MultiCycleNet: Multiple Cycles Self-Boosted Neural Network for Short-term Electric Household Load Forecasting

Rinan Chen [a], Chun Sing Lai [b,c,*], Cankun Zhong [a], Keda Pan [b], Wing W. Y. Ng [a,*], Zhanlian Li [b], Loi Lei Lai [b,*]

[a] Guangdong Provincial Key Laboratory of Computational Intelligence and Cyberspace Information, School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

[b] Department of Electrical Engineering, School of Automation, Guangdong University of Technology, Guangzhou 510006, China

[c] Brunel Institute of Power Systems, Department of Electronic and Electrical Engineering, Brunel University London, London, UB8 3PH, UK

* Corresponding authors: chunsing.lai@brunel.ac.uk; wingng@ieee.org; l.l.lai@ieee.org

Abstract— Household load forecasting plays an important role in future grid planning and operation. However, compared with aggregated load forecasting, household load forecasting faces the challenge of the uncertainty of prolific load profiles. This paper presents a novel multiple cycles self-boosted neural network (MultiCycleNet) framework for household load forecasting, which aims to solve the uncertainty problem of household load profiles through the correlation analysis of electricity consumption patterns in multiple cycles. The basic idea of the proposed framework is that the predictor can learn customers' power consumption patterns better by learning the features and contextual information of relevant load profiles in multiple historical cycles. We use two real-life datasets: 1. the household load consumption dataset from Low Carbon London project led by United Kingdom (UK) Power Networks and 2. the UK Domestic Appliance-Level Electricity (UK-DALE) dataset to evaluate the effectiveness of the proposed framework. Compared with the state-of-the-art methods, experimental results show that the proposed framework is effective and outperforms the state-of-the-art methods by 19.83%, 10.46%, 11.14% and 9.02% in terms of mean squared error, root mean squared error, mean absolute error and mean absolute percent error, respectively.

***Index Terms*—Load forecasting, recurrent neural network, time-series forecasting, multiple historical cycles.**

1. Introduction

With the progress of industrial infrastructure and technology, the power system is developing towards the direction of intelligence, which provides the possibility to improve the efficient utilization of power grid in the future. As reported in [1], in smart cities where Information Communication and Technology (ICT) is merged with the existing traditional infrastructure to coordinate and manage with digital technology. At the core of smart city lies the sensors and actuators embedded in the smart devices that sense the environment to facilitate effective decision making. This idea of smart cities is coming into reality as many countries around the world are developing their smart city models to include domains such as smart energy, smart building and smart health [2].

Accurate load forecasting significantly influences the future power planning and scheduling, especially for household load forecasting [3]. However, unlike system-level or other-level aggregated load forecasting, household load forecasting is extremely challenging because of the high stochastic nature of load profiles. Typically, techniques including aggregation, clustering and preprocessing can be used to reduce the problem with uncertainty. However, these techniques are indirectly avoiding the uncertainty of household load forecasting [4]. For example, aggregation can be used to reduce the uncertainties so that the final load shows mostly regular patterns and easier to perform forecasting, but the problem of low individual prediction accuracy is not essentially solved.

Recently, the development of deep learning [5] provides a new direction for researchers to study load forecasting. Compared with traditional machine learning, e.g., *k*-nearest neighbors (KNN), one of top ten data mining algorithms [6], deep neural network has large-scale parameters, excellent activation functions and flexible training algorithms, which enable learning nonlinear mapping relationship between data better [5]. Long short-term memory neural network (LSTM) [7], a deep neural network which is widely used to process time series, has been used to study household load forecasting by researchers and achieved good results [4], [8]. These methods had made progress compared with traditional forecasting method. However, the correlation of the household load profile in different historical cycles are not considered with only a load profile of a period as an independent sample be trained and predicted, which make model difficult to learn customers' own history power patterns adequately. In other words, the challenge of the uncertainty of household load profiles still exists.

Some studies claim that there is a high correlative relationship between the household load profiles and the behaviors [9]. For example, a family may have a fairly consistent pattern of daily behavior, and if this pattern can be obtained, then better forecast results can be expected. But these methods require additional information, such as usage information about electrical appliances, which are not easily available. On the other hand, some researchers have studied the similarity between different household load profiles to improve the prediction accuracy [10], [11]. These methods need to determine the similarity among customers' load patterns, which depends on the characteristics of the dataset. In this paper, we study how to take advantage of the correlation of a household load profile itself to learn customer's power consumption pattern, so as to improve the prediction accuracy.

More frequently sampled time series (every half hour and every hour) are common in many industries due to the rapid development of sensors and data storage capabilities. Taking load profile of hourly sampling frequency as an example, the power consumption pattern of a family is relatively fixed in a short period, and we can consider using this characteristic to help improving the performance of load forecasting. However, the patterns of power consumption at different times of the day are still quite different. For example, the power consumption patterns in the morning is significantly different from that at night due to the daily activities. It seems inappropriate to expect to eliminate the uncertainty problem through the load profile over a relatively long period. Therefore, we decide to reduce the scale of power consumption patterns to a smaller granularity, such as hour level, and focus on the power consumption pattern of households at the same time of every day for a relatively long period. In other words, if we want to identify load profiles that have a high correlation with the load to be predicted over multiple historical cycles, the result often points to the load profiles that are at the same time as the load to be predicted.

Generally, the correlative historical load profiles and the load to be predicted have highly similar external factors, so they can reflect the common features of data under certain conditions. Furthermore, we can use the contextual information of the load profile, i.e., historical forward and backward data, to learn customers' power consumption patterns, which is similar to the idea of bidirectional recurrent neural networks [12]. In summary, this paper presents a novel MultiCycleNet framework for short-term household load forecasting. The advantage of the proposed framework is that it considers the correlation and the contextual information of household load profiles in multiple cycles, which make full use of a load profile itself to reflect the power consumption pattern of the customer. In fact, a load profile may contain repeated cycles. If the correlative data of these repeated cycles is fed into the model, it would learn the electricity consumption pattern accurately. Therefore, compared to the method that only selects a single cycle of data as input, the proposed method uses multiple cycles information of data to obtain more usable information. At the same time, since only correlative data would be selected, it would not introduce too much redundant data to the model.

We evaluate the proposed framework on the household load consumption dataset from Low Carbon London project led by United Kingdom (UK) Power Networks and the UK Domestic Appliance-Level Electricity (UK-DALE) dataset, and compare it with other forecasting methods. The experimental results show that the proposed method is effective and outperforms the comparative methods in various metrics.

The key contributions in this paper are as follows:

1. This paper proposes a new framework to solve the uncertainty problem of household load forecasting. The proposed framework takes advantage of the correlative and contextual information of the load profile in multiple historical cycles to learn household power consumption pattern. To the best of our knowledge, this is the first time that correlative load series considering contextual information from historical data have been used to learn household power consumption pattern.
2. In the case of fluctuating household load profiles, the proposed method can still learn the household power consumption pattern well and make a reasonable prediction.
3. Experimental results show that the proposed framework is effective and far superior to the comparative methods, which include traditional machine learning algorithms and the state-of-the-art methods in this field.

The rest of this paper is organized as follows. In Section 2, we introduce the related work in load forecasting. Section 3 presents the theory of the proposed framework. Experimental setup and analysis are presented in Section 4. Finally, Section 5 gives the conclusion.

2. Related works

There are many traditional time-series forecasting methods, such as exponential smoothing (ES) [13] and autoregressive integrated moving average (ARIMA) [14]. In practice, however, the uncertainty of load patterns would cause difficulties for model to make prediction, especially for household load forecasting. Some techniques can be used to reduce the uncertainty challenge, such as aggregation, clustering and preprocessing. Sun *et al.* [15] proposed an approach within hierarchical structure for load forecasting at distribution level, in which user-level loads acted as child nodes by aggregating to form parent nodes. The method made use of the similarity between parent and child node to predict load demand from different levels. The regional aggregated load demand forecasting was addressed via combining sister forecasts by Nowotarski *et al.* [16]. An ensemble method [17] was applied for the aggregated load forecasting. First, a hierarchical clustering method was used to cluster the load profile. The prediction was then made separately within each cluster. Finally, the weighted summation predicted results was carried out to obtain the final aggregation load prediction. It generated multiple predictions by changing the number of clusters, which is quite different from the traditional ensemble method. The factor of influence for load forecasting was studied by Wang using k-means [18], a classical and widely used clustering method. A wavelet-based ensemble scheme [19] was applied to generate individual extreme learning machine (ELM)-base predictor [20]. Then an algorithm combining ELM and Levenberg-Marquardt [21] was used to improve the forecasting accuracy of neural networks. Leung *et al.* [22] investigated the use of occupancy space electrical power demand for building cooling load prediction. The Levenberg-Marquardt algorithm was used to process the input data, including the usual external climatic data, pretreated air unit operation schedule and the occupancy space electrical power demand, and then obtained the electrical power demand of the building cooling system. The results revealed that the use of occupancy space power demand would enhance the accuracy of the cooling load prediction. Qiu *et al.* [23] first used empirical mode decomposition (EMD) to decompose load demand series into several intrinsic mode functions (IMFs), then deep learning networks were applied to model each of the extracted IMFs. The above methods can obtain good results in load forecasting in some extent, however, they are indirectly avoiding the uncertainty of household load profiles [4]. In [24], a Particle swarm optimization (PSO)-based [25] improved Wang–Mendel (WM) [26] method was proposed, which combined modeling method based on fuzzy systems and evolutionary algorithms. A modified PSO algorithm was used to optimize the fuzzy rule centroid of the data coverage area, and the complete fuzzy rule set was obtained by extrapolation. Panapakidis *et al.* [27] studied short-term load forecasting for the bus. First, a modified version of artificial neural network was applied to deal with the aggregated load profiles of interconnected systems. Then, in order to enhance the prediction accuracy of the neural network, two new hybrid forecasting models were proposed using the load profiling method. These new models would finally combine the neural network with a clustering algorithm and predicted load profiles for the bus. Zheng *et al.* [3] proposed a Kalman filter-based bottom-up approach for household short-term future load forecasting, and analyzed the accuracy difference between appliance level and home level using the conventional and the bottom-up strategy, respectively.

In recent years, deep learning has made great progress in many fields, such as image recognition [28], machine translation [29], and speech recognition [30]. To this end, researchers tried to use deep neural network to solve the problem of load forecasting. In [31], an interval prediction method was proposed by establishing an index table containing all predication intervals results based on Gaussian distribution and using modified convolution neural network (CNN) [32]. Wu *et al.* [33] studied load forecasting for air compressor system using artificial neural network. Two kinds of artificial neural network, two-layer feed-forward neural network and LSTM, were used to predict the load profiles of air compressor. The results indicated that both artificial neural networks achieved good results for compressors using variable speed drive, only LSTM gave acceptable results for compressors using on/off control, and the results of both artificial neural networks were not satisfactory for load/unload type air compressors. Lai *et al.* [34] proposed a multi-view ensemble framework for short and mid-term load forecasting. Features of multi-view were first extracted from both LSTM and three-level wavelet decomposition. These features combining with some exogenous variables are then used to train the base predictors. Kong *et al.* [8] proposed a framework for short-term household load forecasting based on LSTM model. The case studies showed that LSTM can capture subtle patterns of power consumption and provide good prediction in most cases. A pooling-base deep recurrent neural network (PDRNN) model was proposed by Shi *et al.* [4]. It first fed load profiles of customers into different pools. Then load demand was predicted for each customer using the information shared in the same pool, which aims to solve the over-fitting problem in deep neural network. A data-driven deep learning framework [35] was applied to improve the accuracy of short-term peak load forecasting. First a copula model was used to learn the tail-dependence of power load on electricity price and temperature. Then a deep belief network (DBN) was used for short-term load forecasting using the learned tail-dependence. Chang *et al.* [36] proposed a hybrid model based on wavelet transform and Adam [37] optimized LSTM neural network (WT-Adam-LSTM) for electricity price forecasting. The sequence of the electricity price would be processed by wavelet transform, and then the combination of Adam and LSTM would be used to capture household behaviors for electricity price. In [38], a fuzzy clustering method was first applied to group the load profiles. Then a two-stage ensemble model combining radial basis function neural network (RBFNN) [39] and CNN was applied to model load demand in each group. Finally, the predicted results in each group were aggregated to form the final load prediction. A two-terminal sparse coding neural network proposed by Chen *et al.* [40] used affinity propagation (AP) to group the load profiles considering household power consumption similarity at first. Then a two-terminal sparse auto-encoder was applied to predict the load demand in each group. The encoder first extracted the features of load profiles and performed dimension reduction. The output of the encoder would take as input to a deep neural network composed of a variety of different structures. Duan *et al.* [41] proposed a prediction model for aggregated loads of buildings, which was composed of feature selection and enhanced support vector machine (ESVM) based forecast engine. Besides the aggregated loads of building, the electric vehicle (EV) impact on network was also considered in the method. In [42] a three-level hybrid ensemble short-term load forecasting method consisting of discrete wavelet transform (DWT), particle swarm optimization (PSO), and RBFNN was proposed. The DWT was applied to decompose the data, and PSO was used to obtain the required optimal adjustable parameters of the RBFNN for the forecasting. In [43] a load forecasting model based on an artificial neural network (ANN) was proposed to predict hourly load demand for various seasons of a year. In the model, a global best particle swarm optimization (GPSO) algorithm was applied to enhance the performance of prediction, and a weight bias encoding/decoding scheme was used to improve network training.

Some studies made use of the information of household appliances to improve model's prediction. Kong *et al.* [9] proposed a short-term memory deep learning framework for household load forecasting based on the information of household appliances. It trained a LSTM using household load profiles and other available appliances' consumption information both. Generally, household load pattern has a high correlation with family's behavior. Therefore, a better prediction can be expected if such a pattern can be obtained. The signal of appliances was studied by Dinesh *et al.* [44]. It first decomposed the aggregated household load profile into multiple individual appliances' signals. Then the model was used to predict these signals of appliances. The predicted results of the individual appliances were aggregated to form the final prediction. However, these methods require additional information that may be difficult to obtain. Some studies exploited the similarity among customers' load patterns to improve the prediction. Bandara *et al.* [10] proposed a method for load forecasting based on multiple seasonal patterns and related time series. It first grouped the related load profiles so as to take advantage of the cross-series knowledge among the related time series. Then a series of multiseasonal decomposition techniques were applied to process the load profiles. In the prediction stage, two approaches, deseasonalized approach and seasonal exogenous approach, were used to supplement the learning procedure of LSTM with previous preprocessed load profiles. Quilumba *et al.* [11] proposed a three-stage approach based on the similarity of customers' load patterns. It first grouped power consumption patterns according to the total consumption of a certain period of a day, so as to get a more balanced grouping of customers. Then load forecasting was performed for each group separately. The predicted results of each group were aggregated to form the final system-level load prediction. These methods take advantage of the relevant patterns of load profiles among different customers, but they did not consider the relevant patterns that exist in the customer's own load profiles.

There are some studies considering the correlation for load forecasting. Xu *et al.* [45] claimed that the load accuracy is inconsistent with the power purchase cost through cost computation. In other words, more accurate load forecasting models may not aim for optimal benefit. To solve this problem, Xu *et al.* [45] proposed a beneficial correlated regularization (BCR) method for neural network load forecasting. In the BCR, the training objectives of neural network included the accuracy section part and the power cost part. In [46], 24-hour equations were assembled to form a periodic autoregressive moving average model for load forecasting, which considered the hourly sequence as a periodic correlation process. The author proposed a new method to estimate the parameters of the multi-equation model. First, an independent model was obtained for each hour. Then the parameters for each hour were associated with the previous hours for a joint estimate. Besides, the above method took temperature as explanatory variable and considered the impact of holiday on demand. Although the studies mentioned above and the MultiCycleNet both study the correlation for load forecasting, there are fundamental differences between them. (i). The MultiCycleNet studies the correlation load series in multiple cycles in a customer's historical load profiles to find out the repeating pattern. (ii). The contextual information, i.e., the backward and forward correlative load series, is considered as the correlative load series at the same time to capture a more robust power consumption pattern. In order to facilitate the comparison, Table 1 summarizes the methods comparison between this study and related works.

TABLE 1

METHODS COMPARISON BETWEEN THIS STUDY AND RELATED WORKS

Literature author	Z. Zheng *et al.* [3]	H. Shi *et al.* [4]	C. S. Lai *et al.* [34]	T. Ouyang *et al.* [35]	Z. Chang *et al.* [36]	This study
Methodology	A bottom-up approach using Kalman filter-based model	A pooling-base deep recurrent neural network model (PDRNN)	Multi-view neural network ensemble model	A data-driven framework combining copula model and deep belief network (Copula-DBN)	A hybrid model based on wavelet transform and Adam optimized LSTM neural network (WT-Adam-LSTM)	Multiple cycles self-boosted neural network (MultiCycleNet)
Application scenarios	Household short-term load forecasting	Household load forecasting	Short and mid-term load forecasting	Short-term load forecasting	Electricity price forecasting	Household short-term load forecasting
Compared method	Persistence model, LSTM	Auto regressive integrated moving average (ARIMA), recurrent neural network (RNN), support vector regression (SVR), deep recurrent neural networks (DRNN)	Wavelet transform-radial basis function neural networks (WT-RBFNN), LSTM-radial basis function neural networks (LSTM-RBFNN), Ensemble model	Neural networks (NN), SVR, extreme learning machine (ELM), deep belief network (DBN)	General regression neural network (GRNN), LSTM	LSTM, PDRNN, WT-Adam-LSTM, Copula-DBN, Kalman filter-based model
Description	Proposed a Kalman filter-based bottom-up approach for household load forecasting, which achieves better predictive results than LSTM by aggregating the forecasts of appliances.	Proposed the PDRNN for household forecasting. It first fed load profiles of customers into different pools. Then load demand was predicted for each customer using the information shared in the same pool.	Proposed multi-view ensemble framework for short and mid-term load forecasting. Features of multi-view are first extracted from both LSTM and three-level wavelet decomposition. These features combining with some exogenous variables are then used to train the base predictors.	Proposed the Copula-DNB for short-term load forecasting. First, a copula model was used to learn the tail-dependence of power load on electricity price and temperature. Then a deep belief network was used for short-term load forecasting using the learned tail-dependence.	Proposed the WT-Adam-LSTM for electricity price forecasting. The electricity price data would be processed by wavelet transform, and the combination of Adam and LSTM would be used to capture household behaviors for electricity price.	Proposed a new framework to solve the uncertainty problem of household load forecasting. It made use of the correlative and contextual information of the load profiles in multiple cycles to learn household power consumption pattern.

3. Methodology

In this section, we first formally define the problem of household load forecasting with multiple cyclical load series. Subsequently, we describe the components of the proposed MultiCycleNet framework in detail.

3.1 Problem Statement

For household power consumption pattern, we assume that there is a cycle pattern that can be followed. In this work, we take a day as the minimum cycle. A sequence of household power consumption on day *d* can be represented as $\mathbf{X} = \{x_{d,1}, x_{d,2}, ..., x_{d,k}, ...\}$, where *k* represents the *k*-th half hour on day *d*.

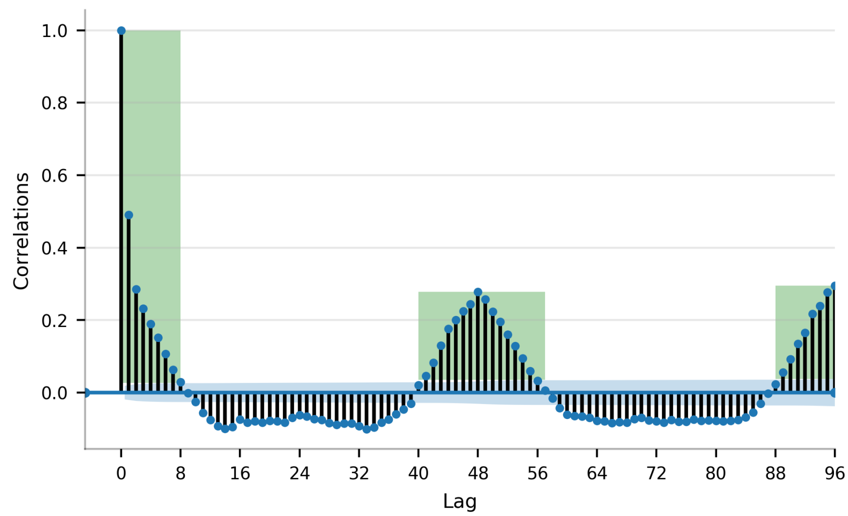

Fig. 1. Autocorrelation analysis of a household load profile.

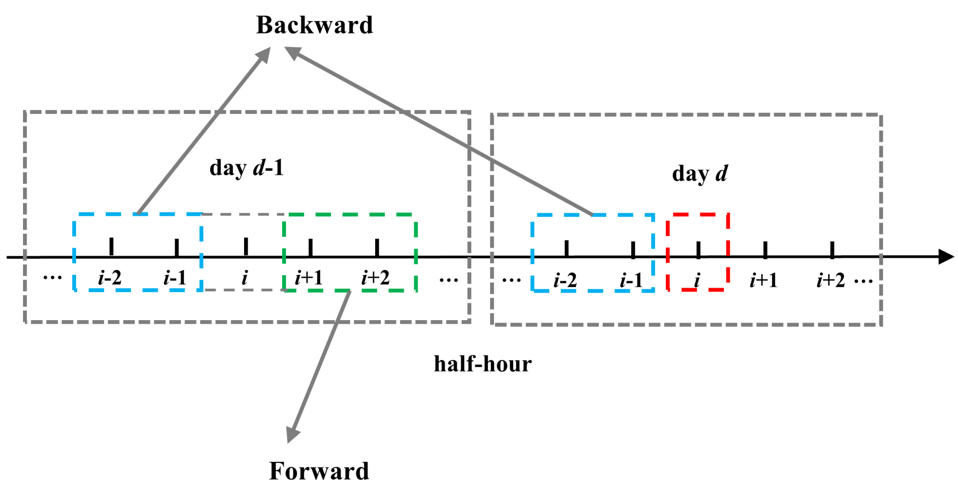

Fig. 2. Backward and forward data relative to the predicted load $x_{d,i}$, where d represents day d and i represents the i-th half hour on a day. Data in the red box represents the predicted load, and data in the blue and green boxes represents the backward and forward data relative to the predicted load, respectively.

Autocorrelation analysis between the load to be predicted $x_{d,i}$ and the historical load of the last 96 consecutive time steps (with time interval of half hour) is shown in Fig. 1. Blue shaded area represents the 95% significance level, and green shaded areas represent the positive correlation values greater than the 95% significance level. It can be observed from Fig. 1 that there are highly correlation between the historical load and the predicted load $x_{d,i}$, especially for the time stamp of $\{x_{(d-1),i}, x_{(d-2),i}, \ldots\}$. In addition, the backward and forward data relative to the predicted load is also important information for prediction. Fig. 2 is an example to illustrate the backward and forward data, in which the data in the blue and green boxes represents the backward and forward data relative to the predicted load, respectively. It is worth noting that the backward historical load series, such as

$\{x_{(d-1),(i-1)}, x_{(d-1),(i-2)}, \dots\}$, and the forward historical load series, such as $\{x_{(d-1),(i+1)}, x_{(d-1),(i+2)}, \dots\}$, around the load at the same time as the predicted load is highly correlative to $x_{d,i}$ as well. As mentioned in Section 1, these correlative load series usually have similar external factors, which can reflect the power consumption patterns of households under certain conditions. We use $\mathbf{X}_{c-di}$ to indicate the correlative load series of $x_{d,i}$ in multiple cycles. The model $f(\cdot)$ for household load forecasting can be defined as follows:

$$x_{d,i} = f(\mathbf{X}_{c-di}, \theta) \tag{1}$$

where θ is the parameters of the MultiCycleNet. Fig. 3 illustrates the key processes of the proposed framework. The time series in the first box represent the observations, which will be fed into the next layer later for data normalization. The multiple cycle processing layer would calculate the numbers of cycle and the sizes of window of backward and forward data. After that, the observations would be split into multiple parts according to the cycle and the window of backward and forward data, and then fed into the recurrent neural network model to get the forecast results. The MultiCycleNet framework has three layers: 1) normalization

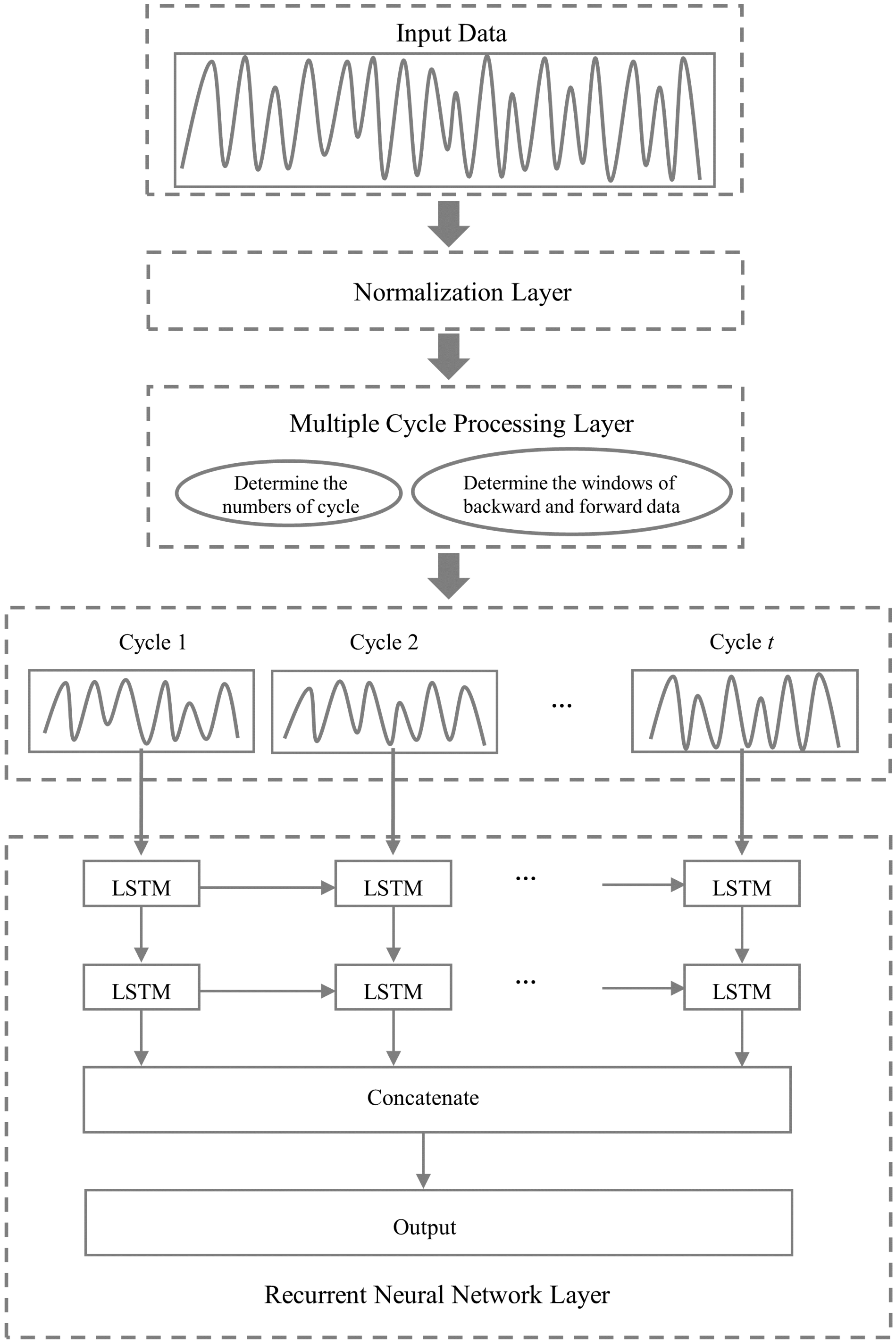

Fig. 3. The key processes of the MultiCycleNet framework.

layer; 2) multiple cycle processing layer; and 3) recurrent neural network layer. In the following section, we discuss each component of the MultiCycleNet framework in detail.

3.2 Normalization Layer

The MultiCycleNet is a predictor based on historical load profiles of households. Therefore, there is a huge difference between different household load profile ranges. In order to stabilize the predictor's training, normalization strategy is performed in the data preprocessing stage. In this paper, we use the min-max normalization method to preprocess the load profiles. To some extent, normalization can avoid the network saturation effect caused by the boundary of network activation function [47].

3.3 Multiple Cycle Processing Layer

This section discusses how to select the multiple historical load series for load prediction. There are two levels needed to be considered for the multiple historical load series, namely, the time level and the cycle level. The time level refers to the window that contains backward and forward load series at a certain time. The cycle level refers to the numbers of historical cycles. It is intuitive to think that the load at the same time in each cycle is highly correlated. In addition, for the load at a certain time, forward and backward load series around it is also correlated, which is determined by the household behaviors. Generally, household power consumption level is stable in a short period, so it needs to determine that short period in multiple historical cycles and obtain the corresponding load series. Then we will take advantage of the correlative load series above to learn the household power consumption patterns to solve the problem of uncertainty of load profiles.

3.3.1 Backward and Forward Load Series Selection

In order to determine the correlative load series on time level, partial autocorrelation function (PACF) is applied to analyze the characteristic of load series. Initially, household load profile in the first month of each quarter in a certain period is taken as a sample. Then statistics of correlation is made on those samples at the time level. Finally, mean value is chosen as final windows' value of correlative load series at the time level according to the statistical results. The selection process at the time level is presented in Table 2.

TABLE 2

CORRELATIVE LOAD SERIES SELECTION AT THE TIME LEVEL

Process1: Time Level Correlative Load Series Selection
Input: Household load profiles
Output: The windows' range of correlative load series at the time level
1. Extract every household load profile in the first month of each quarter in a certain year as a sample
2. Concatenate the above samples into a new dataset
3. Use PACF to analyze the autocorrelation of the new dataset
4. Select the lags with cyclical multiples as the centers
5. Count the number of other lags that greater than 95% significance level on the left and right sides of the centers
6. Compute the average of the counted values in 5

3.3.2 Multiple Cycles Selection

The key of correlative load selection at the cycle level is to determine the numbers of cycle to help model learn household power consumption patterns over a period of time. We use trail-and-error method to determine the numbers of correlative load at the cycle level. The selection process at the cycle level is presented in Table 3.

TABLE 3

CORRELATIVE LOAD SERIES SELECTION AT THE CYCLE LEVEL

Process2: Cycle Level Correlative Load Selection
Input: Household load profiles
Output: The windows' range of correlative load series at the cycle level
1. Take the number of days in a quarter as the value range at the cycle level
2. Generate a sequence S with same interval within the value range in Step 1
3. Use moving window transformation strategy on the historical load profiles to generate training samples with different cycle value according to the sequence S
4. Train different models using different training samples in Step 3
5. Determine the optimal numbers of cycle according to the performance among different models

3.4 Recurrent Neural Network Layer

LSTM is one of the most widely used variants of recurrent neural networks, which aims to solve the disability of standard recurrent neural network [48]. The key components of LSTM are the cell state and gates. The cell state is similar to the memory that remembers the state of information up to the present. The gates are used to control the selection of input information and the change of the cell state. Generally, the capability of the gate is realized through sigmoid function, tanh function, and point-wise multiplication operation. LSTM has three gates, namely, forget gate, input gate and output gate. The forget gate is used to control what information the cell state should discard. The input gate is used to determine what information the cell will store. The input gate consists of an input layer and a tanh layer. The input layer determines which values to update, and the tanh layer generates a candidate value that can be added to the cell state. By combining the outputs of the input layer and the tanh layer, input gate generates the final update information for the cell state. Cell state's update is jointly determined by the forget gate and the input gate. In other word, information is discarded and updated at the same time. The output gate determines what the model will output. In the output stage, the model uses the sigmoid function to determine the candidate output at current time step. Then the updated cell state is processed through a tanh layer to generate a new vector and the two processing results above are combined to create the final output values. The derivation formula of LSTM is as follows [7]:

$$f_t = \sigma(\boldsymbol{W_f} \cdot [h_{t-1}, x_t] + b_f) \tag{2}$$

$$i_t = \sigma(\boldsymbol{W_i} \cdot [h_{t-1}, x_t] + b_i) \tag{3}$$

$$\tilde{C}_t = \tanh(\boldsymbol{W_C} \cdot [h_{t-1}, x_t] + b_C) \tag{4}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{5}$$

$$o_t = \sigma(\boldsymbol{W}_o \cdot [h_{t-1}, x_t] + b_o) \tag{6}$$

$$h_t = o_t \odot \tanh(C_t) \tag{7}$$

where f_t, i_t, and o_t represent forget gate, input gate, and output gate of the LSTM cell respectively, while x_t represents the input at time step *t*. Also $\tilde{C}_t$ represents the candidate state, which is used to generate the cell state C_t. The hidden output of the cell at different time steps are represented by h_{t-1} and h_t. W_f, W_i, W_o and W_c represent weight matrices of forget gate, input gate, output gate, and tanh layer of the input gate. The biases of the gates are denoted as b_f, b_i, b_o and b_c respectively. In these equations, $\odot$

represents the point-wise multiplication operation, σ represents the sigmoid activation function, and tanh stands for the hyperbolic tangent function. The mean squared error (MSE) is used as the loss function of the proposed framework. The MSE is given by:

$$\mathcal{L} = \frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2 \tag{8}$$

where y_t and $\hat{y}_t$ refer to the actual load value and the prediction of model at time step *t* respectively. Also, *n* is the number of samples in the training set. MSE is a widely used metric in regression problem, which essentially minimizes the square differences between the actual and the predicted value.

4. Experiment

4.1 DataSet Description

To evaluate the performance of the proposed method, this study used two public datasets. Dataset I is the Smart Meter Energy Consumption Data in London Households, which took part in the UK Power Networks led Low Carbon London project between November 2011 and February 2014 [49]. The dataset had gathered load profiles for 5,567 customers in London containing energy consumption, in kWh (per half hour), household identifier, date, time and acorn group. There are two different groups in the dataset. The first group is subjected to Dynamic Time of Use (dToU) energy prices throughout 2013. Another group is not subject to the dToU tariff. These groups are equally treated in the experiment since we focus on the problem of uncertainty of household load forecasting. Dataset II is the UK-DALE dataset [50] which was collected from five residential houses in the UK from a period of about five years. Since the first house in the UK-DALE dataset contains complete appliance usage information, only data from the first house in UK-DALE was used in the experiment. In this work, the dataset is split into training set and test set. The splitting process is as follows: For a household long-term load profile spanning several years, load profile located in the previous part is taken as the training data. The subsequent part of the load profile is used as the test data to verify the prediction accuracy of model. More details about the process of data splitting can be found in Section 4.6.

4.2 Error Metrics

The mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), mean absolute percent error (MAPE) and symmetric mean absolute percent error (SMAPE) are metrics used to evaluate the predictive performance of the models [51]. The MSE, RMSE, MAE, MAPE and SMAPE are defined as follows:

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2 \tag{9}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \tag{10}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}\left|y_t - \hat{y}_t\right| \tag{11}$$

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right| \times 100\% \tag{12}$$

$$SMAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{\left|y_t - \hat{y}_t\right|}{\left|y_t\right| + \left|\hat{y}_t\right|} \times 200\% \tag{13}$$

where y_t and $\hat{y}_t$ refer to the true value and the predicted value of the model at time step *t*, respectively. Also, *n* is the number of samples in the test set. The dataset is first divided into training set and test set, and then the models are trained several times to obtain their best performance on the test set. After that, the error metrics mentioned above are applied to evaluate the performance of the model. Furthermore, when calculating the MAPE and SMAPE values for the load dataset, to avoid problems for zero forecasts and actual observations, we add a constant term 1 to the denominator of (12) and (13).

4.3 Hyperparameter Tuning and Optimization

The hyperparameters play an important role in model's prediction besides the internal parameters of model. Random search [52] is a simple and useful method for hyperparameters tuning, which can provide more possibilities and choices than grid search. However, random search cannot use previous hyperparameters tuning results to improve the subsequent tuning process. Annealing [53] schedule hyperparameters tuning is a variant of random search, which can take advantage of previous hyperparameters tuning results to improve the subsequent tuning process while retains the characteristics of random search. When using the annealing method to search for hyperparameters, it needs to define the search space of hyperparameters and the number of trials in advance. In our experiment, the maximum number of trials for each model is 1000. The value of the parameter is randomly determined during the first trial. In the next trial, the algorithm will randomly perturb the hyperparameter combination tried in the previous step to generate a new solution. If the new solution has a better result, it will be accepted. However, if the result becomes worse, the new solution will only be accepted with a certain probability. It is worth noting that this method is suitable for use when the test time of each combination is not long and there are enough computing resources.

In fact, the models used in the experiment have many hyperparameters. Taking LSTM as an example, the different combinations of hyperparameters, such as the activation function, the number of network layers and the optimizer, may produce different prediction results. Since the number of combinations is too large, it would take a lot of time to try out all of these combinations. Aiming to evaluate the effectiveness of the proposed framework, the authors have selected a few but important hyperparameters for tuning. To this end, the main hyperparameters are selected as the ultimate tuning targets, which are listed in Table 4.

TABLE 4

HYPERPARAMETERS TUNING TARGETS SUMMARY

Model hyperparameters	Minimum value	Maximum value
Long short-term memory neural network (LSTM) cell dimension	50	800
Gate recurrent unit (GRU) cell dimension	50	800
Temporal convolutional networks (TCN) filter	50	800
TCN kernel	1	6
Batch size	50	600
Epoch	10	50
Decision tree max depth	2	30
Decision tree min samples split	2	30
Decision tree min sample leaf	1	30
K nearest neighbors (KNN) neighbor	3	30
Support vector regression (SVR) epsilon	0	1
SVR penalty parameter	0	1
SVR kernel	{linear, poly, rbf}	

4.4 State-of-the-art methods comparison

The proposed method is compared with other traditional methods including support vector regression (SVR), decision tree, *k*-nearest neighbors regression (KNN), gate recurrent unit (GRU) [54], temporal convolutional networks (TCN) [55], light gradient boosting machine (LightGBM) [56] and LSTM. These methods cover machine learning methods and deep learning methods.

Subsequently, several current state-of-the-art methods in the space of load forecasting are used to compare with the proposed method. For example, Shi *et al.* [4] proposed a novel pooling-based deep recurrent neural network (PDRNN) which used customer-pool technique to solve the challenges in household load forecasting and over-fitting. Kong *et al.* [8] proposed the LSTM* method which made use of various load-related information, such as time, week, and holiday information. Ouyang *et al.* [35] used DBN and a copula model to improve the prediction of peak load. Chang *et al.* [36] proposed a hybrid model based on wavelet transform and Adam optimized LSTM neural network (WT-Adam-LSTM) for electricity price forecasting. Zheng *et al.* [3] proposed the Kalman filter-based bottom-up approach for household short-term future load forecasting, and analyzed the accuracy difference between appliance level and home level.

4.5 Multiple Cycle backward and forward historical data

The parameters about the cycles used in the MultiCycleNet is identified by the method mentioned in Section 3. Fig. 4 illustrates that using 9 previous cycles is a better choice than others in this case. Generally, using more cycles capture more information, but also has a higher chance of containing more noise, which will lead to deterioration and volatility of the model performance. It can be seen from Fig. 4 that higher volatility appears with a large number of cycles, such as 63-87 than small cycle numbers, such as 3-27.

To evaluate the impact of the contextual information of the load profiles on the predicted results in multiple cycles, we set up two experiments to compare the results between 1) not using historical forward load series as the input, and 2) using both backward and forward load series as the input. A sample that does not contain forward load series and a sample that contains backward and forward load series both are shown in Fig. 5. It is assumed that half-hour *i* at day *d* is the time of load that we want to predict, the blue and red boxes represent backward and forward load series respectively. Table 5 shows the experimental results. It can be seen that using both backward and forward load series outperform not using historical forward load series with respect to mean MSE, mean RMSE and mean MAPE values.

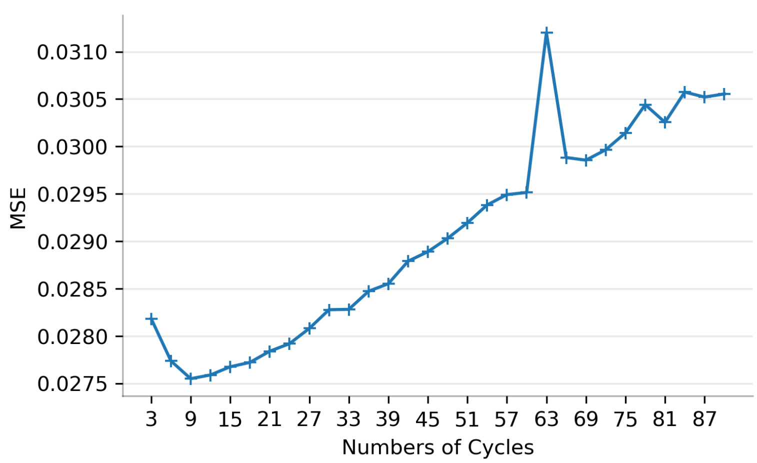

Fig. 4. The prediction accuracy of model using incrementally growing numbers of cycle.

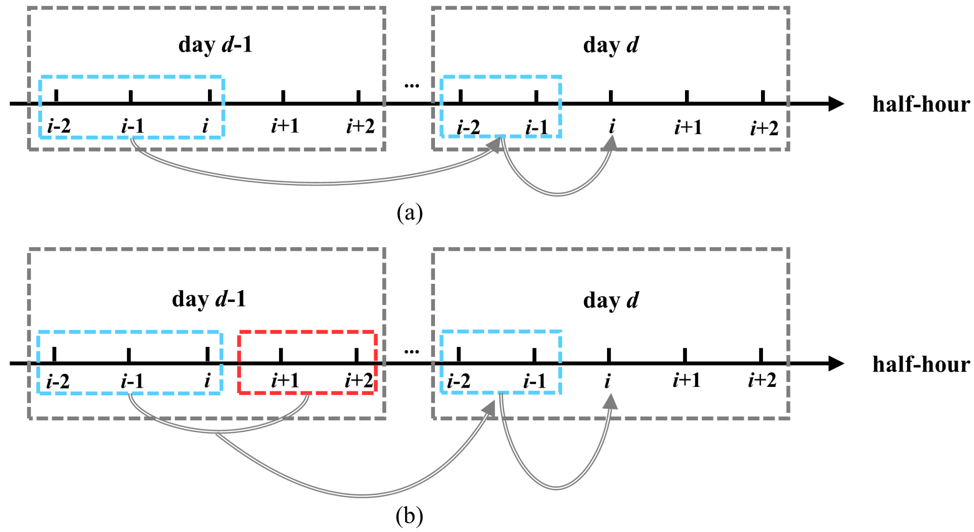

Fig. 5. The blue box represents the backward load data and the red box represents the forward load data, and *i* represents the *i*-th half hour on a day. (a) The load series in the blue boxes are taken as the input. (b) The load series in the blue and red boxes are taken as the input.

TABLE 5

THE IMPACT OF THE CONTEXTUAL INFORMATION OF LOAD PROFILES

Model	Mean MSE $(kWh)^2$	Mean RMSE (kWh)	Mean MAE (kWh)	Mean MAPE (%)
No-forward	0.0281	0.1676	**0.0799**	5.88
Backward + forward	**0.0278**	**0.1667**	0.0804	**5.82**

4.6 Results and discussion

In this experiment, time series cross-validation is used to evaluate the performance of models. There are a series of test datasets in the cross-validation process, and each test dataset is composed of some individual observations. The corresponding training data

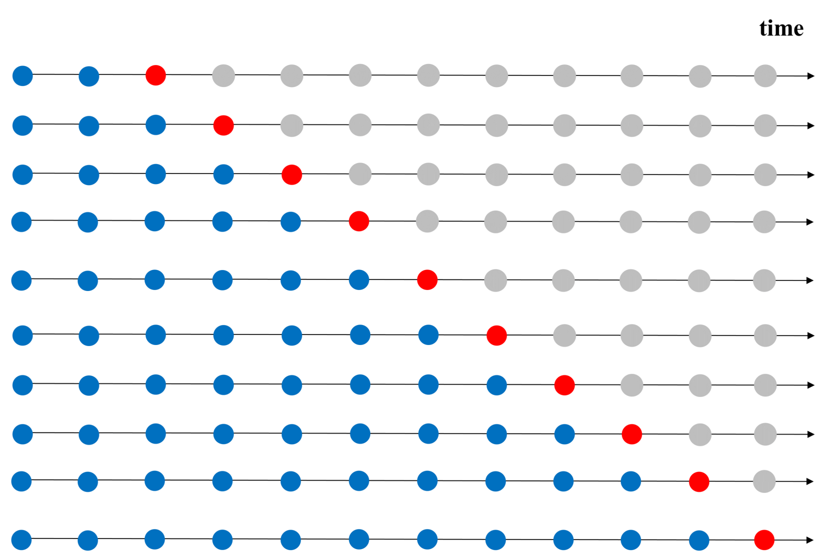

Fig. 6. Time series cross-validation procedure. Each row represents a data division, where the blue points form the training dataset, and the red points form the test dataset. No future observations can be used to construct predictions.

set only contains observations that occurred before the observations that make up the test dataset. Therefore, no future observations can be used to construct predictions. Fig. 6 illustrates the cross-validation procedure in the experiment, in which the blue points form the training dataset and the red points form the test dataset. 10-fold cross-validation was performed and their mean value was used as the final prediction result in this study.

Table 6 summarizes the evaluation results of all comparative methods and the MultiCycleNet for the London, UK household load consumption dataset. The results in Table 6 show that MultiCycleNet achieves the best results on each metric and outperforms the state-of-the-art methods by 19.83%, 10.46%, 11.14% and 9.02% in terms of mean MSE, mean RMSE, mean MAE and mean MAPE respectively. It is notable that deep learning methods, e.g., LSTM, GRU, and TCN receive better average performance compared to those in traditional machine learning, e.g., SVR, decision tree, and KNN, with respect to mean MSE and mean RMSE. However, it is also noticeable that SVR outperforms some deep neural networks with respect to mean MAPE. The phenomenon that SVR outperforms deep neural networks in some metrics was noticed in [4], and KNN does not perform very well in individual forecasting in [8]. LightGBM is a gradient boosting framework that uses tree-based learning algorithms and has faster training speed and lower memory usage. In the experiment, LightGBM is the fastest model to complete the training, and the operation efficiency is very high. In terms of predictive performance, LightGBM is better than traditional machine learning algorithms, but it is still not as good as deep learning models.

TABLE 6

COMPARATIVE EXPERIMENT RESULTS FOR THE LONDON, UK HOUSEHOLD LOAD CONSUMPTION DATASET

Model	Mean MSE $(kWh)^2$	Mean RMSE (kWh)	Mean MAE (kWh)	Mean MAPE (%)
SVR	0.0427	0.2067	0.0912	6.24
Decision tree	0.0721	0.2686	0.1246	8.90
KNN [6]	0.0421	0.2052	0.0997	6.82
LSTM [7]	0.0386	0.1963	0.0929	6.40
GRU [54]	0.0387	0.1968	0.0949	6.56
TCN [55]	0.0384	0.1961	0.0889	5.99
LSTM* [8]	0.0351	0.1874	0.0914	6.39
PDRNN [4]	0.0348	0.1865	0.0891	6.20
Copula-DBN [35]	0.0488	0.2209	0.0993	7.12
WT-Adam-LSTM [36]	0.0376	0.1938	0.0899	6.12
LightGBM [56]	0.0411	0.2028	0.0920	6.21
MultiCycleNet	**0.0279**	**0.1670**	**0.0790**	**5.45**

For the state-of-the-art methods, PDRNN used customer-pool technique aiming to solve the over-fitting problem, and LSTM* made use of various load-related information, such as time, week, and holiday information, and Copula-DBN used copula model to compute the information about peak load. WT-Adam-LSTM used the wavelet transform to process the nonlinear data for having a more stable variance. However, MultiCycleNet does not use the technique or the information above and outperforms the state-of-the-art methods, i.e., PDRNN, LSTM*, Copular-DBN and WT-Adam-LSTM, by 19.83%, 10.46%, 11.34% and 10.95% in terms of mean MSE, mean RMSE, mean MAE and mean MAPE respectively. As mentioned previously, the correlative load series in multiple historical cycles usually have similar external factors, which can reflect the power consumption patterns of households under certain conditions. In other words, repeating and frequent activities would be included in the input data when using correlative load series in multiple cycles. Therefore, robust prediction results can be achieved when the input data includes more reliable information.

The UK-DALE dataset is another dataset used to evaluate the effectiveness of the proposed method. Table 7 summarizes the evaluation results of the Kalman filter-based model and the MultiCycleNet on the UK-DALE dataset.

TABLE 7

HOUSE-LOAD FORECASTING ACCURACY ON THE HOUSE AND APPLIANCE LEVELS ON UK-DALE DATASET

SMAPE (%)	Kalman filter-based model [3]	MultiCycleNet
Strategy 1: forecasting the aggregated	28.2	**6.8**
Strategy 2: aggregating the forecasts	15.1	**6.2**

There are two strategies: (1) the conventional strategy, and (2) the bottom-up strategy, used in [3] to evaluate the difference on household load forecasting accuracy on the house and appliance levels. The Strategy 1 forecasts the load directly at the household level, and the Strategy 2 aggregates the forecasts made at the appliance level.

Generally, the information recorded under different granularities usually has greater differences. Fine-grained information, such as load profile in half an hour, is usually more detailed, and coarse-grained information, such as load profile in days, is usually rough. Considering that fine-grained information is more suitable for the proposed method to analyze the correlative information in the historical cycles, therefore, in this study, the MultiCycleNet first predicted the household electricity consumption for half an hour, and then aggregated the forecast results during the day to form the finally daily forecast results for Strategy 1. On the other hand, considering that home-level electricity consumption information is also important for daily forecasting, for Strategy 2, the home-level electricity consumption data was also used for the daily forecasting in addition to the appliance-level electricity consumption data. The results in Table 7 shows that the MultiCycleNet outperforms the Kalman filter-based model by 75.9% and 58.9% in terms of SMAPE for Strategy 1 and Strategy 2, respectively. On the other hand, the MultiCycleNet shows better performance on Strategy 2 than Strategy 1 due to the increased electricity consumption information of appliances. To predict the next day electricity consumption, the Kalman filter-based model used the last 10 days of historical data, while the MultiCycleNet first extracted correlative data from multiple cycles, and then used these data to predict household future load. Therefore, the MultiCycleNet can learn the household electricity consumption pattern better from historical data and obtain more accurate prediction results.

It is also worth mentioning that the patterns of power consumption at different times of a day are still quite different, for example, the power consumption patterns in the morning and in the evening would be significantly different, which can be interpreted by the daily activities. To this end, it seems inappropriate to select the correlative load series over a relatively long period, as it may contain different power consumption patterns with the load to be predicted.

For an in-depth study, state-of-the-art methods having better predicted results in Table 6 (LSTM* and PDRNN) are selected to further compare their performance with MultiCycleNet in the case of high fluctuation of the load profiles. In this case, the fluctuation is determined by the standard deviation of the load profiles. The high fluctuation of load profiles is usually a huge challenge in the field of household load forecasting as it will bring great problems in the accurate prediction. Table 8 shows the MSE values of the top 5 users with the most fluctuating load profiles in the dataset. Fig. 7 shows the above users true and predicted load profiles in three days for visualization.

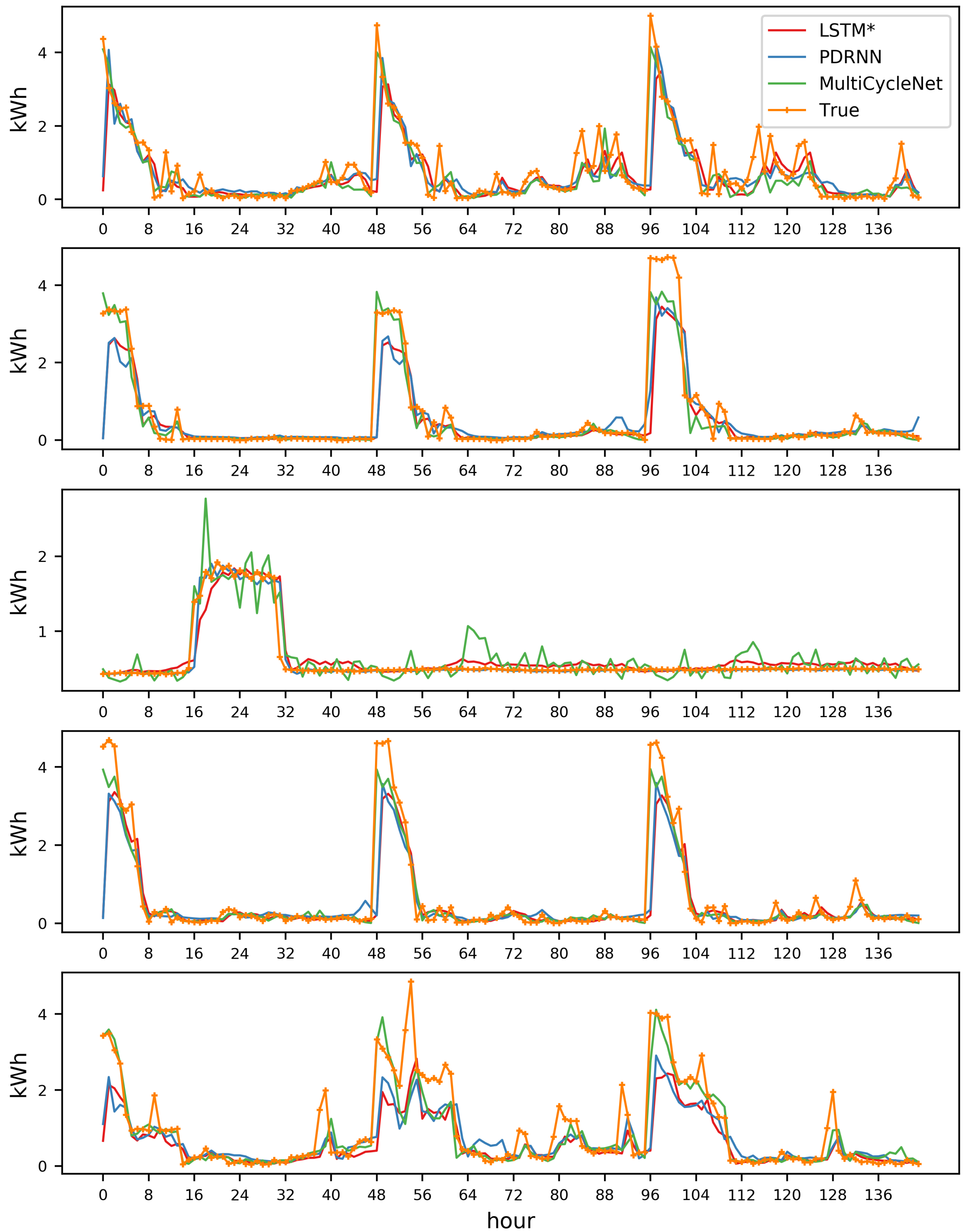

Fig. 7. The load profiles of the top 5 users having the most fluctuation in the dataset, including the true value and the predicted value of each model.

TABLE 8

THE MSE VALUES OF THE TOP 5 USERS WITH THE MOST FLUCTUATING LOAD PROFILES

Model	Profile 1	Profile 2	Profile 3	Profile 4	Profile 5
LSTM*	0.399	0.384	0.259	0.484	0.568
PDRNN	0.389	0.289	**0.146**	0.461	0.487
MultiCycleNet	**0.164**	**0.080**	0.181	**0.073**	**0.113**

The results in Table 8 show that the proposed method achieves better predicted performance than the comparative methods on four out of top five users with the most fluctuating load profiles, which indicates that the proposed method is able to learn power consumption pattern better than the comparative methods in the case of high fluctuation. However, the MultiCycleNet does not achieve the best performance in profile 3, which can be interpreted by the effect of the invalid information stored in the previous cycles. The proposed method would use data in multiple historical cycles for learning. If too much invalid information appears in multiple historical cycles, the learning of the model would be affected, and therefore inaccurate predictions would be made. In other words, the MultiCycleNet needs to take some time to adapt to the customer's new patterns, while forgetting all the invalid information it learned earlier.

4.6.1 Analysis of the residual signals

Fig. 8 illustrates the results of residual signal analysis of the proposed method. Fig. 8 (a) shows the autocorrelation of the residuals. It shows that there is a peak at the lag 1 order, however, in the case of other lag orders, the autocorrelation is not large.

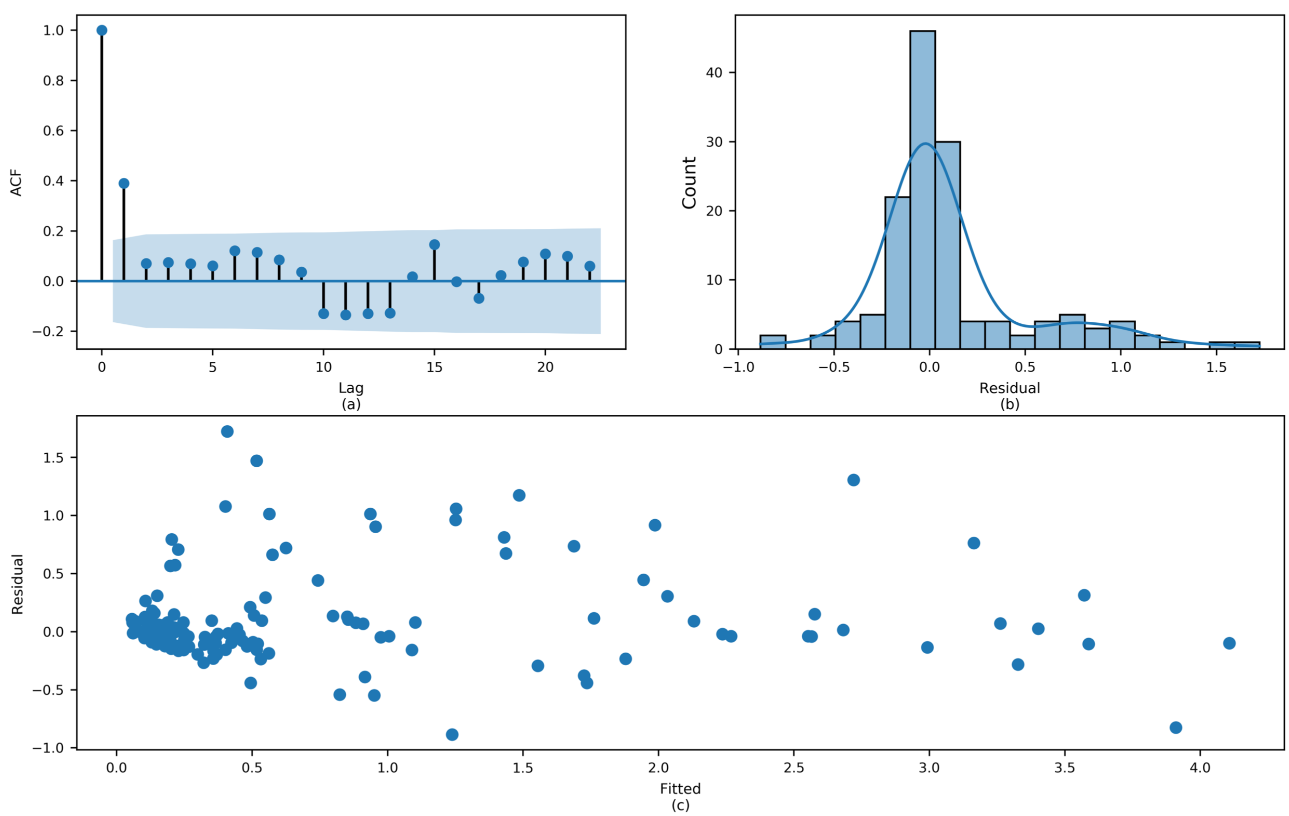

Fig. 8. Analysis of the residual signals for the MultiCycleNet. (a) Image of autocorrelation function of residuals. (b) Histogram of residuals. (c) Relationship between fitted values and residuals.

On the other hand, although the right tail is slightly longer as shown in Fig. 8 (b), the distribution of the residuals conforms to the normal distribution. Fig. 8 (c) shows the relationship between the fitted values and the residuals. The figure shows that there is no obvious rule between the fitted values and the residuals, and the residuals are centered at 0, showing a symmetrical form. Therefore, it can be inferred from the results in Fig. 8 that the difference between the predicted values and the observed values is random and unpredictable. In other words, there is no interpretable and predictable information in the predicted error of the proposed method.

4.6.2 Statistical test of models

In order to test whether the results of the experiment are affected by accidental factors, we used the paired sample t-test to evaluate the differences of the models. The paired sample t-test is used to determine whether the mean difference between the test error rates of the two models is zero. If the predictive performance of the two models is the same, then their test error rates should also be the same.

TABLE 9

PAIRED SAMPLE T-TESTS OF MODELS

Models	p-value
LSTM, MultiCycleNet	0.0036
LSTM*, MultiCycleNet	0.0013
PDRNN, MultiCycleNet	0.0037

Table 9 summarizes the results of the paired sample t-test for models, and it illustrates that all p-values of $p < 0.05$ between the proposed model and LSTM, LSTM* and PDRNN, which means that the results are significant. Therefore, there is a significant difference in the predictive performance between the models that we have tested, and the model with the smaller average error rate has better performance.

5. Conclusion

This paper presents a novel framework for household load forecasting, namely, MultiCycleNet framework, which aims to address the uncertainty of household load forecasting. The proposed framework is based on the idea that load series with high correlation in multiple historical cycles often reflect the common pattern of time series. The level of household power consumption is usually stable in a short period. Therefore, we consider using these correlative load series distributed in multiple cycles to improve the accuracy of household load forecasting.

The proposed framework is introduced and discussed in detail in this paper. First, we explain the preprocessing method of load profiles in order to accelerate the convergence rate of the model and obtain the stable prediction results. Then, the concept of correlative load series in multiple cycles is discussed. The correlative load series consist of two levels, namely, the time level and the cycle level. Furthermore, we give the key processes about how to generate the correlative load series in multiple cycles. After that, an introduction about the inference module and the technique used in the proposed framework is given. We evaluate the effectiveness of the proposed framework on two real-life datasets based in the United Kingdom. The experiment results show that the proposed framework can be a competitive approach among the state-of-the-art methods in several performance metrics. And the statistical test of models shows that there is a significant difference in the predictive performance between the proposed model and the comparative methods. For future work, different weights can be assigned to different historical load series, aiming to study how to take advantage of customers' own historical load profiles better to solve the uncertainty of household load forecasting.

Acknowledgement

This work is supported by National Natural Science Foundation of China under Grants 61876066 and 61572201, Guangzhou Science and Technology Plan Project 201804010245, Department of Finance and Education of Guangdong Province 2016 [202]: Key Discipline Construction Program, China; the Education Department of Guangdong Province: New and Integrated Energy System Theory and Technology Research Group [Project Number 2016KCXTD022]; Brunel University London BRIEF Funding, UK.

References

[1] M. A. Ahad, S. Paiva, G. Tripathi, and N. Feroz, “Enabling technologies and sustainable smart cities,” *Sustainable Cities and Society*, vol. 61, p.102301, 2020.

[2] C. S. Lai, L. L. Lai, and Q. H. Lai, Smart grids and big data analytics for smart cities, Springer International Publishing: New York, NY, USA, October 2020, 491 pages.

[3] Z. Zheng, H. Chen, and X. Luo, “A Kalman filter-based bottom-up approach for household short-term load forecast,” *Applied Energy*, vol.250, pp. 882–894, 2019.

[4] H. Shi, M. Xu, and R. Li, “Deep learning for household load forecasting—A novel pooling deep RNN,” *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5271–5280, 2018.

[5] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[6] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient kNN classification with different numbers of nearest neighbors," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1774-1785, May 2018.

[7] S. Hochreiter, and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, “Short-term residential load forecasting based on LSTM recurrent neural network,” *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841–851, 2019.

[9] W. Kong, Z. Dong, D. Hill, F. Luo, and Y. Xu, "Short-term residential load forecasting based on resident behavior learning," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 1087-1088, 2018.

[10] K. Bandara, C. Bergmeir, and H. Hewamalage, "LSTM-MSNet: Leveraging forecasts on sets of related time series with multiple seasonal patterns," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1-14, 2020.

[11] F. Quilumba, W. Lee, H. Huang, D. Wang, and R. Szabados, "Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 911-918, 2015.

[12] M. Schuster, and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673-2681, Nov. 1997.

[13] Z. Dong, D. Yang, T. Reindl, and W. M. Walsh, “Short-term solar irradiance forecasting using exponential smoothing state space model,” *Energy*, vol. 55, pp. 1104–1113, 2013.

[14] M. Khashei, and M. Bijari, “A novel hybridization of artificial neural networks and arima models for time series forecasting,” *Applied Soft Computing*, vol. 11, no. 2, pp. 2664–2675, 2011.

[15] X. Sun, P. B. Luh, K. W. Cheung, W. Guan, L. D. Michel, S. Venkata, and M. T. Miller, “An efficient approach to short-term load forecasting at the distribution level,” *IEEE Transactions on Power Systems*, vol. 31, no. 4, pp. 2526–2537, 2015.

[16] J. Nowotarski, B. Liu, R. Weron, and T. Hong, “Improving short term load forecast accuracy via combining sister forecasts,” *Energy*, vol. 98, pp. 40–49, 2016.

[17] Y. Wang, Q. Chen, M. Sun, C. Kang, and Q. Xia, "An ensemble forecasting method for the aggregated load with sub profiles," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3906-3908, 2018.

[18] X. Wang, W. Lee, H. Huang, R. Szabados, D. Wang, and P. Van Olinda, "Factors that impact the accuracy of clustering-based load forecasting," *IEEE Transactions on Industry Applications*, vol. 52, no. 5, pp. 3625-3630, 2016.

[19] S. Li, P. Wang, and L. Goel, "A novel wavelet-based ensemble method for short-term load forecasting with hybrid neural networks and feature selection," *IEEE Transactions on Power Systems*, vol. 31, no. 3, pp. 1788-1798, 2016.

[20] G. B. Huang, Q. Y. Zhu, and C. K. Siew, “Extreme learning machine: Theory and applications,” *Neurocomputing*, vol. 70, pp. 489–501, 2006.

[21] K. Y. Chan, T. S. Dillon, J. Singh, and E. Chang, "Neural-Network-Based Models for Short-Term Traffic Flow Forecasting Using a Hybrid Exponential Smoothing and Levenberg–Marquardt Algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 644-654, June 2012.

[22] M. Leung, C. Norman, L. L. Lai, and T. T. Chow, “The use of occupancy space electrical power demand in building cooling load prediction,” *Energy and Buildings*, vol. 55, pp. 151–163, 2012.

[23] X. Qiu, Y. Ren, P. N. Suganthan, and G. A. Amaratunga, “Empirical mode decomposition based ensemble deep learning for load demand time series forecasting,” *Applied Soft Computing*, vol. 54, pp. 246–255, 2017.

[24] X. Yang, J. Yuan, J. Yuan, and H. Mao, “An improved wm method based on pso for electric load forecasting,” *Expert Systems with Applications*, vol. 37, no. 12, pp. 8036–8041, 2010.

[25] J. Kennedy, and R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1995, pp. 1942-1948 vol.4.

[26] Li-Xin Wang, "The WM method completed: a flexible fuzzy system approach to data mining," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 6, pp. 768-782, Dec. 2003.

[27] I. P. Panapakidis, “Application of hybrid computational intelligence models in short-term bus load forecasting,” *Expert Systems with Applications*, vol. 54, pp. 105–120, 2016.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904-1916, 1 Sept. 2015.

[29] B. Zhang, D. Xiong, and J. Su, “Neural machine translation with deep attention,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 1, pp. 154–163, 2018.

[30] P. Agrawal, and S. Ganapathy, “Modulation filter learning using deep variational networks for robust speech recognition,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 244–253, 2019.

[31] X. Ma, and Y. Dong, “An estimating combination method for interval forecasting of electrical load time series,” *Expert Systems with Applications*, p. 113498, 2020.

[32] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis,” Deep learning for computer vision: A brief review,” *Computational Intelligence and Neuroscience*, vol. 2018, 2018.

[33] D. C. Wu, B. B. Asl, A. Razban, and J. Chen, “Air compressor load forecasting using artificial neural network,” *Expert Systems with Applications*, p. 114209, 2020.

[34] C. S. Lai, Y. Yang, K. Pan, J. Zhang, H. Yuan, W. W. Y. Ng, Y. Gao,Z. Zhao, T. Wang, M. Shahidehpour, and L. L. Lai, “Multi-view neural network ensemble for short and mid-term load forecasting,” *IEEE Transactions on Power Systems*, vol. 36, no. 4, pp. 2992–3003, 2021.

[35] T. Ouyang, Y. He, H. Li, Z. Sun, and S. Baek, “Modeling and forecasting short-term power load with copula model and deep belief network,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 3, no. 2, pp. 127–136, 2019.

[36] Z. Chang, Y. Zhang, and W. Chen, “Electricity price prediction based on hybrid model of adam optimized lstm neural network and wavelet transform,” *Energy*, vol. 187, p. 115804, 2019.

[37] D. Kingma, and J. Ba, “Adam: A Method for Stochastic Optimization,” *International Conference on Learning Representations*, May. 2015.

[38] G. Sideratos, A. Ikonomopoulos, and N. D. Hatziargyriou, “A novel fuzzy-based ensemble model for load forecasting using hybrid deep neural networks,” *Electric Power Systems Research*, vol. 178, p. 106025, 2020.

[39] J. Moody and C. J. Darken, “Fast learning in networks of locally-tuned processing units,” *Neural Computation*, vol. 1, no. 2, pp. 281–294, 1989.

[40] H. Chen, S. Wang, S. Wang, and Y. Li, "Day-ahead aggregated load forecasting based on two-terminal sparse coding and deep neural network fusion," *Electric Power Systems Research*, vol. 177, p. 105987, 2019.

[41] M. Duan, A. Darvishan, R. Mohammaditab, K. Wakil, and O. Abedinia, “A novel hybrid prediction model for aggregated loads of buildings by considering the electric vehicles,” *Sustainable Cities and Society*, vol. 41, pp. 205–219, 2018.

[42] E. O.-N. Jnr, Y. Y. Ziggah, and S. Relvas, “Hybrid ensemble intelligent model based on wavelet transform, swarm intelligence an artificial neural network for electricity demand forecasting,” *Sustainable Cities and Society*, vol. 66, p. 102679, 2021.

[43] M. Q. Raza, M. Nadarajah, D. Q. Hung, and Z. Baharudin, “An intelligent hybrid short-term load forecasting model for smart power grids,” *Sustainable Cities and Society*, vol. 31, pp. 264–275, 2017.

[44] C. Dinesh, S. Makonin, and I. Bajic, "Residential power forecasting using load identification and graph spectral clustering," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 11, pp. 1900-1904, 2019.

[45] F. Y. Xu, X. Cun, M. Yan, H. Yuan, Y. Wang, and L. L. Lai, "Power market load forecasting on neural network with beneficial correlated regularization," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 5050-5059, Nov. 2018.

[46] E. Caro, J. Juan, and J. Cara, “Periodically correlated models for short term electricity load forecasting,” *Applied Mathematics and Computation*, vol. 364, p. 124642, 2020.

[47] K. Bandara, C. Bergmeir, and S. Smyl, “Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach,” *Expert Systems with Applications*, vol. 140, p. 112896, 2020.

[48] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp. 85–117, 2015.

[49] https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households

[50] Kelly J, and W. Knottenbelt. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. Sci Data 2015;2:150007. https://doi.org/10.1038/sdata.2015.7.

[51] A. Jierula, S. Wang, T.-M. Oh, and P. Wang, “Study on accuracy metrics for evaluating the predictions of damage locations in deep piles using artificial neural networks with acoustic emission data,” *Applied Sciences*, vol. 11, no. 5, p. 2314, 2021.

[52] J. Bergstra, and Y. Bengio, “Random search for hyper-parameter optimization,” *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 281–305, 2012.

[53] C. W. Tsai, C. H. Hsia, S. J. Yang, S. J. Liu, and Z. Y. Fang, “Optimizing hyperparameters of deep learning in predicting bus passengers based on simulated annealing,” *Applied Soft Computing*, vol. 88, p. 106068, 2020.

[54] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *Advances in Neural Information Processing Systems*, Dec. 2014.

[55] S. Bai, J. Z. Kolter, and V. Koltun, “Convolutional sequence modeling revisited,” *International Conference on Learning Representations*, Apr. 2018.

[56] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, “Lightgbm: A highly efficient gradient boosting decision tree,” *Advances in Neural Information Processing Systems*, vol. 30, pp. 3146–3154, 2017