IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 2, 2020, accepted March 14, 2020, date of publication March 18, 2020, date of current version March 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2981817

A Deep Learning Method for Short-Term Residential Load Forecasting in Smart Grid

YE HONG[1], YINGJIE ZHOU[1], (Member, IEEE), QIBIN LI[2], WENZHENG XU[1], (Member, IEEE), AND XIUJUAN ZHENG[3]

[1]College of Computer Science, Sichuan University, Chengdu 610065, China
[2]Key Laboratory of Low-Grade Energy Utilization Technologies and Systems, Ministry of Education, College of Energy and Power Engineering, Chongqing University, Chongqing 400044, China
[3]College of Electrical Engineering, Sichuan University, Chengdu 610065, China

Corresponding authors: Yingjie Zhou (yjzhou@scu.edu.cn) and Qibin Li (qibinli@cqu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61801315 and Grant 51876015.

ABSTRACT Residential demand response is vital for the efficiency of power system. It has attracted much attention from both academic and industry in recent years. Accurate short-term load forecasting is a fundamental task for demand response. While short-term forecasting for aggregated load data has been extensively studied, load forecasting for individual residential users is still challenging due to the dynamic and stochastic characteristic of single users' electricity consumption behaviors, i.e., the variability of the residential activities. To address this challenge, this paper presents a short-term residential load forecasting framework, which makes use of the spatio-temporal correlation existing in appliances' load data through deep learning. Multiple time series are conducted in the framework to describe electricity consumption behaviors and their internal spatio-temporal relationship. And a method based on deep neural network and iterative ResBlock is proposed to learn the correlation among different electricity consumption behaviors for short-term load forecasting. Experiments based on real world measurements have been conducted to evaluate the performance of the proposed forecasting approach. The results show that both the appliances' load data and iterative ResBlocks can help to improve the forecasting performance. Compared with existing methods, measurements on Root Mean Squared Error, Mean Absolute Error and Mean Absolute Percentage Error for the proposed approach are reduced by 3.89%-20.00%, 2.18%-22.58% and 0.69%-32.78%. In addition, further experiments are conducted to evaluate the impact of using appliances' load data, iterative ResBlocks as well as other factors for the proposed approach.

INDEX TERMS Smart grid, short-term load forecasting, deep learning, residential load forecasting, iterative ResBlocks.

I. INTRODUCTION

Energy is the one of the main drivers of human activity. Demand response is of crucial importance for maintaining the reliable and efficient operation of the smart grid system [1]–[4]. In residential power distribution, demand response can manage the power delivery from power system to users and smooth the system load. Short-term load forecasting (STLF) predicts users' demand in the near future, which provides the key information for making decisions in residential demand response. On one hand, STLF can help to satisfy users' electricity demand and reduce the risk of outages. If the result of STLF indicates that users' electricity demand will exceed the capacity of system in a residential community, the power company can incentivize residential users to shift their power consumption by hiking up the electricity price [5]. On the other hand, STLF can help to benefit the power company and residential users in economic aspects. Based on the results of STLF, the power company can calculate optimal pricing strategies for residential electricity usage [3]. For residential users, they can take appropriate countermeasures if the STLF results indicate that there will be a power shortage [6].

Existing STLF methods could be categorized into two kinds: aggregated load forecasting [7]–[9] and individual users' load forecasting [10]. Both of the forecasting methods construct features based on historical records of load since the value of short-term electricity consumption is related with

The associate editor coordinating the review of this manuscript and approving it for publication was Ahmed F. Zobaa.

previous values. The aggregated load forecasting gives the estimation of the total electricity consumption for a group of users in a specific area, such as a city or a residential community.

Short-term forecasting for aggregated load data has been extensively studied. Time series analysis has been applied on the STLF problem, including Auto Regressive Moving Average (ARMA) based method [11]–[13], [37], [42] and Support Vector Regression (SVR) based method [14], [15], [45]. Wei and Zhengang [11] combined the ARMA based method and other statistical methods to improve the performance for aggregated load forecasting. Pappas *et al.* [12] developed an offline model with enhanced the ARMA based method to predict the power usage provided by a power company in Greece. Huang and Shih [13] proposed a method that could improve the forecasting accuracy through a modified ARMA based method with non-Gaussian process. Amini *et al.* [40] apply an ARIMA based model for electric vehicles' demand forecasting. The method decouples conventional electrical load and charging demand of EV to forecast them independently, which can reduce forecasting errors. Yang *et al.* [14] presented an SVR based method to forecast power consumption at a city scale. They developed a grid search approach to automatically tune the model parameters, which can reduce the difficulty in the parameter optimization phase. Velasco *et al.* [15] presented a load forecasting model based on the SVR method for country-wide power usage. Ren et.al [45] employed an ensembled method for STLF, which consists of SVR, random forest and Xgboost. Other methods [16]–[20] take extra conditions into account to improve the forecasting results.

Load forecasting for individual users gives the estimation of the total electricity consumption of an individual user, e.g., a resident. However, this problem is still challenging. A major reason is that electricity consumption behaviors of single user is stochastic. The stochasticity is introduced by the uncertainty of the time that electricity consumption activities occurred [21]. Another reason for the difficult of individual users' load forecasting is the consumption of electricity usage is dynamic, even for a specific application of the same user.

In order to provide supporting accurate information for residential demand response, STLF for individual users attracts increasing interests recently. Some methods [22]–[24] apply clustering techniques to obtain the groups of users that have similar consumption behaviors. Teeraratkul *et al.* [22] proposed a shape based clustering method for STLF. Based on the profile of load curves, the method uses dynamic time warping technique to cluster the load curve and find a canonical shape for each set of curves. Then, a Markov model is used to conduct the individual users' load forecasting. Quilumba *et al.* [23] grouped individual users based on similar consumption behaviors, which are represented by users' load data. Based on the clustering results, a neural network employing weather and calendar features is used in the prediction phase.

More recently, researchers explored using deep learning techniques to perform STLF for individual users, due to its ability to extract the latent features of users' electricity consumption behaviors and less domain knowledge requirements compared to traditional methods. Classical deep learning models are used for STLF, such as Deep Neural Network (DNN) [43], Extreme Learning Machine [36]. Kong *et al.* [10] proposed a Long Short-Term Memory RNN (LSTM-RNN) based framework for residential STLF. Experiment results showed that their method outperforms traditional machine learning methods. Shi *et al.* [21] proposed a pooling based deep RNN. The pooling stage uses load data of neighbors to generate new features of inputs, which increases the data volume and helps to solve the over-fitting problem.

Although the above methods made progresses in some aspects, the historical data they employed are the overall load data of single resident, which cannot include the spatio-temporal correlation existing among appliances' load data. The spatio-temporal correlation mentioned here is the spatio correlation among electricity consumption behaviors of different kind of appliances and the temporal correlation between the historical electricity consumption behaviors and future electricity consumption behaviors. For a single user, the spatio correlation exists in the user's electricity consumption behaviors of different appliances. For instance, household members may have daily routines that using washing machine after taking a shower, or opening refrigerator before making a meal. The temporal correlation is the similarity of the historical electricity consumption behaviors and the future consumption behaviors. More specifically, the time of an electricity consumption behavior (e.g., electricity usage of washing machine) happened in the future are probably close to the time that the same behavior happened in the past. These correlations exist inside the load profile, and is significant for individual users' load forecasting.

Consequently, some researchers employed the load data of different appliances for load forecasting. These researches employed different methods to explore the correlation among different appliances. Dinesh *et al.* [41] used nonintrusive appliance load monitoring techniques [26] to collect the appliances' load and forecast household load based on graph spectral clustering. Mohi Ud Din *et al.* [42] applied the appliances-level load as the input features of neural network structure to forecast short-term load, and PCA technique is employed for feature reduction.

In this paper, we explore using the spatio-temporal correlation among different kinds of electricity consumption behaviors to improve the performance of STLF. We present a framework of STLF for individual users based on the correlation information. Multiple time series are conducted in the framework to describe electricity consumption behaviors for different applications and their internal spatio-temporal relationship. And a method based on Deep Neural Network and iterative ResBlocks is proposed to learn the correlation among consumption behaviors for STLF. ResBlock that consists of few stacked layers and one skip connection is based

TABLE 1. Nomenclature.

Symbol/ Abbreviation	
Symbols	
X	input of the IRBDNN model
Y	output of the IRBDNN model
T	number of iterations
Θ	weights, biases associated with the model
W_t	the linear projection
$E_i(t)$	load data for the appliance i in time interval t
$E_0(t)$	the overall load in time interval t
Abbreviations	
STLF	Short Term Load Forecasting
DNN	Deep Neural Network
RNN	Recurrent Neural Network
ARMA	Auto Regressive Moving Average
IRBDNN	Iterative Resblocks Based Deep Neural Network
ELM	Extreme Learning Machine
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error

on the building block for ResNet [30]. The proposed method based on iterative ResBlocks is able to learn both shallow and deep features of the input vectors, which makes use of the correlation information among different kinds of electricity consumption behaviors. The contributions of this paper are summarized as follows:

1. We designed a framework of STLF for individual users, which improves the performance of STLF by using the spatio-temporal correlation among appliances' load data.

2. We proposed a method based on DNN with iterative ResBlocks to learn the spatio-temporal correlation among different kinds of electricity consumption behaviors in the framework. To the best of our knowledge, this is the first work using iterative ResBlocks to learn the latent features of electricity consumption behaviors for the short-term residential load forecasting problem.

3. Experiments using real world data were conducted to evaluate the performance of the proposed approach.

Table 1 illustrates all symbols and abbreviations used in this paper. The rest of this paper is organized as follows. The designed framework and the proposed Iterative Resblocks Based Deep Neural Network (IRBDNN) model are presented in Section 2 and 3. In Section 4, we evaluate the performance of the proposed forecasting approach. And in Section 5 is the conclusion of this paper.

II. PRORPOSED FRAMEWORK

In this section, we present the proposed STLF framework and the IRBDNN based method. Figure 1 is the schematic overview of the proposed STLF framework. As shown in Figure 1, the proposed STLF framework consists of four modules, including data acquisition, data preprocessing, model training and load forecasting. The data acquisition module collects measurements from household smart meters, which report electricity consumption data of appliances for each consumer. The spatio-temporal correlation is included in the appliances' consumption data, which is the output of the data acquisition module. Data cleaning, data integration and data transformation are conducted in the data preprocessing module to improve the data quality for the input of STLF model. In the model training module, a deep learning model based on deep neural network and iterative ResBlock is designed to learn the spatio-temporal correlation among different electricity consumption behaviors. Also, a parameter optimization step is included in this module to further enhance the learning ability of the proposed method. After the preprocessing steps and the model training procedure, the proposed model is able to calculate the predicted values for an individual user. These components work together to form a suitable solution for residential STLF.

A. DATA ACQUISITION MODULE

As shown in the Figure 2, we would like to employ appliances' load for residential STLF. This module's output is the appliances' load data that contains the spatio-temporal correlation among different kinds of electricity consumption behaviors. There are two ways to obtain the appliances' load for residential STLF. The first one is to install load monitoring infrastructures for each appliance, which can report the electricity usage of the appliances. This way requires extra equipment, which will increase the cost of power system. The other way is to use nonintrusive appliance load monitoring techniques, which applies disaggregation algorithms to decompose the entire household load into appliances' load data [26] or separates certain appliance's load from the entire household power consumption [25]. The algorithms [26], [27] describe the characteristics of the appliances based on high frequency load data (1 Hz or even higher). A typical disaggregation algorithm is sparse coding [28], [29], which has high temporal resolution ability to separate different applications. At the end of this module, appliances' load that contains the spatio-temporal correlation are obtained.

B. DATA PREPROCESSING MODULE

Usually, the load data acquired from the data acquisition module are not suitable for the input of forecasting model directly. This is because the original data may have missing values, various formats and high computational needs in the real world. The residential load data need to be preprocessed and transformed into a suitable form for the STLF model.

The first step to preprocess the residential load is data cleaning. In the real world, the problem of missing values is usually the most popular problem, which may be caused by hardware failures. We consider two ways to handle this

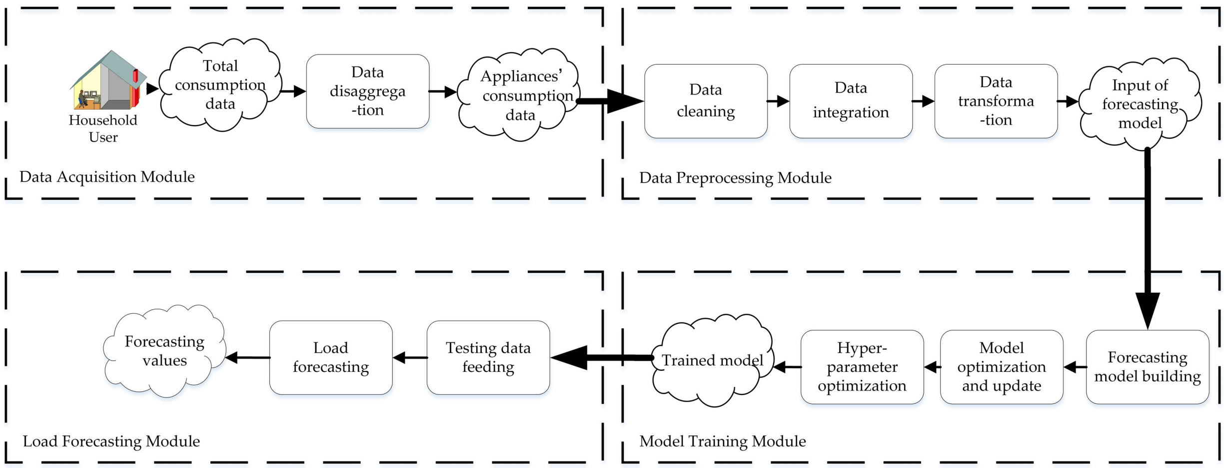

FIGURE 1. Schematic overview of the short-term load forecasting framework.

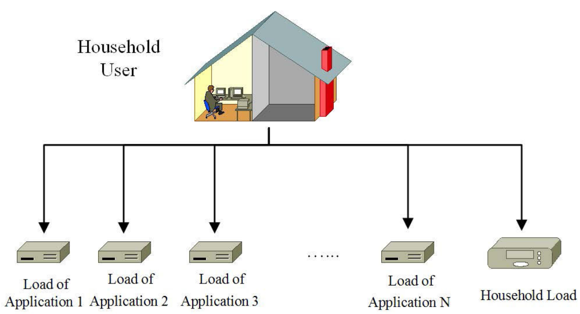

FIGURE 2. Disaggregation of household electricity usage.

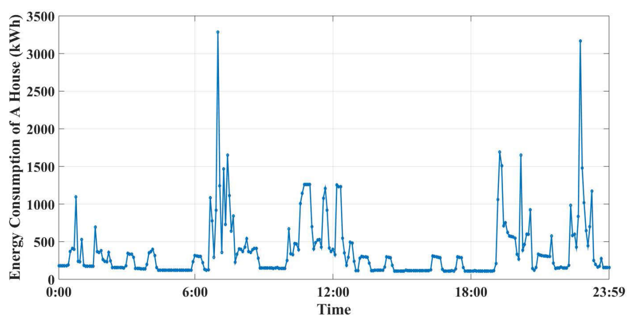

FIGURE 3. An example of daily energy consumption for a house in the greater Boston area of the U.S.

problem in the residential load. 1) If the length of the missing values' duration is acceptable, the missing values can be estimated according to the values before and after the missing durations and the values in other days in the corresponding time. 2) If the length of the missing values' duration is too long that is not acceptable, the load data of this day will be ignored in the model training phase.

The next step is data integration. When appliances' load monitoring infrastructures report electricity usage, the recording frequency of each monitor may be different. Therefore, data integration needs to be conducted to provide a uniform format for the input of forecasting model. One of the data integration approaches is calculating the sum for load data in a duration to form a dataset that has lower recording frequency than the original dataset. For instance, in order to get time series which records 1 value each minute, load data of appliances need to be integrated by calculating the sum of load data in every minute. Also, normalization needs to be conducted in each appliance's load data, which aims to equal the influence of different kinds of appliances' load data on the forecasting results.

The recording frequency of load monitoring infrastructures are usually at a second level, such as one measurement every three seconds. However, the second level load data is not suitable for STLF due to the following reasons. Firstly, load forecasting at the second level is impossible due to the dynamic and stochastic characteristic of electricity consumption behaviors. Secondly, there is too much noise in the load records at a second level, which may increase the training difficulties of forecasting models. Therefore, it is necessary to conduct the data transformation step, which aims to obtain a simplified representation of the original dataset. The recording frequency of the processed data set after the data transformation step will be lower than the original dataset. For example, the recording frequency will be transformed into one measurement each hour from one measurement every three seconds. An example of daily energy consumption for a house in the greater Boston area of the U.S. is illustrated in Figure 3.

C. MODEL TRAINING MODULE AND FORECASTING MODULE

The forecasting model should learn the spatio-temporal correlation among different kinds of electricity consumption behaviors. The model is expected to have the ability that captures the characteristics among appliances' load in both space and time. In the model training module, a specific forecasting model that can learn the spatio-temporal correlation needs to be built. Then the model needs to be optimized and updated to obtain an effective solution. After these steps, the training of the forecasting model is completed.

The last module of the framework is the load forecasting module. Based on the model output in the model training module and the historical records, a predicted load value will be calculated this step.

III. PROPOSED MODEL

A. THE IRBDNN MODEL

In this subsection, we introduce the proposed the IRBDNN model, which implements the forecasting model building step in the model training module of Figure 1. The proposed model employs iterative ResBlocks in a DNN and the model is able to learn the spatio-temporal correlation among different kinds of user's electricity consumption behaviors. The input of the IRBDNN model includes both the spatio correlation and the temporal correlation of a residential user, in terms of residential user's electricity consumption behaviors of different appliances and historical electricity consumption behaviors, respectively. The structure of the IRBDNN model enables the model to learn both the deep features and the shallow features from the input. Moreover, the structure of iterative ResBlocks enables the learning ability of the deep IRBDNN model is no worse than the learning ability of a shallow IRBDNN model. Detailed analysis for the learning ability of the proposed model is discussed in Section IV. C, which shows that the employment of iterative ResBlocks can help to improve the forecasting performance.

The proposed model is employed to learn the nonlinear relationship between the input features and the output value. Generally, the increase of model depth strengthens the learning ability of the neural network. However, the performance of deep learning model may degrade in practice. The possible reasons for the degraded performance may be the intrinsic characteristic of data or the optimization difficulties of deep learning model. He *et al.* [30] proposed a method using ResBlocks to obtain a better performance than compared methods only using stacked hidden layers. Inspired by their work, we propose a model based on DNN and iterative ResBlocks for the STLF task. The iterative ResBlocks are able to learn the spatio-temporal correlation in the STLF task and ensure the learning ability of the model.

ResBlock is a structure that is different to stacked layers. It is similar to the building block for ResNet [30] that is widely used in the image classification problem, but with a twist.

The input and output of the skip connection in a ResBlock can be in different dimensions, while in the building block for ResNet they are usually the same. The basic structure of ResBlock consists of two components, i.e., few stacked layers and one skip connection. Stacked layers consist of few hidden layers and adjacent layers are directly connected. A structure of two stacked layers are shown in Figure 4(a). The input of the stacked layers is x and the output of the

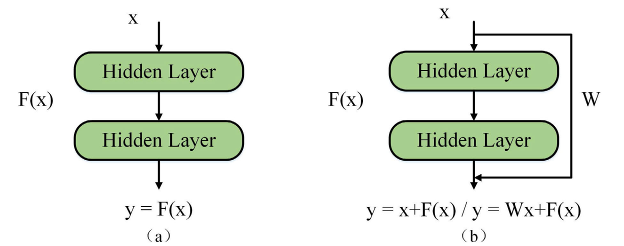

FIGURE 4. Comparison of stacked layers and ResBlock. (a) Structure of 2 stacked layers; (b) Structure of a ResBlock.

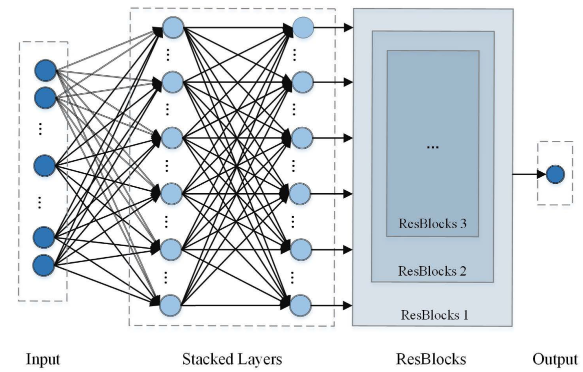

FIGURE 5. An illustration of the IRBDNN structure.

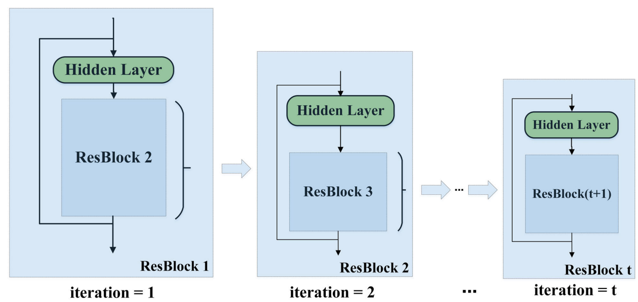

FIGURE 6. The iterative ResBlocks in the IRBDNN.

stacked layers is $y = F(x)$. Figure 4(b) shows the structure of a ResBlock, which consists of two stacked layers and one skip connection. When the input and the output of the skip connections have the same dimension, the skip connection is a typical identical mapping. Therefore, the output of the corresponding ResBlock is $y = F(x) + x$. When the input and output of the skip connection are in different dimensions, the skip connection performs as a linear projection to match the changes of dimensions. The output of the corresponding ResBlock is $y = F(x) + Wx$, where W is a linear projection. When the stacked layers and the ResBlocks contain the same number of hidden layers, the skip connection ensures that the learning ability of the ResBlock is no worse than the learning ability of the stacked layers.

The IRBDNN model is constructed by the stacked layers and the iterative ResBlocks. The structure is shown in Figure 5, and Figure 6 illustrates the insight for the structure of the iterative ResBlocks. We denote the number of iterations as t. When $t = 0$, the ResBlocks module degenerates into a non-iterative structure, and the IRBDNN model degenerates into a DNN model. When $t = 1$, the input of the first ResBlock (denoted as *ResBlock* 1) is added to the output of the *ResBlock* 1 by a skip connection. The *ResBlock* 1 consists of three parts: m stacked layers, the *ResBlock* 2 and a skip connection. When $t = t_0$, the iterations are repeated t_0 times. The ResBlock t_0 consists of m stacked layers, ResBlock $(t_0 + 1)$ and a skip connection. If iteration t_0 is the last iteration, ResBlock $(t_0 + 1)$ degenerate into n stacked layers ($m > 0, n > 0$).

For each ResBlock in the IRBDNN model, the input of ResBlock is linked to the output directly by skip connection, which ensures the learning capability of the current ResBlock with deeper embedded ResBlock is no worse than the learning capability of the ResBlock without deeper embedded ResBlock. The structure enables the model to make full use of the spatio-temporal correlation among the different consumption behaviors.

As it is described above, when the number of iterations is 0, the structure of the IRBDNN model degenerates into a structure of DNN. When the number of iterations is no less than 1, the iteration procedures of the IRBDNN model are presented as below:

$$y = F(x_1, \Theta_1) + W_0(x_0) \tag{1}$$

$$F(x_1, \Theta_1) = F(x_2, \Theta_2) + W_1(x_1) \tag{2}$$

$$\cdots$$

$$F(x_t, \Theta_t) = F(x_{t+1}, \Theta_{t+1}) + W_t(x_t) \tag{3}$$

where $F(x_{t+1}, \Theta_{t+1})$ is the output of $(m + n)$ staked layers in the ResBlock $(t + 1)$ with the input x_t, $(t + 1)$ is the number of iterations, y is the output of the IRBDNN model, Θ denotes weights, biases associated with the model and W_t denotes the linear projection to match the possible changes of dimensions.

B. INPUT OF THE IRBDNN MODEL

In this subsection, we construct multiple time series based on appliances' load data to form the input of the IRBDNN model. The time series consist of a number of load values and each load value represents the energy consumption for a duration. We use "time interval" to denote a duration in the rest of this paper. The preprocessed load data for the appliance i in time interval t is represented as $E_i(t)$, and the overall load in time interval t is denoted as $E_0(t)$. The sequence for forecasting the overall load $E_0(t_0 + 1)$ is shown as below:

$$\begin{aligned} X(t) = \{&E_1(t_0 - T + 1), E_1(t_0 - T + 2), \ldots, E_1(t_0), \ldots, \\ &E_i(t_0 - T + 1), E_i(t_0 - T + 2), \ldots, E_i(t_0), \ldots, \\ &E_0(t_0 - T + 1), E_0(t_0 - T + 2), \ldots, E_0(t_0)\} \end{aligned} \tag{4}$$

where $X(t)$ is the input vector consisting of two parts, including the appliances' load data and the overall load; the length of the historical records for forecasting is denoted as T.

C. SEQUENTIAL GRID SEARCH METHOD FOR HYPER-PARAMETER OPTIMIZATION

This subsection introduces the hyper-parameters optimization method for the proposed the IRBDNN model. As it is described in Section 3.1, the IRBDNN architecture consists of the stacked layers and the iterative ResBlocks. The number of hidden neurons in each layer of IRBDNN model are the same, which is denoted as N. The weight matrix that connects the hidden neurons in $(l - 1)$th layer and the neurons in l th layer is W^l, which is a $N \times N$ matrix. b^l is a vector that contains N elements which represents the bias of the hidden neurons in lth layer. The output of $(l - 1)$th layer is denoted as a^{l-1}. Then the output of layer lth is $a^l = \sigma(W^l a^{l-1} + b^l)$, where σ denotes the activation function.

The loss function applied in the model is given by (5):

$$Loss = \sqrt{\frac{1}{N} \sum (t(n) - p(n))^2} \tag{5}$$

where $t(n)$ and $p(n)$ denote the truth load data and the predicted load data in the n-th time interval, and N is the number of predicted time intervals in the training set.

In deep learning methods, numbers of hyper-parameters need to be optimized. An exhaustive grid search for all hyper-parameters is time-consuming. To address this problem, we design a sequential grid search approach to optimize hyper-parameters for the proposed the IRBDNN model, which is inspired by Ismail *et al.* work [31]. In this paper, the following hyper-parameters are optimized. The firsthyper-parameter is the number of neuron N in each layer, searched in (100, 150, 200, 300, 400, 450, 500). The second hyper-parameter is the learning rate LR, searched in (0.001, 0.0001, 0.00001). And the last one is the initializer I that can be used for the IRBDNN parameters, searched in (Normal, Uniform, Glorot Normal, Glorot Uniform). The sequential grid search process is illustrated in Figure 7 and Figure 8. The Algorithm 1 described in Figure 7 is the main program and the Algorithm 2 described in Figure 8 is its subprogram. The sequential grid search can be divided into three parts. In the first part, the hyper-parameters are initialized to build the initial forecasting model. N and LR have significant impact on the learning ability of the forecasting model. Therefore, we optimize N and LR synchronously to define Model 1 in the second part. I is set an initialized value first, then it is optimized in the third part. After the above process, Model 2 is defined with the optimal N, LR and I.

IV. RESULTS

To evaluate the performance of the proposed framework and the IRBDNN based method, we compare the forecasting performance of the proposed method with existing algorithms including the ARMA based method, the ELM based method and the SRX [42] based method. The above experiments are conducted based on the Redd dataset [32].

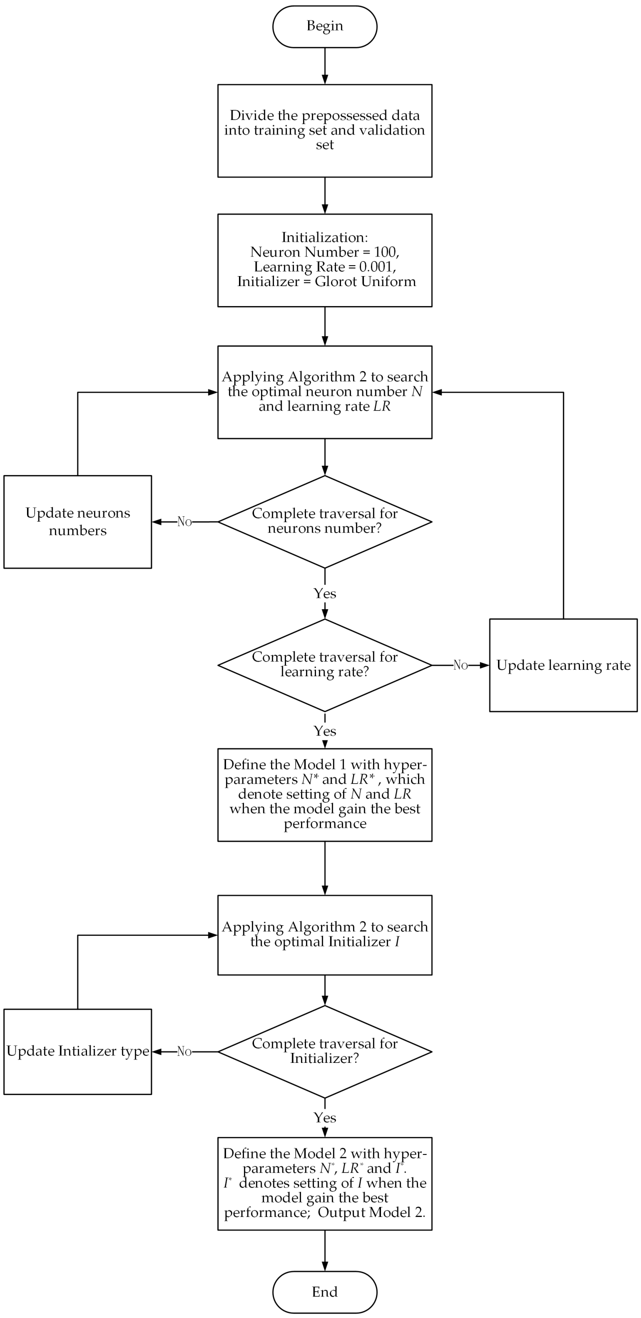

FIGURE 7. The flow diagram for the main program of the sequential grid search approach.

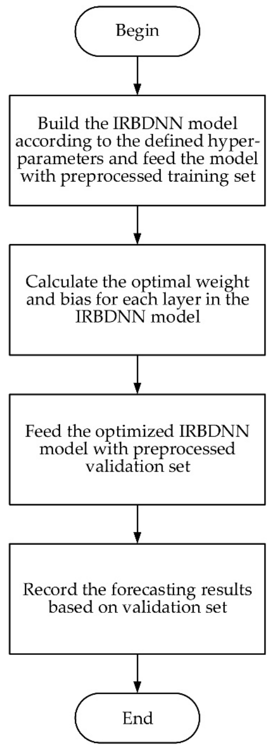

FIGURE 8. The flow diagram for the subprogram of the sequential grid search approach.

A. DATASETS

In this section, we introduce the Redd dataset and the pre-processing steps for it. The Redd dataset is a public available data set, which records the consumption data of appliances in residential users' houses from March, 2011 to July, 2011. In our experiments, we employ the data that the recording frequency is every three seconds for each record.

Data preprocessing for the Redd dataset includes data cleaning, data integration and data transformation. The data cleaning step is conducted according to the description in the Section 2.2. After the data cleaning steps, we sum up the appliances' load data in each 30 minutes to get a time series that consist of 48 values each day, then apply the min-max normalization for the time series.

The Redd dataset that we use contains ten appliances (oven, refrigerator, dishwasher, kitchen outlet, washer dryer, bedroom, lighting, electric heat, microwave and stove). We obtained 32 days' load data for one user and separate them into three subsets: 20 days' data, 4 days' data and 7 days' data for the training set, the validation set and the testing set, respectively. Table 2 shows the descriptive statistics of the overall load data in the three subsets, including total size, mean value, maximum value, minimum value and standard deviation.

An example of the daily load for all appliances is illustrated in Figure 9. Bars with different colors illustrate the appliances' load and a red line presents the overall load. As shown in the figure, the time series of appliances' load are quite dynamic. For example, the load record of oven is higher than 2500kWh around 7:30, while the load data is lower than 1500kWh around 20:00. Moreover, the spatial correlation among appliances is obvious in the figure. For instance, the oven and the microwave are used together around 7:30 and around 20:00. Also, lighting' load record value will increase when the oven and microwave are used.

B. EXPERIMENT RESULTS

In the experiments, we use Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) as the metrics of performance evaluation.

TABLE 2. Descriptive statistics of the three subsets.

Descriptive Statistics	Training Set	Validation Set	Testing Set
Total size	25 Days	4 Days	7 Days
Mean(kWh)	301.84	325.95	288.79
Maximum(kWh)	3406.64	3116.00	2603.17
Minimum(kWh)	55.82	56.99	55.53
Standard deviation	384.94	454.74	345.14

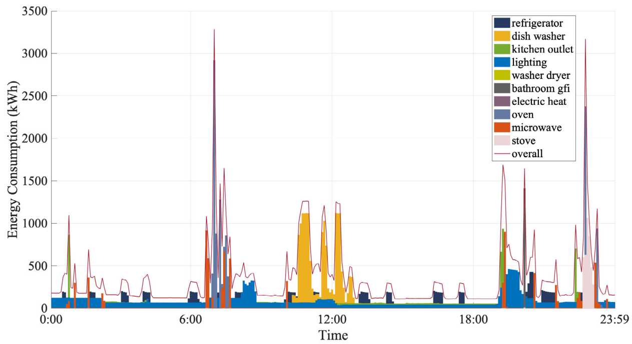

FIGURE 9. An example of energy consumption over a day for one user in the Redd dataset.

RMSE is root mean square error, which is sensitive to errors and suitable to evaluate the accuracy of forecasting results [33]. *MAE* is mean absolute error, which is a useful metric widely used in model evaluation [34]. *MAPE* is mean absolute percentage error between the predicted value and the truth value, which is able to avoid the offset problem of errors [35]. The definitions of the metrics are as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum(y-p)^2} \tag{6}$$

$$MAE = \frac{1}{m}\sum|(y-p)| \tag{7}$$

$$MAPE = \frac{1}{m}\sum\frac{|(y-p)|}{y} * 100\% \tag{8}$$

where m is the number of values that represent the load data for a duration of 30 minutes in the testing set, y is the truth value and p is the predicted value. We implemented an 8 layers IRBDNN with 3 iterations. The number of the stacked layers in each ResBlock is 1 layer, 1 layer, and 2 layers, respectively. The sequential grid search algorithm is applied in the hyper-parameter tuning step, and the search ranges of hyper-parameters are presented in Table 3.

In order to evaluate the effectiveness of our framework and the proposed IRBDNN based method, we conducted two groups of experiments. In the first group, the experiments employ both the appliances' load data and the overall load. The DNN based method, the SRX based method, and the proposed IRBDNN based method are conducted in this group. The experiments in another group includes the ELM based method and the ARMA based method, which only employ the overall load. For the ELM based method, the search range of the hidden neurons is the same with the IRBDNN based method. And for the DNN based method, the number of hidden layers is searched in range (3, 4, 6, 8), and the search ranges of other hyper-parameters are the same as that of the IRBDNN based method described in Table 2.

TABLE 3. Range of hyper-parameters for grid search.

Hyper-parameter	Range for Grid Search
Numbers of Neurons: N	100, 150, 200, 300, 400, 450, 500
Learning Rate: LR	0.001, 0.0001, 0.00001
Initializer: I	Normal, Uniform, Glorot Normal, Glorot Uniform

TABLE 4. The comparison for results of forecasting methods.

Day	Metric	IRBDNN	DNN	SRX	ARMA	ELM
Day 1	RMSE	131.457	130.99	125.08	143.23	186.37
	MAE	91.2	90.26	89.66	95.90	150.27
	MAPE	0.6446	0.6427	0.5196	0.5619	1.088
Day 2	RMSE	**221.14**	228.26	280.19	265.65	261.12
	MAE	**126.61**	133.90	171.12	177.38	169.40
	MAPE	**0.5119**	0.5853	0.6772	0.8524	0.8427
Day 3	RMSE	**137.70**	140.52	115.75	150.48	166.49
	MAE	**98.20**	101.09	82.93	104.69	116.49
	MAPE	**0.7267**	0.7276	0.5530	0.6887	1.004
Day 4	RMSE	170.87	167.85	169.50	191.66	163.11
	MAE	118.11	113.19	107.35	125.23	122.23
	MAPE	0.6014	0.5676	0.5068	0.6725	0.7180
Day 5	RMSE	**122.91**	123.03	123.58	136.12	164.95
	MAE	**89.36**	89.63	92.04	104.24	136.12
	MAPE	**0.5169**	0.5171	0.5341	0.6043	0.8368
Day 6	RMSE	**461.83**	477.39	525.32	566.69	549.02
	MAE	**229.15**	235.94	263.22	313.87	303.44
	MAPE	**0.4664**	0.5009	0.4937	0.7891	0.7695
Day 7	RMSE	**443.24**	477.50	510.89	593.56	509.63
	MAE	**245.79**	256.72	289.04	355.43	291.65
	MAPE	**0.8436**	0.8667	1.057	1.434	1.018
Overall	RMSE	**277.54**	288.78	313.62	346.91	326.36
	MAE	**142.64**	145.82	156.48	182.39	184.23
	MAPE	**0.6159**	0.6283	0.6202	0.8003	0.8968

Table 4 shows the forecasting results for each day in the testing set and the forecasting results for the whole testing set. It can be observed that the IRBDNN based method performs the best and the second best on Day 2, Day 3, Day 5 Day 6 and Day 7. The last row illustrates the overall results of five methods. The IRBDNN based method outperforms other methods, and the DNN based method performs the second best, better than the SRX based method, the ELM based method and the ARMA based method. We can observe from the table that the IRBDNN based method performs better than other methods on the three metrics, which indicates that the proposed method using deep neural network and

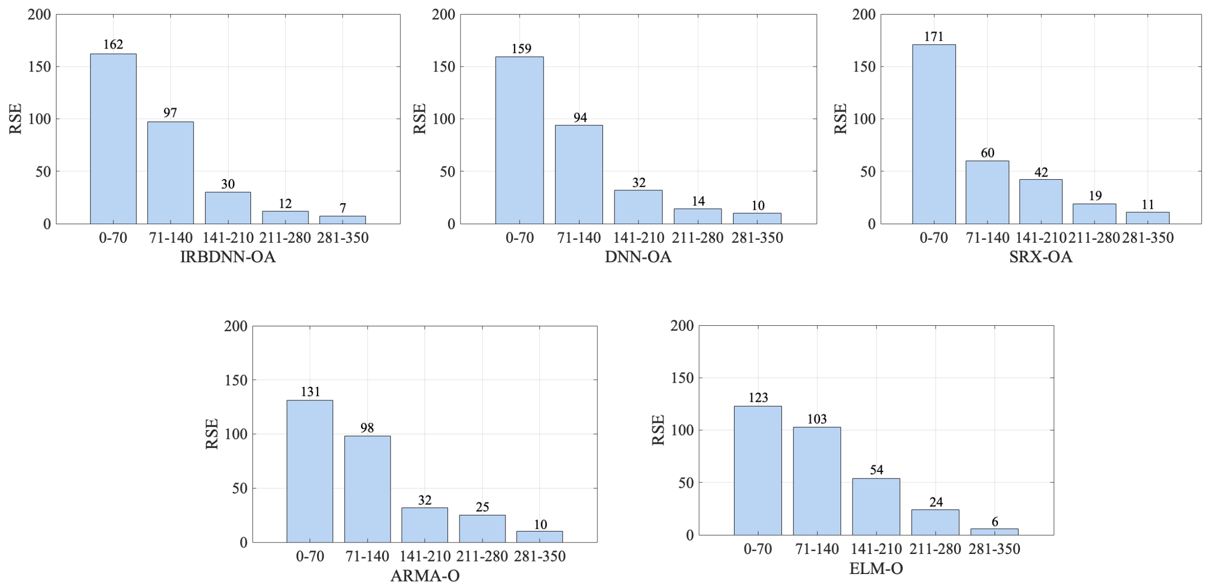

FIGURE 10. Error distributions of the results of the forecasting methods. (The suffix 'AO' indicates that the model employs both appliances' load and the overall load data, and the suffix 'O' indicates that the model only employs the overall load data).

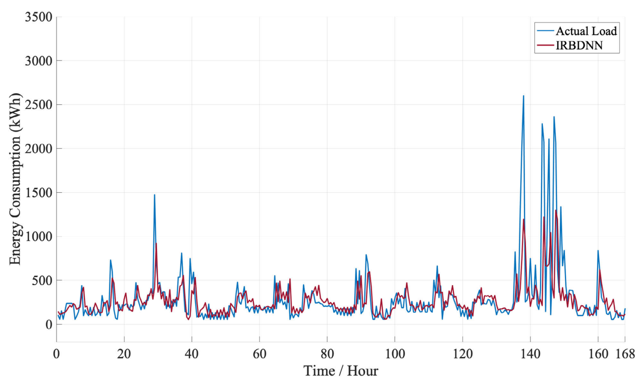

FIGURE 11. Forecasting Performance of the proposed method for a week.

iterative ResBlock can effectively learn the spatio-temporal correlation among consumption behaviors. The forecasting results of a residential user using the proposed method as well as the actual load are illustrated in Figure 11.

Figure 10 is the error distributions of the results of the forecasting methods. We can observe from the figure that the error distributions of the IRBDNN-OA, DNN-OA and SRX-OA perform better than ELM-O and ARMA-O that only employ the overall load data. More specifically, the number of RSEs between 0 and 140 (the first two bars) in the IRBDNN-OA's forecasting result is over 255, which is more than DNN-OA, SRX-OA, ARMA-O and ELM-O. In other words, the number of RSEs larger than 140 in the IRBDNN-OA's forecasting results is the fewest among the forecasting result of the five methods, because the RSE is calculated for each predicted value and the number of RSEs for each method is equal. The number of RSEs larger than 210 in the forecasting result of the IRBDNN-OA is around 25, while the number of RSEs for the other methods that is larger than 210 is larger than 25. The IRBDNN based method's forecasting results have less extreme large RSEs (>210) than other methods and has smaller RSEs (0-140) than other methods. Considering the overall performance in Table 4, we can draw the conclusion that the IRBDNN based method generally performs better than other methods.

C. DISCUSSION

To obtain a more comprehensive understanding for the performance of the proposed approach, four additional sets of experiments are conducted. The training data, validation data and the testing data in these experiments are the same as the data used in the Section IV, B.

1) PERFORMANCE ANALYSIS OF SPATIO-TEMPORAL CORRELATION AND ITERATIVE ResBlocks

In the first set, three groups of experiments are conducted to verify the impact of spatio-temporal correlation among different appliances and iterative ResBlocks. The three groups are the IRBDNN group, the DNN group and the SRX group. Each group includes two cases with different historical data: 1) the overall load data; 2) the overall load data and the appliances' load data. The experiments results are shown in Figure 12.

The results from Figure 12 show that both the appliances' load data and iterative ResBlocks can help to improve the forecasting performance. Detailed analysis is presented as follows. Firstly, comparing two experiments in each group, the case employs both appliances' load and the overall load performs better than the case only employs the overall load data. We can conclude that the employment of the appliances'

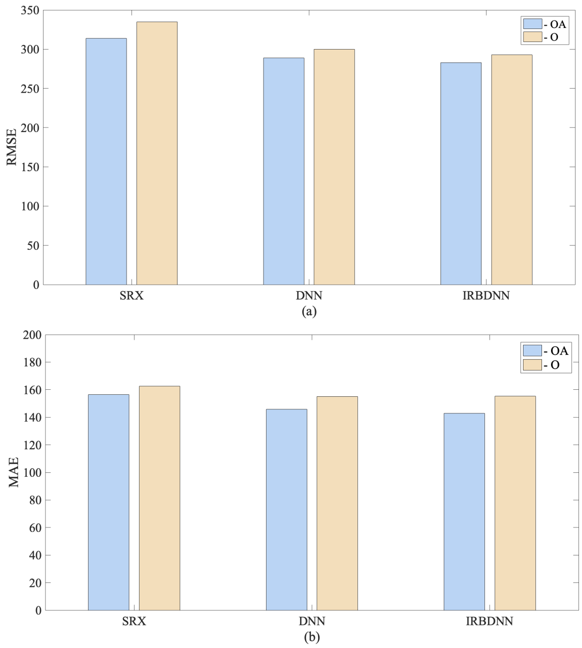

FIGURE 12. Impact of employing the appliances' load data. (a) RMSE; (b) MAE.

TABLE 5. Details for the network structure of the IRBDNN model with different number of iterations.

Num_of_Iterations	Num_of_Layers	Num_of_Skip_Connections
Iterations = 1	3	1
Iterations = 2	4	2
Iterations = 3	6	3
Iterations = 4	8	4

load data can help to improve the forecasting performance. Secondly, the case 'IRBDNN-AO' performs better than the case 'DNN-AO' and the case 'SRX-AO'. The same trend can be observed from the cases that only employ the overall load. Case 'IRBDNN-O' performs better than other case 'DNN-O' and case 'SRX-O'. These results indicate that the employment of iterative ResBlocks can better capture the spatio-temporal correlation among different appliances than compared methods and help to improve the forecasting performance. However, the improvement of forecasting performance is at the cost of computation efficiency. We further analysis the cost of computation time in the fourth part of this subsection.

2) IMPACT OF ITERATION TIMES ON THE IRBDNN BASED METHOD

In the second set of experiments, we test the IRBDNN models with 0 iteration, 1 iteration, 2 iterations and 3 iterations, respectively. Table 5 shows the details for the network structure of the IRBDNN model with different number of iterations. The sequential grid search method is used to

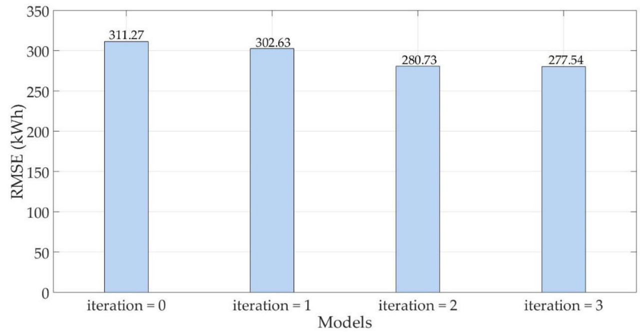

FIGURE 13. RMSE of the IRBDNN model with different number of iterations.

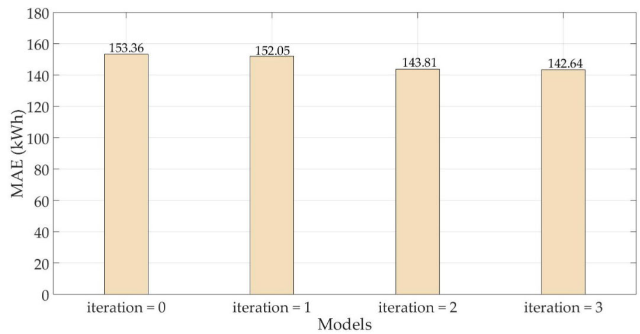

FIGURE 14. MAE of the IRBDNN model with different number of iterations.

optimal the hyper-parameters, and the search ranges of the hyper-parameters are illustrated in Table 3. Figure 13 and Figure 14 illustrate the RMSE and MAE of the experiment results.

From Figure 13, we can observe that the IRBDNN model with 3 iterations performs the best among all models. The RMSE decreases when the number of iterations (i.e., the number of ResBlocks) increases. The same trend can be observed from the MAE in Figure 14. There is no experiment of the IRBDNN model with 4 iterations because the performance of the IRBDNN model with 3 iterations is close to the performance of the IRBDNN model with 2 iterations, which only improves 1.14% on RMSE and 0.81% on MAE. When the number of iterations continuously increases, the improvement on the RMSE and MAE will not be obvious. The results indicate that the IRBDNN based method intends to have improved performances with the increment of the iteration number.

3) IMPACT OF HIDDEN NEURON NUMBERS ON THE IRBDNN BASED METHOD

In the third group of experiments, we explore the influence of the hidden neuron numbers for the 8-layer IRBDNN model with 3 iterations, and the results are illustrated in Figure 15 and Figure 16.

According to the results in Figure 15 and Figure 16, we can observe that optimal hidden neuron number is 300. The changes on RMSE and MAE for different number of hidden

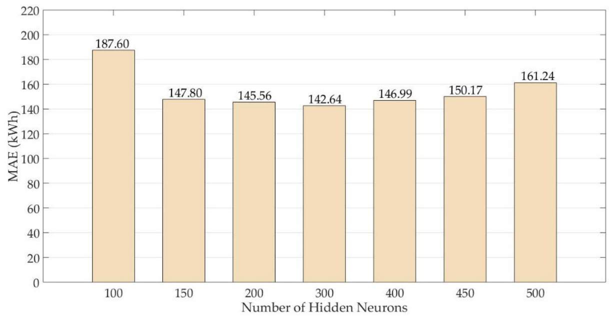

FIGURE 15. Influence on MAE of hidden neuron number.

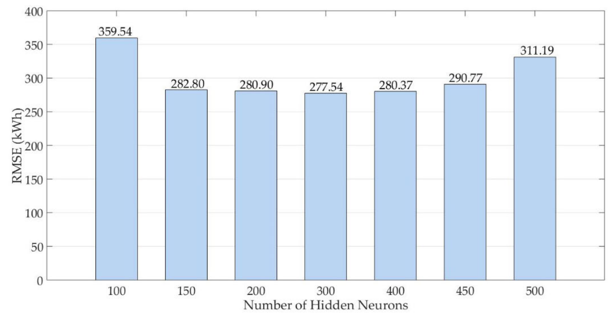

FIGURE 16. Influence on RMSE of hidden neuron number.

neurons are similar. Both of them decrease when the hidden neuron number increases from 100 to 300, while both of them increase when the number of hidden neurons is larger than 300.

4) ANALYSIS OF COMPUTATION TIME ON THE IRBDNN BASED METHOD

In order to analysis the computational cost, we compared the training time and the testing time of the IRBDNN based method and the based DNN method, since both of them are implemented on neural network. The experiments are conducted on a personal computer equipped with a 2.5GHz Intel i5 Core Processer and 8GB RAM. We compared the training time and the testing time for both methods with two types of historical data, i.e., 1) only the overall load (denoted as input type ‘O’); 2) both the overall load and the appliances’ load (denoted as input type ‘OA’).

Table 6 shows the computational cost for both methods. The training time is recorded by running 200 epochs for the whole training set. And the testing time is recorded by the whole testing set, which has 7 day’s data. We can observe from the table that the employment of the appliance’s load increases the computation cost for the training process and the testing process for both methods, which indicates that the improvement of the forecasting performance by employing the appliances’ load is at the cost of computation time. Also, we can observe from the table that the employment of iterative ResBlocks does not evidently increase the computation cost of the IRBDNN based method that adopting skip connections in the network structure. Thus, although there is additional

TABLE 6. Running time for the IRBDNN models.

Forecasting Model	Input Type	Training Time / s (200 Epochs)	Testing Time / s (on 7 days' data)
IRBDNN	OA	101.783816	0.005257
	O	43.915002	0.002217
DNN	OA	102.69109	0.005436
	O	45.450106	0.002196

computation cost on the proposed method, the overall training time as well as the overall testing time are acceptable for the power company since they are relatively short and the training process is usually off-line.

V. CONCLUSION

In this paper, we explored using the spatio-temporal correlation among different kinds of appliances to predict the short-term electricity demand for individual residential users.

An effective STLF framework that includes the data acquisition module, the data preprocessing module, the model training module and the load forecasting module was proposed. Multiple time series were conducted in the framework to describe electricity consumption behaviors and their internal spatio-temporal relationship. In order to fully exploit the correlation of user behaviors and characteristics of users’ consumption patterns, a method based on DNN and iterative ResBlocks was proposed to learn the correlation. A grid search method was employed in the hyper-parameter optimization phase. The proposed method and several existing forecasting methods were evaluated on a real world dataset.

The results show that the IRBDNN based method outperforms other compared methods. Moreover, we demonstrated that both the appliances’ load data and iterative ResBlocks can help to improve the forecasting performance. In addition, experiment results indicate that the IRBDNN based method intends to have improved performances with the increment of the iteration number. In the future work, we intend to employ the correlation defined in communication networks [38], [39] to express the spatio-temporal correlation among different residential users to improve the performance of STLF. Also, we will explore predicting thermophysical properties of matter [44].

REFERENCES

[1] H. J. Monfared, A. Ghasemi, A. Loni, and M. Marzband, “A hybrid price-based demand response program for the residential micro-grid,” *Energy*, vol. 185, pp. 274–285, Oct. 2019.

[2] J. A. Gomez-Herrera and M. F. Anjos, “Optimal collaborative demand-response planner for smart residential buildings,” *Energy*, vol. 161, pp. 370–380, Oct. 2018.

[3] M. H. Albadi and E. F. El-Saadany, “A summary of demand response in electricity markets,” *Electric Power Syst. Res.*, vol. 78, no. 11, pp. 1989–1996, Nov. 2008.

[4] W. Yuan, J. Huang, and Y. J. A. Zhang, “Competitive charging station pricing for plug-in electric vehicles,” *IEEE Trans. Smart Grid*, vol. 8, no. 2, pp. 627–639, Dec. 2017.

[5] M. Muratori and G. Rizzoni, "Residential demand response: Dynamic energy management and time-varying electricity pricing," *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 1108–1117, Mar. 2016.

[6] T. Chen, "A collaborative fuzzy-neural approach for long-term load forecasting in taiwan," *Comput. Ind. Eng.*, vol. 63, no. 3, pp. 663–670, Nov. 2012.

[7] L. Han, Y. Peng, Y. Li, B. Yong, Q. Zhou, and L. Shu, "Enhanced deep networks for short-term and medium-term load forecasting," *IEEE Access*, vol. 7, pp. 4045–4055, 2019.

[8] Z. Yu, Z. Niu, and W. Tang, "Deep learning for daily peak load forecasting—A novel gated recurrent neural network combining dynamic time warping," *IEEE Access*, vol. 7, pp. 17184–17194, 2019.

[9] M. R. Haq and Z. Ni, "A new hybrid model for short-term electricity load forecasting," *IEEE Access*, vol. 7, pp. 125413–125423, 2019.

[10] 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 Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan. 2019.

[11] L. Wei and Z. Zhen-gang, "Based on time sequence of ARIMA model in the application of short-term electricity load forecasting," in *Proc. Int. Conf. Res. Challenges Comput. Sci.*, Dec. 2009, pp. 11–14.

[12] S. S. Pappas, L. Ekonomou, D. C. Karamousantas, G. E. Chatzarakis, S. K. Katsikas, and P. Liatsis, "Electricity demand loads modeling using AutoRegressive moving average (ARMA) models," *Energy*, vol. 33, no. 9, pp. 1353–1360, Sep. 2008.

[13] S.-J. Huang and K.-R. Shih, "Short-term load forecasting via ARMA model identification including non-Gaussian process considerations," *IEEE Trans. Power Syst.*, vol. 18, no. 2, pp. 673–679, May 2003.

[14] Y. Yang, J. Che, C. Deng, and L. Li, "Sequential grid approach based support vector regression for short-term electric load forecasting," *Appl. Energy*, vol. 238, pp. 1010–1021, Mar. 2019.

[15] L. Clark, D. Lou, D. Michelle, G. T. Alegata, and G. C. Luna, "Day-ahead load forecasting using support vector regression machines," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 3, pp. 22–27, 2018.

[16] L. Hernández, C. Baladrón, J. M. Aguiar, B. Carro, A. Sánchez-Esguevillas, and J. Lloret, "Artificial neural networks for short-term load forecasting in microgrids environment," *Energy*, vol. 75, pp. 252–264, Oct. 2014.

[17] X. Cao, S. Dong, Z. Wu, and Y. Jing, "A data-driven hybrid optimization model for short-term residential load forecasting," in *Proc. IEEE Int. Conf. Comput. Inf. Technol., Ubiquitous Comput. Commun., Dependable, Autonomic Secure Comput., Pervasive Intell. Comput.*, Oct. 2015, pp. 283–287.

[18] Y. Liang, D. Niu, and W.-C. Hong, "Short term load forecasting based on feature extraction and improved general regression neural network model," *Energy*, vol. 166, pp. 653–663, Jan. 2019.

[19] M. Tucci, E. Crisostomi, G. Giunta, and M. Raugi, "A multi-objective method for short-term load forecasting in European countries," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3537–3547, Sep. 2016.

[20] K. M. Powell, A. Sriprasad, W. J. Cole, and T. F. Edgar, "Heating, cooling, and electrical load forecasting for a large-scale district energy system," *Energy*, vol. 74, pp. 877–885, Sep. 2014.

[21] H. Shi, M. Xu, and R. Li, "Deep learning for household load forecasting—A novel pooling deep RNN," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5271–5280, Mar. 2017.

[22] T. Teeraratkul, D. O'Neill, and S. Lall, "Shape-based approach to household electric load curve clustering and prediction," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5196–5206, Sep. 2017.

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

[24] J. D. Rhodes, W. J. Cole, C. R. Upshaw, T. F. Edgar, and M. E. Webber, "Clustering analysis of residential electricity demand profiles," *Appl. Energy*, vol. 135, pp. 461–471, Dec. 2014.

[25] K. X. Perez, W. J. Cole, J. D. Rhodes, A. Ondeck, M. Webber, M. Baldea, and T. F. Edgar, "Nonintrusive disaggregation of residential air-conditioning loads from sub-hourly smart meter data," *Energy Buildings*, vol. 81, pp. 316–325, Oct. 2014.

[26] G. W. Hart, "Nonintrusive appliance load monitoring," *Proc. IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[27] J. Z. Kolter, S. Batra, and A. Y. Ng, "Energy disaggregation via discriminative sparse coding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1153–1161.

[28] C.-N. Yu, P. Mirowski, and T. K. Ho, "A sparse coding approach to household electricity demand forecasting in smart grids," *IEEE Trans. Smart Grid*, vol. 8, no. 2, pp. 738–748, Jan. 2016.

[29] Y. Wang, Q. Chen, C. Kang, Q. Xia, and M. Luo, "Sparse and redundant representation-based smart meter data compression and pattern extraction," *IEEE Trans. Power Syst.*, vol. 32, no. 3, pp. 2142–2151, May 2016.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[31] M. Ismail, M. Shahin, M. F. Shaaban, E. Serpedin, and K. Qaraqe, "Efficient detection of electricity theft cyber attacks in AMI networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2018, pp. 1–6.

[32] J. Z. Kolter and M. J. Johnson, "REDD: A public data set for energy disaggregation research," in *Proc. Sustkdd*, vol. 25, 2011, pp. 59–62.

[33] X. Gao, X. Li, B. Zhao, W. Ji, X. Jing, and Y. He, "Short-term electricity load forecasting model based on EMD-GRU with feature selection," *Energies*, vol. 12, no. 6, p. 1140, 2019.

[34] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)," *Geoscientific Model Develop. Discuss.*, vol. 7, no. 1, pp. 1525–1534, 2014.

[35] S. Kim, G. Lee, G.-Y. Kwon, D.-I. Kim, and Y.-J. Shin, "Deep learning based on multi-decomposition for short-term load forecasting," *Energies*, vol. 11, no. 12, p. 3433, 2018.

[36] G. Huang, S. Song, J. N. D. Gupta, and C. Wu, "Semi-supervised and unsupervised extreme learning machines," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2405–2417, Dec. 2014.

[37] P. J. Brockwell, R. A. Davis, and M. V. Calder, *Introduction to Time Series and Forecasting*, vol. 2. New York, NY, USA: Springer, 2002. [Online]. Available: https://link.springer.com/book/10.1007/978-1-4757-2526-1#authorsandaffiliationsbook

[38] Z. Qin, Y. Wang, H. Cheng, Y. Zhou, Z. Sheng, and V. C. M. Leung, "Demographic information prediction: A portrait of smartphone application users," *IEEE Trans. Emerg. Topics Comput.*, vol. 6, no. 3, pp. 432–444, Jul. 2018.

[39] H. Huang, H. Yin, G. Min, H. Jiang, J. Zhang, and Y. Wu, "Data-driven information plane in software-defined networking," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 218–224, 2017.

[40] M. H. Amini, A. Kargarian, and O. Karabasoglu, "ARIMA-based decoupled time series forecasting of electric vehicle charging demand for stochastic power system operation," *Electr. Power Syst. Res.*, vol. 140, pp. 378–390, Nov. 2016.

[41] C. Dinesh, S. Makonin, and I. V. Bajic, "Residential power forecasting using load identification and graph spectral clustering," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 66, no. 11, pp. 1900–1904, Nov. 2019.

[42] G. Mohi Ud Din, A. U. Mauthe, and A. K. Marnerides, "Appliance-level short-term load forecasting using deep neural networks," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Maui, HI, USA, Mar. 2018, pp. 53–57.

[43] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec. 2017.

[44] Y. Zhou, Q. Li, and Q. Wang, "Energy storage analysis of UIO-66 and water mixed nanofluids: An experimental and theoretical study," *Energies*, vol. 12, no. 13, p. 2521, 2019.

[45] L. Ren, L. Zhang, H. Wang, and L. Qi, "An ensemble model based on machine learning methods for short-term power load forecasting," in *Proc. IOP Conf. Ser., Earth Environ. Sci.*, vol. 186, Oct. 2018, Art. no. 012042.

YE HONG received the B.Sc. degree in computer science from Sichuan University, China, in 2017, where she is currently pursuing the master's degree in computer science. Her research interests include behavioral data analysis and deep learning applications.

YINGJIE ZHOU (Member, IEEE) received the Ph.D. degree from the School of Communication and Information Engineering, University of Electronic Science and Technology of China (UESTC), China, in 2013. He was a Visiting Scholar with the Department of Electrical Engineering, Columbia University, New York. He is currently an Assistant Professor with the College of Computer Science, Sichuan University (SCU), China. His current research interests include network management, behavioral data analysis, and resource allocation.

QIBIN LI received the Ph.D. degree from the College of Energy and Power Engineering, Chongqing University (CQU), Chongqing, China, in 2014. He was a Visiting Scholar with the Department of Earth and Environmental Engineering, Columbia University, New York. He is currently an Associate Professor with the College of Energy and Power Engineering, CQU. His current research interests include molecular dynamics (MD) and experimental study of thermodynamic properties of working fluid, thermal circulation system and its thermal management, and the analysis of thermal and mechanical properties of new materials in thermal-force coupling field.

WENZHENG XU (Member, IEEE) received the B.Sc., M.E., and Ph.D. degrees in computer science from Sun Yat-sen University, Guangzhou, China, in 2008, 2010, and 2015, respectively. He was a Visitor with The Australian National University and The Chinese University of Hong Kong. He is currently an Associate Professor with Sichuan University. His research interests include wireless ad hoc and sensor networks, mobile computing, approximation algorithms, combinatorial optimization, online social networks, and graph theory.

XIUJUAN ZHENG received the B.Eng. degree in electronic information engineering from Northwestern Polytechincal University, Xi'an, China, in 2004, the M.Eng. degree in biomedical engineering from Zhejiang University, Hangzhou, China, in 2006, and the Ph.D. degree in electronic and information engineering from The Hong Kong Polytechnic University, in 2011. She was a Postdoctoral Fellow with the School of Medicine, Shanghai Jiao Tong University, from 2011 to 2013. Since 2014, she has been an Associate Professor with the College of Electrical Engineering, Sichuan University, Chengdu, China. Her research interests include quantitative methods for biomedical signal and image analysis and their interdisciplinary applications.