in validation datasets.

The frequency-domain image of the load power signals can also represent the load features, and they, as the input of DNN, can improve the accuracy of load feature extraction. In our model, the time-frequency conversion technology forms the load frequency-domain features. To ensure the validity of our data set, we use the CF method to improve the quality of the power signals and reduce the error of signal acquisition. After the data preparation and preprocessing, the datasets are inputted into our proposed model to extract the spatio-temporal features of load and identify the specific load power from all kinds of household appliances. The detailed load identification results can be presented in Figures 15 and 16.

As Figures 15 and 16 indicate, MCTS-CI shows a good recognition effect and stability in the test data set of the actual electricity consumption. Significantly, the experimental result of the water heater agrees with and refrigerator with the real sampled data in the usage scenarios of two different houses, which shows that MCTS-CI can achieve excellent recognition for the electrical appliance with high-frequency usage. The same results appear in the micro-wave oven of No. 5 house and the induction cooker of No. 6 house. For the household appliances with low-frequency usage, more data are needed to train the model. Our proposed method has outstanding practical value and better performance for correctly categorizing the online load.

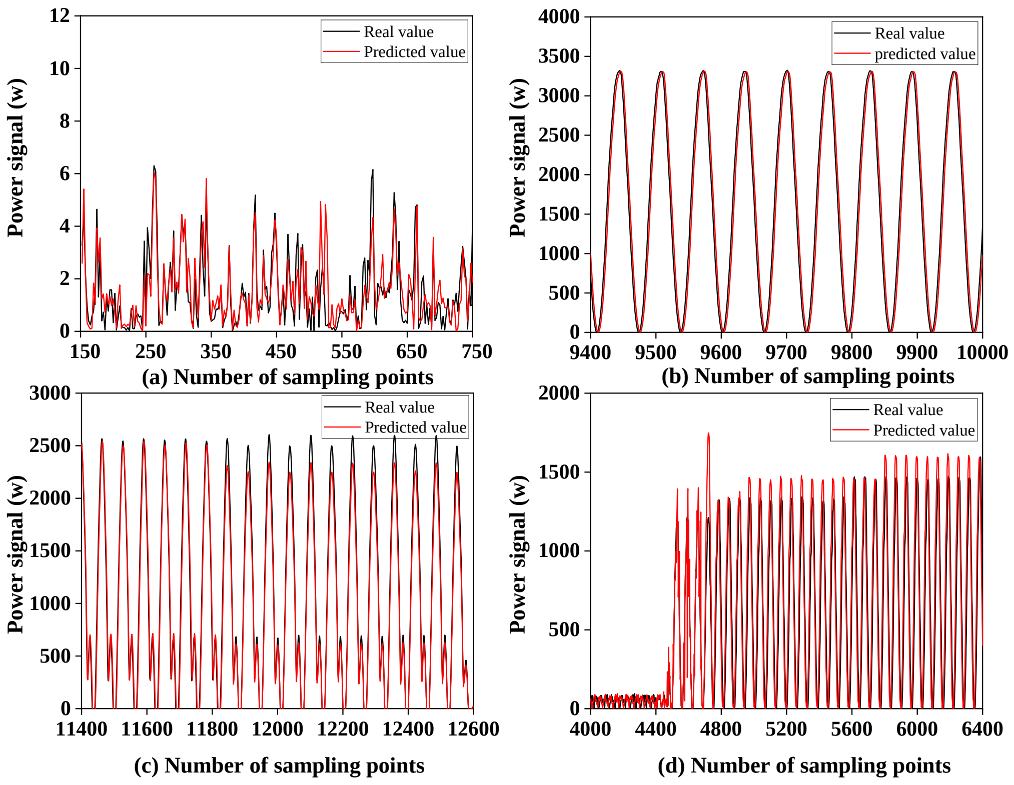

Figure 15. The predicting results of the specific load for the house No. 5 using our method. (a) The result for refrigerator; (b) the result for water heater; (c) the result for micro-wave oven; (d) the result for induction cooker.

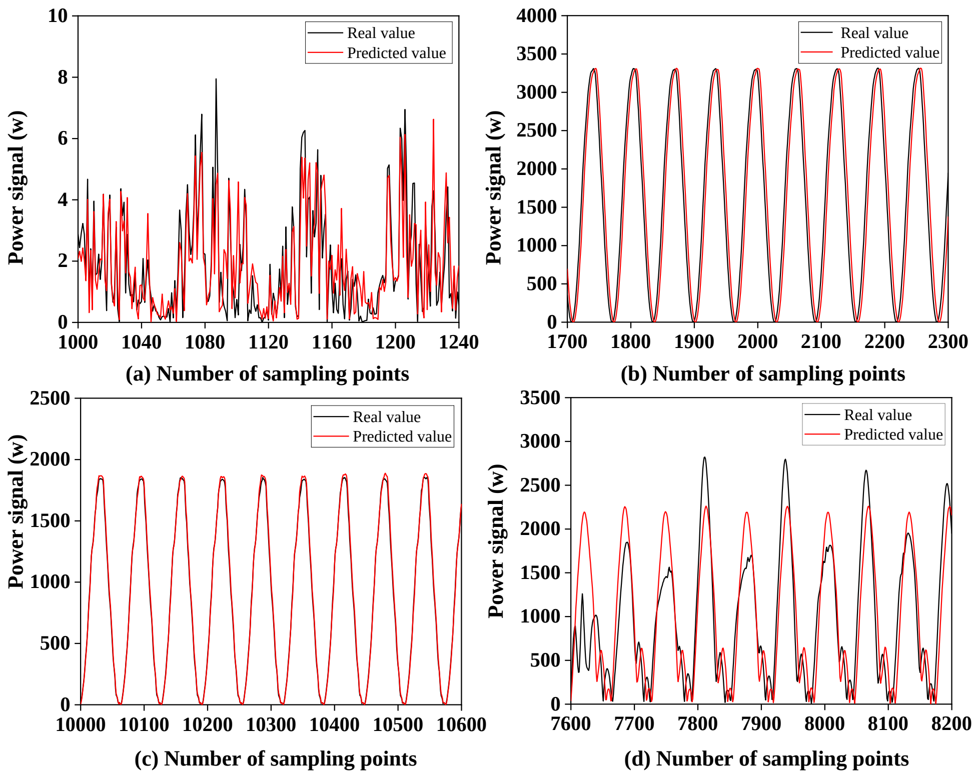

Figure 16. The predicting results of the specific load for the house No. 6 using our method. (a) The result for refrigerator; (b) the result for water heater; (c) the result for micro-wave oven; (d) the result for induction cooker.

5. Conclusions

CGC is applied in the IoT system of the smart grid for realizing the rational allocation of the various resources and improve the production efficiency. NILMI is the most efficient way, can improve the performance and stability of CGC in the smart grid. In this paper, we develop a novel algorithm framework, MCTS-CI, which is a multi-stage algorithm framework. MCTS-CI involves a load data sharing platform, LPF-KB, a load spatio-temporal feature extraction mechanism based on ATM-DNN, and a set of optimized policy networks based on MCTC. LPF-KB can enhance the utilization rate of load sampled data to realize the fine-grained storage of load; ATM-DNN can improve the distinctiveness and effectiveness of the combined load features; the extraction of load spatio-temporal features can improve the learning ability of load representations; MCTS is used to achieve the optimal strategy for reducing the error rate of load identification and energy prediction in the online scenario. By comparison with four benchmark models, the experimental results prove that MCTS-CI has outstanding performance for improving the accuracy of the load combination identification and reducing the identification error rate of the unknown load.

The method can be widely applied for predicting household electricity consumption in online scenarios. The method will as a design idea for realizing the short and medium-term load identification with the external weather and periodic temporal features to improve the intelligent adjustment performance of the IoT system in the smart grid. In the future, we will utilize MCTS-CI to address the more complex and variable load combination scenarios.

Acknowledgments

The authors of this paper were supported by S&T Program of Hebei through grant 20310101D, and National Key R&D Program of China through grant 2021YFB1714800.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. D. P. M. Abellana, A systemic analysis of green computing adoption using genetically evolved fuzzy cognitive map: A Philippine scenario, *Kybernetes*, **50** (2020), 2668–2696. https://doi.org/10.1108/K-05-2020-0263
2. W. M. Zheng, Q. W. Chai, J. hang, X. S. Xue, Ternary compound ontology matching for cognitive green computing, *Math. Biosci. Eng.*, **18** (2021), 4860–4870. https://doi.org/10.3934/mbe.2021247
3. X. Liu, Y. Li, X. Zhang, W. Lu, M. Xiong, Energy-efficient resource optimization in green cognitive internet of things, *Mobile Networks Appl.*, **25** (2020), 107750–107770. https://doi.org/10.1007/s11036-020-01510-w
4. Y. M. Jiang, M. S. Liu, H. Peng, M. Z. A. Bhuiyan, A reliable deep learning-based algorithm design for IoT load identification in smart grid, *Ad Hoc Networks*, **123** (2021), 102643–102673. https://doi.org/10.1016/j.adhoc.2021.102643
5. Y. Liu, L. Zhong, J. Qiu, J. Lu, W. Wang, Unsupervised domain adaptation for non-intrusive load monitoring via adversarial and joint adaptation network, *IEEE Trans. Ind. Inf.*, **18** (2021), 266–277. https://doi.org/10.1109/TII.2021.3065934
6. H. Peng, J. X. Li, S. Z. Wang, L. H. Wang, Q. R. Gong, R. Y. Yang, et al., Hierarchical taxonomy-aware and attentional graph capsule RCNNs for large-scale multi-label text classification, *IEEE Trans. Knowl. Data Eng.*, **33** (2021), 2505–2519 https://doi.org/10.1109/TKDE.2019.2959991
7. A. F. Moreno Jaramillo, D. M. Laverty, D. J. Morrow, J. Martinez del Rincon, A. M. Foley, Load modelling and non-intrusive load monitoring to integrate distributed energy resources in low and medium voltage networks, *Renewable Energy*, **179** (2021), 445–466. https://doi.org/10.1016/j.renene.2021.07.056
8. R. V. A. Monteiro, J. C. R. de Santana, R. F. S. Teixeira, A. S. Bretas, R. Aguiar, C. E. P. Poma, Non-intrusive load monitoring using artificial intelligence classifiers: Performance analysis of machine learning techniques, *Electr. Power Syst. Res.*, **198** (2021), 107347. https://doi.org/10.1016/j.epsr.2021.107347

9. H. X. Wang, J. S. Zhang, C. B. Lu, C. Y. Wu, Privacy preserving in non-intrusive load monitoring: A differential privacy perspective, *IEEE Trans. Smart Grid*, **12** (2020), 2529–2543. https://doi.org/10.1109/TSG.2020.3038757

10. D. Hua, F. Q. Huang, L. J. Wang, W. T. Chen, Simultaneous disaggregation of multiple appliances based on non-intrusive load monitoring, *Electr. Power Syst. Res.*, **193** (2021), 106887. https://doi.org/10.1016/j.epsr.2020.106887

11. M. Dincecco, S. Squartini, M. J. Zhong, Transfer learning for non-intrusive load monitoring, *IEEE Trans. Smart Grid*, **11** (2020), 1419-1429. https://doi.org/10.1109/TSG.2019.2938068

12. G. A. Raiker, R. B. Subba, U. Loganathan, S. Agrawal, A. S. Thakur, J. P. Barton, et al., Energy disaggregation using energy demand model and IoT based control, *IEEE Trans. Ind. Appl.*, **57** (2020), 1746–1754. https://doi.org/10.1109/TIA.2020.3047016

13. F. Ciancetta, G. Bucci, E. Fiorucci, S. Mari, A. Fioravanti, A new convolutional neural network-based system for NILM applications, *IEEE Trans. Instrum. Meas.*, **70** (2021), 1501112. https://doi.org/10.1109/TIM.2020.3035193

14. A. Faustine, L. Pereira, C. Klemenjak, Adaptive weighted recurrence graphs for appliance recognition in non-intrusive load monitoring, *IEEE Trans. Smart Grid*, **12** (2020), 398–406. https://doi.org/10.1109/TSG.2020.3010621

15. Y. Himeur, A. Alsalemi, F. Bensaali, A. Amira, An intelligent nonintrusive load monitoring scheme based on 2D phase encoding of power signals, *Int. J. Intell. Syst.*, **36** (2020), 72–93. https://doi.org/10.1002/int.22292

16. Y. Yang, J. Zhong, W. Li, T. A. Gulliver, S. Li, Semisupervised multilabel deep learning based nonintrusive load monitoring in smart grids, *IEEE Trans. Ind. Inf.*, **16** (2020), 6892–6902. https://doi.org/10.1109/TII.2019.2955470

17. M. Kaselimi, N. Doulamis, A. Voulodimos, E. Protopapadakis, A. Doulamis, Context aware energy disaggregation using adaptive bidirectional LSTM models, *IEEE Trans. Smart Grid*, **11** (2020), 3054–3067. https://doi.org/10.1109/TSG.2020.2974347

18. H. Chen, Y. H. Wang, C. H. Fan, A convolutional autoencoder-based approach with batch normalization for energy disaggregation, *J. Supercomput.* **11** (2021), 2961-2978. https://doi.org/10.1007/s11227-020-03375-y

19. P. Hao, J. Li, Y. He, Y. Liu, M. Bao, L. Wang, et al., Large-scale hierarchical text classification with recursively regularized deep graph-CNN, in *Web Conference 2018: Proceedings of the 2018 World Wide Web Conference*, Macao, Lyon, France, (2018), 1063–1072. https://doi.org/10.1145/3178876.3186005

20. H. Peng, J. Li, Q. Gong, Y. Ning, S. Wang, L. He, Motif-matching based subgraph-level attentional convolutional network for graph classification, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **34** (2021), 5387–5394. https://doi.org/10.1609/aaai.v34i04.5987

21. A. Moradzadeh, O. Sadeghian, K. Pourhossein, B. Mohammadiivatloo, A. Anvarimoghaddam, Improving residential load disaggregation for sustainable development of energy via principal component analysis, *Sustainability*, **12** (2020), 1–14. https://doi.org/10.3390/su12083158

22. P. Hao, J. Li, Y. Song, Y. Liu, Incrementally learning the hierarchical softmax function for neural language models, in *Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, CA, **31** (2017), 3267–3273. https://doi.org/10.1609/aaai.v31i1.10994

23. A. U. Rehman, T. T. Lie, B. Vallès, S. R. Tito, Event-detection algorithms for low sampling nonintrusive load monitoring systems based on low complexity statistical features, *IEEE Trans. Instrum. Meas.*, **69** (2019), 751–759. https://doi.org/10.1109/TIM.2019.2904351

24. M. S. Tsai, Y. H. Lin, Modern development of an adaptive non-intrusive appliance load monitoring system in electricity energy conservation, *Appl. Energy*, **96** (2012), 55–73. https://doi.org/10.1016/j.apenergy.2011.11.027

25. Z. J. Zhou, Y. M. Xiang, H. Xu, Y. S. Wang, D. Shi, Z. W. Wang, Self-organizing probability neural network-based intelligent non-intrusive load monitoring with applications to low-cost residential measuring devices, *Trans. Inst. Meas. Control*, **96** (2020), 635–645. https://doi.org/10.1177/0142331220950865

26. C. Chen, P. Gao, J. Jiang, H. Wang, P. Li, S. Wan, A deep learning based non-intrusive household load identification for smart grid in China, *Comput. Commun.*, **177** (2021), 175-184. https://doi.org/10.1016/j.comcom.2021.06.023

27. D. L. Su, Q. Shi, H. Xu, W. Wang, Nonintrusive load monitoring based on complementary features of spurious emissions, *Electronics*, **8** (2019), 1002. https://doi.org/10.3390/electronics8091002

28. F. Ciancetta, G. Bucci, E. Fiorucci, S. Mari, A. Fioravanti, A new convolutional neural network-based system for NILM applications, *IEEE Trans. Instrum. Meas.*, **70** (2020), 1501112. https://doi.org/10.1109/TIM.2020.3035193

29. D. Ding, J. Li, K. Zhang, H. Wang, K. Wang, T. Cao, Non-intrusive load monitoring method with inception structured CNN, *Appl. Intell.*, **30** (2021), 1–18. https://doi.org/10.1007/s10489-021-02690-y

30. H. Peng, J. Li, Y. Song, R. Yang, R. Ranjan, P. S. Yu, et al., Streaming social event detection and evolution discovery in heterogeneous information networks, *ACM Trans. Knowl. Discovery Data*, **15** (2021), 1–33. https://doi.org/10.1145/3447585

31. Z. Jia, L. Yang, Z. Zhang, H. Liu, F. Kong, Sequence to point learning based on bidirectional dilated residual network for non-intrusive load monitoring, *Int. J. Electr. Power Energy Syst.*, **129** (2021), 106837. https://doi.org/10.1016/j.ijepes.2021.106837

32. H. Peng, R. Zhang, Y. Dou, R. Yang, J. Zhang, P. S. Yu, Reinforced neighborhood selection guided multi-relational graph neural networks, *ACM Trans. Inf. Syst.*, **40** (2021), 1–46. https://doi.org/10.1145/3490181

33. C. Dinesh, S. Makonin, I. V. Bajić, Residential power forecasting using load identification and graph spectral clustering, *IEEE Trans. Circuits Syst. II Express Briefs*, **66** (2019), 1900–1904. https://doi.org/10.1109/TCSII.2019.2891704

34. H. Peng, J. Li, Z. Wang, R. Yang, M. Liu, M. Zhang, et al., Lifelong property price prediction: A case study for the toronto real estate market, *IEEE Trans. Knowl. Data Eng.*, **40** (2021). https://doi.org/10.1109/TKDE.2021.3112749

35. Z. Wu, C. Wang, H. Zhang, W. Peng, W. Liu, A time-efficient factorial hidden Semi-Markov model for non-intrusive load monitoring, *Electr. Power Syst. Res.*, **199** (2021), 107372. https://doi.org/10.1016/j.epsr.2021.107372

36. N. Henao, K. Agbossou, S. Kelouwani, Y. Dubé, M. Fournier, Approach in nonintrusive type I load monitoring using subtractive clustering, *IEEE Trans. Smart Grid*, **8** (2017), 812–821. https://doi.org/10.1109/TSG.2015.2462719

37. D. Yang, X. Gao, L. Kong, Y. Pang, B. Zhou, An event-driven convolutional neural architecture for non-intrusive load monitoring of residential appliance, *IEEE Trans. Consum. Electron.*, **66** (2020), 173–182. https://doi.org/10.1109/TCE.2020.2977964

38. T. T. H. Le, S. Heo, H. Kim, Toward load identification based on the hilbert transform and sequence to sequence long short-term memory, *IEEE Trans. Smart Grid*, **12** (2021), 3252–3264. https://doi.org/10.1109/TSG.2021.3066570

39. H. Rafiq, X. Shi, H. Zhang, H. Li, M. K. Ochani, A. A. Shah, Generalizability improvement of deep learning based non-intrusive load monitoring system using data augmentation, *IEEE Trans. Smart Grid*, **99** (2021), 75–114. https://doi.org/10.1109/TSG.2021.3082622

40. S. Makonin, F. Popowich, I. V. Bajić, B. Gill, L. Bartram, Exploiting HMM sparsity to perform online real-time nonintrusive load monitoring, *IEEE Trans. Smart Grid*, **7** (2016), 2575–2585. https://doi.org/10.1109/TSG.2015.2494592

41. S. P. Cai, Z. M. Sun, J. Yan, D. H. Tang, Y. Chen, Z. Y. Zhou, Fisher information and online SVR-based dynamic modeling methodology for meteorological sensitive load forecasting in smart grids, *IEEE Trans. Smart Grid*, **104** (2021), 513–527. https://doi.org/10.1007/s00202-021-01308-3

42. V. Álvarez, S. Mazuelas, J. A. Lozano, Probabilistic load forecasting based on adaptive online learning, *IEEE Trans. Smart Grid*, **36** (2021), 3668–3680. https://doi.org/10.1109/TPWRS.2021.3050837

43. Y. Du, F. Li, Intelligent multi-microgrid energy management based on deep neural network and model-free reinforcement learning, *IEEE Trans. Smart Grid*, **11** (2019), 1066–1076. https://doi.org/10.1109/TSG.2019.2930299

44. C. Wang, S. Mei, H. Yu, S. Cheng, L. Du, P. Yang, Unintentional islanding transition control strategy for three-/single-phase multimicrogrids based on artificial emotional reinforcement learning, *IEEE Syst. J.*, **15** (2021), 5464–5475. https://doi.org/10.1109/JSYST.2021.3074296

45. H. Peng, H. Wang, B. Du, M. Bhuiyan, H. Ma, J. Liu, et al., Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting, *Inf. Sci.*, **521** (2020), 277–290. https://doi.org/10.1016/j.ins.2020.01.043

46. H. Peng, H. Li, Y. Song, V. W. Zheng, J. Li, Differentially private federated knowledge graphs embedding, *IEEE Trans. Knowl. Data Eng.*, **40** (2021). https://doi.org/10.1145/3459637.3482252

47. Y. Li, R. Wang, Z. Yang, Optimal scheduling of isolated microgrids using automated reinforcement learning-based multi-period forecasting, *IEEE Trans. Sustainable Energy*, **13** (2022), 159–169. https://doi.org/10.1109/TSTE.2021.3105529

48. D. Cao, W. Hu, X. Xu, Q. Wu, Q. Huang, Z. Chen, et al., Deep reinforcement learning based approach for optimal power flow of distribution networks embedded with renewable energy and storage devices, *J. Mod. Power Syst. Clean Energy*, **9** (2021), 1101–1110. https://doi.org/10.35833/MPCE.2020.000557
49. J. Li, H. Peng, Y. Cao, Y. Dou, H. Zhang, P. S. Yu, et al., Higher-order attribute-enhancing heterogeneous graph neural networks, *IEEE Trans. Knowl. Data Eng.*, **40** (2021). https://doi.org/10.48550/arXiv.2104.07892
50. L. Xi, L. Zhou, L. Liu, D. Duan, Y. Xu, L. Yang, et al., A deep reinforcement learning algorithm for the power order optimization allocation of AGC in interconnected power grids, *CSEE J. Power Energy Syst.*, **6** (2020), 712–723. https://doi.org/10.17775/CSEEJPES.2019.01840
51. D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, et al., Mastering the game of Go without human knowledge, *Nature*, **550** (2017), 354–359. https://doi.org/10.1038/nature24270
52. R. Yao, X. Lu, H. Zhou, J. Lai, A novel category-specific pricing strategy for demand response in microgrids, *IEEE Trans. Sustainable Energy*, **12** (2020), 182–195. https://doi.org/10.1109/TSTE.2021.3106329
53. Z. Xiao, C. Fan, J. Yuan, X. Xu, W. Gang, Comparison between artificial neural network and random forest for effective disaggregation of building cooling load, *Case Stud. Therm. Eng.*, **28** (2021), 101589. https://doi.org/10.1016/j.csite.2021.101589
54. R. Yao, H. Zhou, D. Zhou, H. Zhang, State characteristic clustering for nonintrusive load monitoring with stochastic bhaviours in smart grids, *Complexity*, **2021** (2021), 8839595. https://doi.org/10.1155/2021/8839595
55. H. Peng, J. Li, Q. Gong, Y. Song, P. S. Yu, Fine-grained event categorization with heterogeneous graph convolutional networks, in *Twenty-Eighth International Joint Conference on Artificial Intelligence IJCAI-19*, (2019), 3238–3245. https://doi.org/10.48550/arXiv.1906.04580
56. Y. J. Zhang, L. Yu, Z. J. Fang, N. N. Xiong, L. J. Zhang, H. Y. Tian, An end-to-end deep learning model for robust smooth filtering identification, *Future Gener. Comput. Syst.*, **127** (2021), 182–195. https://doi.org/10.1016/j.future.2021.09.004
57. Q. Liu, K. M. Kamoto, X. Liu, M. Sun, N. Linge, Low-complexity non-intrusive load monitoring using unsupervised learning and generalized appliance models, *IEEE Trans. Consum. Electron.*, **65** (2019), 28–37. https://doi.org/10.1109/TCE.2019.2891160
58. H. Shuai, H. He, Online scheduling of a residential microgrid via Monte-Carlo tree search and a learned model, *IEEE Trans. Smart Grid*, **12** (2020), 1073–1087. https://doi.org/10.1109/TSG.2020.3035127
59. T. Shao, H. Zhang, K. Cheng, K. Zhang, L. Bie, The hierarchical task network planning method based on Monte Carlo tree search, *Knowl.-Based Syst.*, **225** (2021), 107067. https://doi.org/10.1016/j.knosys.2021.107067
60. R. Lu, S. H. Hong, M. Yu, Demand response for home energy management using reinforcement learning and artificial neural network, *IEEE Trans. Smart Grid*, **10** (2019), 6629–6639. https://doi.org/10.1109/TSG.2019.2909266

61. V. de Carvalho Neiva Pinheiro, A. L. Francato, W. B. Powell, Reinforcement learning for electricity dispatch in grids with high intermittent generation and energy storage systems: A case study for the Brazilian grid, *Int. J. Energy Res.*, **44** (2020), 8635–8653. https://doi.org/10.1002/er.5551
62. H. Shuai, H. He, Online scheduling of a residential microgrid via Monte-Carlo tree search and a learned model, *IEEE Trans. Smart Grid*, **12** (2020), 1073–1087. https://doi.org/10.1109/TSG.2020.3035127
63. X. Liu, A. Fotouhi, Formula-E race strategy development using artificial neural networks and Monte Carlo tree search, *Neural Comput. Appl.*, **32** (2020), 15191–15207. https://doi.org/10.1007/s00521-020-04871-1
64. E. J. Powley, P. I. Cowling, D. Whitehouse, Memory bounded Monte Carlo tree search, in *Thirteenth Artificial Intelligence and Interactive Digital Entertainment Conference*, **13** (2017), 94–100. Available from: https://ojs.aaai.org/index.php/AIIDE/article/view/12932.
65. R. Bonfigli, A. Felicetti, E. Principi, M. Fagiani, S. Squartini, F. Piazza, Denoising autoencoders for non-intrusive load monitoring: improvements and comparative evaluation, *Energy Build.*, **158** (2018), 1461–1474. https://doi.org/10.1016/j.enbuild.2017.11.054
66. Y. Himeur, A. Alsalemi, F. Bensaali, A. Amira, Smart non-intrusive appliance identification using a novel local power histogramming descriptor with an improved k-nearest neighbors classifier, *Sustainable Cities Soc.*, **67** (2021), 102764. https://doi.org/10.1016/j.scs.2021.102764
67. L. D. Nolasco, A. E. Lazzaretti, B. M. Mulinari, DeepDFML-NILM: A new CNN-based architecture for detection, feature extraction and multi-label classification in NILM signals, *IEEE Sens. J.*, **22** (2022), 501–509. https://doi.org/10.1109/JSEN.2021.3127322
68. Z. Huang, X. Wei, Y. Kai, Bidirectional LSTM-CRF models for sequence tagging, *Comput. Sci.*, **2015** (2015). https://doi.org/10.48550/arXiv.1508.01991