Ultra-Short-Term Industrial Power Demand Forecasting Using LSTM Based Hybrid Ensemble Learning

Article *in* Power Systems, IEEE Transactions on · December 2019

DOI: 10.1109/TPWRS.2019.2963109

Mao Tan
Xiangtan University
46 PUBLICATIONS **477** CITATIONS
SEE PROFILE

Siping Yuan
Xiangtan University
2 PUBLICATIONS **200** CITATIONS
SEE PROFILE

Ultra-Short-Term Industrial Power Demand Forecasting Using LSTM Based Hybrid Ensemble Learning

Mao Tan, *Member, IEEE*, Siping Yuan, Shuaihu Li, Yongxin Su, Hui Li, and Feng He

***Abstract*—Power demand forecasting with high accuracy is a guarantee to keep the balance between power supply and demand. Due to strong volatility of industrial power load, ultra-short-term power demand is difficult to forecast accurately and robustly. To solve this problem, this article proposes a Long Short-Term Memory (LSTM) network based hybrid ensemble learning forecasting model. A hybrid ensemble strategy—which consists of Bagging, Random Subspace, and Boosting with ensemble pruning—is designed to extract the deep features from multivariate data, and a new loss function that integrates peak demand forecasting error is proposed according to bias-variance tradeoff. Experimental results on open dataset and practical dataset show that the proposed model outperforms several state-of-the-art time series forecasting models, and obtains higher accuracy and robustness to forecast peak demand.**

***Index Terms*—Short-term load forecasting, power demand forecasting, deep learning, ensemble learning, long short-term memory (LSTM).**

Nomenclature

LSTM	Long Short Term Memory
SVR	Support Vector Regression
DBN	Deep Belief Network
Seq2Seq	Sequence to Sequence
RFR	Random Forest Regression
EDA	Exploratory Data Analysis
BRSB	Bagging, Random Subspace and Boosting
MAPE	Mean Absolute Percentage Error
MAE	Mean Absolute Error
NRMSE	Normalized Root Mean Square Error
PAPE	Peak Absolute Percentage Error
RNN	Recurrent Neural Network

I. Introduction

Keeping the balance of power supply and demand is a guarantee to improve stability and energy efficiency of power system, and accurate power load demand forecasting is the basis of power control [1]. In particular, in many developing countries, industrial users are the major part of power energy consumers; e.g., in China, about 70 percent of energy is consumed by industrial users. Ultra-short-term forecasting, which generally refers to predicting the load of the future time period a few minutes in advance, is widely used in preventive control and emergency management of power system. By carrying on high accuracy ultra-short-term demand forecasting, power intensive enterprises can make intelligent demand control strategy to adjust electricity scheduling in real time, so as to improve their energy efficiency and economical benefits [2]. Therefore, ultra-short-term demand forecasting has been a important problem in industry.

As a typical time series forecasting problem, statistics based methods—such as auto-regressive integrated moving average [3], grey perdition [4]—have been proposed to handle this problem. However, these models are usually inadequate to fit the complex feature of power consumption and usually result in low forecasting accuracy [5], [6]. Over the past few years, a number of machine learning algorithms, such as artificial neural network, fuzzy neural network, and support vector regression, and some hybrid networks [7]–[10], have been successfully applied to tackle non-linear feature fitting in load forecasting.

Power load forecasting also benefits from deep learning, and get good results in many real-world applications [11]. One of the commonly used deep learning models for time series forecasting is the Recurrent Neural Network (RNN). Researchers applied RNN [12], [13] in electric load forecasting, and found that its performance is obviously better than traditional back propagation neural networks. Some variants of RNN are proposed and achieved better performance than RNN on time series forecasting. For example, Long Short-Term Memory network (LSTM) and Gated Recurrent Unit network (GRU) are usually more powerful than traditional RNN in some load forecasting tasks [14]. Kong *et al.* [15] verified that LSTM based model is suitable for residential load forecasting, and claimed that the model can be further improved if taking more energy sequences from other measurable appliances. Marino *et al.* [16] found that the LSTM based Sequence to Sequence (Seq2Seq) architecture is superior

Manuscript received June 4, 2019; revised October 2, 2019 and November 18, 2019; accepted December 21, 2019. Date of publication December 31, 2019; date of current version June 22, 2020. This work was supported in part by the National Natural Science Foundation of China (61873222, 51777179) and Hunan Provincial Key Research and Development Program (2017GK2244). Paper no. TPWRS-00788-2019. *(Corresponding author: Mao Tan.)*

M. Tan, S. Yuan, S. Li, Y. Su, and H. Li are with the College of Information and Engineering, Xiangtan University, Xiangtan 411105, China (e-mail: mr.tanmao@gmail.com; ysphhb@gmail.com; lishuaihu2010@126.com; suyongxin@xtu.edu.cn; lihui7402@126.com).

F. He is with Hunan Valin Xiangtan Iron and Steel Co., Ltd., Xiangtan 411101, China (e-mail: 163hefeng@163.com).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TPWRS.2019.2963109

to standard LSTM model when forecasting a one-minute ahead residential load. For load forecasting with combined heat and power, Kuan *et al.* [17] designed a concatenated LSTM that consists of two LSTM neural networks, which brings a significant performance improvement compared with single LSTM. He [18] successively integrated Convolutional Neural Network (CNN) and RNN to extract implicit features from historical load series, and took a dense layer to transform other features. However, Fan [19] found that using one-dimensional convolutional and recurrent operations together may not be helpful for building energy predictors in terms of performance variations. Besides the classical time series networks mentioned above, some other deep learning models (e.g., deep belief network (DBN), deep residual network (DRN), etc.) are also used in short-term load forecasting and get good performance with small error [20], [21].

Although many time series forecasting models have been proposed and applied in power load forecasting, there are still some challenges in the real-world load forecasting; i.e., even if an excellent forecasting model is obtained, the forecasting accuracy could fluctuate under different conditions. In this context, ensemble learning methods have been adopted to improve the performance of unstable predictors. Lahouar *et al.* [22] used Random Forest to fuse multiple features for short-term load forecasting, and obtained high accuracy and satisfactory results. Zheng *et al.* [23] used a XGBoost algorithm to pre-calculate the weights of external features, and combined similar days selection and empirical mode decomposition to train a series of separated LSTM model. Qiu *et al.* [24] proposed an DBN-SVR ensemble model that aggregates the outputs of multiple DBNs by SVR, which model outperformed the SVR, Feed-forward NN, DBN and ensemble NN. Dong *et al.* [25] took CNN and clustering methods to enhance forecasting performance. Generally, ensemble deep learning incurs more computational costs than individual model. To solve this problem, combining multiple base learners with ensemble pruning [26] was proposed to bring less computational and memory cost. In addition, feature selection significantly affects the performance of deep learning model. In some time-consumed forecasting tasks, automated feature selection usually gives a computationally efficient model with higher accuracy than the arbitrarily chosen features [27]. Din *et al.* [28] proposed a model to capture the related factors in different time periods by exploratory data analysis (EDA), and found that the synergistic use of EDA with Deep Neural Networks (DNNs) can obtain higher accuracy for short-term load forecasting.

Different from residential and commercial power demands, industrial power demand generally consists of several high power shock loads. Due to strong volatility, ultra-short-term industrial power demand is more difficult to forecast accurately than conventional power demands [29]. In addition, several factors could make the power consumption patterns highly nonlinear and difficult to capture, such as climate condition, time periods, holiday or working days shifting, even some other social activities [30].

In this paper, we propose a novel deep ensemble learning forecasting model required in ultra-short-term power demand control practice. The main contributions of this paper can be summarized as follows: 1) a bias-variance tradeoff based hybrid ensemble strategy is designed to improve both the accuracy and robustness of forecasting; and 2) a new loss function that integrates peak demand forecasting error and a new accuracy metric named Peak Absolute Percentage Error (PAPE) are proposed, which are efficient to enhance the model ability to forecast peak demand.

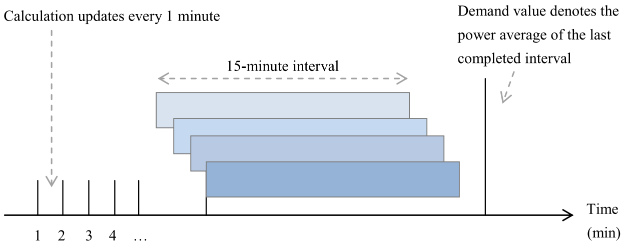

Calculation updates every 1 minute
Demand value denotes the power average of the last completed interval
15-minute interval
1
2
3
4
…
Time (min)

Fig. 1. An example of sliding block power demand.

The rest of this paper is structured as follows. Section II describes the problem in this work and analyzes the technical requirements. Section III focuses on the implementation of the forecasting model. In Section IV, the assessment of the proposed model on an open dataset is given, and in Section V, a practical case study is provided. Finally, the conclusion of this work, as well as the future works are introduced in Section VI.

II. Problem Description

To clarify the problem in this work, we first state that the power demand represents the average of active power within a specified time period. Fig. 1 illustrates the power demand measurement with sliding time block, which is a widely used method to obtain the most comprehensive power demand. As shown, the power demand interval is defined as 15 minutes, and the step size of sliding time block is set to 1 minute. At any specific time, the determined demand value represents the active power average in the last completed time interval.

Power demand is an important indicator to measure the smoothing of power system. For example, in industrial practice with two-part electricity tariffs, the industrial users can reduce electricity costs by controlling maximum power demand, as well as improve the safety of power system operation. To implement maximum power demand control, it is necessary to monitor the change of power demand at any time and forecast the future power demand before the end of a measurement interval, thus to judge if the demand will exceed its limit.

What's more, in order not to affect normal production, the alarm signal of exceeding limit can not be misreported frequently, that is to say, we need an excellent model with both of the minimal bias and variance error to improve the accuracy and robustness of forecasting. At the same time, the forecasting accuracy of peak demand should be concentrated, because one peak demand may lead to a maximum power demand during a settlement period.

In addition, in order to provide sufficient operation time for control program, we need to forecast power demand as early as possible. Fig. 2 provides a scenario of hierarchical power

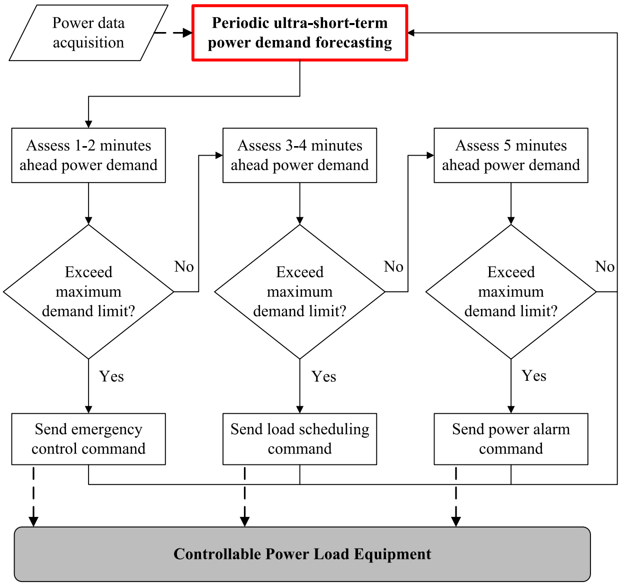

Power data acquisition
Periodic ultra-short-term power demand forecasting
Assess 1-2 minutes ahead power demand
Assess 3-4 minutes ahead power demand
Assess 5 minutes ahead power demand
Exceed maximum demand limit?
No
Exceed maximum demand limit?
No
Exceed maximum demand limit?
No
Yes
Yes
Yes
Send emergency control command
Send load scheduling command
Send power alarm command
Controllable Power Load Equipment

Fig. 2. The flowchart of power demand control.

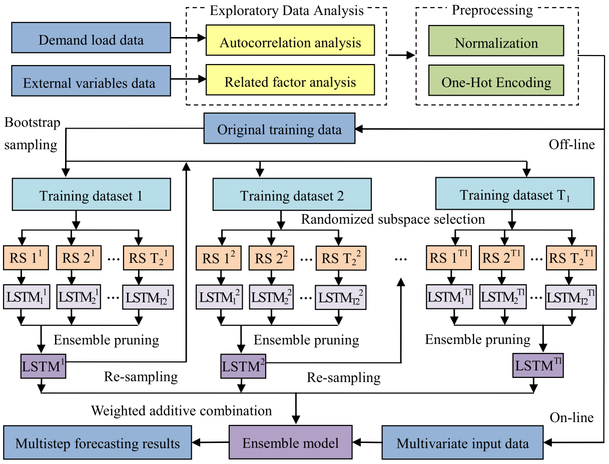


Fig. 3. The full flowchart of the proposed model.

demand control process in this work, in which the dotted lines represent data flow, and the solid lines represent operation flow. As the figure shows, a periodic ultra-short-term forecasting model provides minutes ahead power demand, and different control commands will be sent to the controllable power load equipments according to emergency level. To implement the control strategy, 1 to 5 minutes ahead demands need to be predicted respectively and simultaneously.

III. Methodology

In this section, we focus on the implementation of the forecasting approach. The process flowchart and the diagram of ensemble networks can be seen in Fig. 3. As shown in this figure, we use the power demand data and external data as the input data sources of our model, and construct the feature space using an exploratory data analysis (EDA) based on raw data. Next, a preprocessing will be conducted on the chosen data to normalize the various data, which facilitates the learning model in the next stage. Then, through our hybrid ensemble strategy, several groups of LSTM networks will be off-line trained by different sub-datasets and feature subspace, and then combined to form an ensemble model by sequentially training with ensemble pruning. Finally, the trained model will be used to on-line forecast the multi-step demand values. In the following parts of this section, we will detail the key points in the model implementation.

A. Exploratory Data Analysis and Preprocessing

Blindly construction of model input feature space may lead to reduced data utilization effectiveness, so we use an EDA approach to explore the autocorrelation, periodicity, trend of the power demand data, as well as the correlation between the power demand and other external variables. By analyzing the lag correlation of the present demand against its previous values, we can decide the number of input time-steps. The autocorrelation coefficient can be expressed as

$$R(l) = \sum_{i=1}^{N-l} \frac{(x_i - \bar{x})(x_{i+l} - \bar{x})}{\sum_{i=1}^{N}(x_i - \bar{x})^2}, \tag{1}$$

where N is the number of samples in training dataset, l denotes the time lag, x_i is sequence value at time i, x_{i+1} represents the value at time $i + 1$, and $\bar{x}$ is the mean of samples that could approximately replace the true mean. The denominator represents the variance of the samples.

Ultra-short-term power demand has both strong randomness and trend characteristics, and it is a time series related to many complex factors. In this task, according to the preliminary related factor analysis, a series of external variables are considered as power demand related factors and collected, such as three-phase current, three-phase active power, and time period.

Min-max normalization is then used to scale the data to range [0, 1], which is compatible to the output range of the LSTMs activation function, and guarantees that each input data contributes proportionally to the outputs. The normalized x_i can be defined as

$$\tilde{x}_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, \tag{2}$$

where $x_{\min}$ and $x_{\max}$ represents the minimum and maximum values of variables in the time series, respectively. At the same time, according to initial analysis of EDA, given time period i, the time related variables, hour, day-of-week and month, will be preprocessed through one-hot encoding.

B. Deep Learning Algorithm

As a classical and excellent time series network, LSTM is adopted as base learner due to its strong capability in processing time series data. The architecture of the LSTM model in this work is shown in Fig. 4, which contains two layers of LSTM network with 50 units in first layer and 100 units in second layer.

To reduce overfitting, a dropout layer is set after each LSTM layer to perform the regularization operation, where the rates

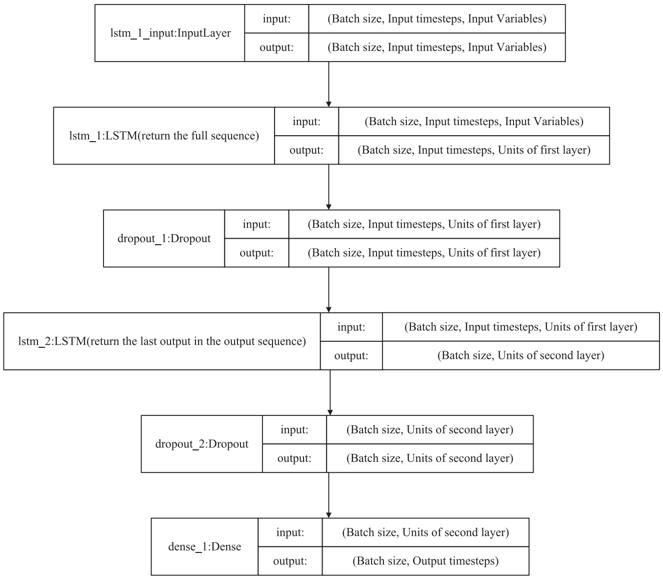

lstm_1_input:InputLayer
input: (Batch size, Input timesteps, Input Variables)
output: (Batch size, Input timesteps, Input Variables)
lstm_1:LSTM(return the full sequence)
input: (Batch size, Input timesteps, Input Variables)
output: (Batch size, Input timesteps, Units of first layer)
dropout_1:Dropout
input: (Batch size, Input timesteps, Units of first layer)
output: (Batch size, Input timesteps, Units of first layer)
lstm_2:LSTM(return the last output in the output sequence)
input: (Batch size, Input timesteps, Units of first layer)
output: (Batch size, Units of second layer)
dropout_2:Dropout
input: (Batch size, Units of second layer)
output: (Batch size, Units of second layer)
dense_1:Dense
input: (Batch size, Units of second layer)
output: (Batch size, Output timesteps)

Fig. 4. The architecture of LSTM model in this article.

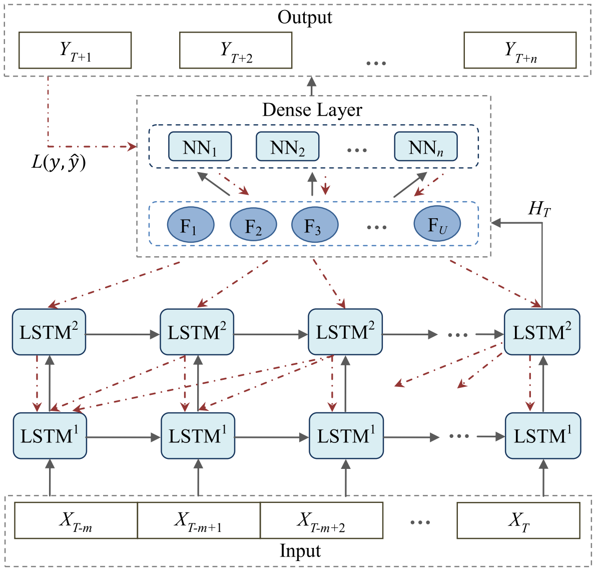


Fig. 5. The forward and back propagation of LSTM in the model.

for different dropout layers can be set respectively, e.g., 0.2 and 0.3 in this work. The activation functions of LSTM layer in the model are the default Sigmod and Tanh. However, as for the Dense layer, because of strong temporal correlation and tendency of power demand time series data, and strong requirement of high accuracy of peak demand forecasting, we employ a simple linear activation function $f(x) = x$ in the last fully connected output layer, which is proved by preliminary experiments to have better performance than the non-saturating or non-linear activation function, such as ReLU or LeakyReLU. In fact, $f(x) = x$ is the standard linear activation function in Keras.

For the purpose of this work—predicting peak demand for maximum demand control, our model is expected to predict power demand as accurately as possible in any condition, especially some unforeseen peak demand. Due to sudden and dramatic change of peak demand, the sample distribution could be uneven for the peak demand forecasting. From machine-learning perspective, the bias-variance decomposition theory attempts to decompose the expected generalization error of learning algorithm into three parts, i.e., bias, variance and intrinsic noise, where the intrinsic noise is the lower bound on the expected error of any learning algorithm on the target, and it indicates the difficulty of learning problem itself; the bias measures how closely the average estimate of the learning algorithm is able to approximate the target; the variance measures how much the estimate of the learning approach fluctuates for different training sets of the same size with same or different sample distribution. In other words, the bias describes the overall accuracy of the model, and the variance describes the stability of the model under different conditions.

However, in practice, the bias and variance can‘t be really calculated because we don’t know the real distribution of data. The decomposition relies on the average of all datasets, but actually we often obtain a limited observation dataset. In this context, we were inspired from bias-variance tradeoff to design a loss function

$$L(y,\hat{y}) = e \times \underset{0<i\leq N}{\text{mean}} (y_i - \hat{y}_i)^2 + (1-e) \times \underset{0<i\leq N}{\max} (y_i - \hat{y}_i)^2, \tag{3}$$

where y is the actual value, and $\hat{y}$ is the predicted value. When updating the parameters of learning algorithm in each batch of training, we consider that the loss function should not only pay attention to high accuracy—which is represented by the first part in the loss function, but also strong stability—which we think can be represented by the second part of the loss function. This loss function helps the model training by combining two different kinds of error—average error on all samples and maximum error on different sample distribution—with a coefficient e, and makes a tradeoff between the two errors, which could ensure that the model exhibits robust prediction performance for each case. For peak demand forecasting scenario—the number of peak demand samples is small, this loss function can make the prediction error as small as possible. Moreover, during the model training, the adaptive learning rate method RMSprop is adopted as the optimizer because it is suitable for dealing with a non-stationary target and has been proven to be an excellent optimizer for recurrent neural networks.

On the whole, the forward propagation and back propagation of this LSTM model are shown in Fig. 5. At first, the input layer of model gets a multivariate sequence $\{X_{T-m}, X_{T-m+1}, \ldots, X_T\}$, m is the input length of time steps. Then, the second LSTM layer output a sequence that represents the hidden layer state H_T of time step T, H_T is a feature vector $\{F_1, F_2, \ldots, F_U\}$, U is the unit number of the second layer. At last, the decoder part— a fully connected neural network, passes H_T to Dense layer that has n output neurons, and generate the predicted values for the n time steps. During each epoch of training, the loss function $L(y,\hat{y})$ and RMSprop optimizer are used to update parameters. Specifically, BP (Back Propagation) algorithm is used in Dense layer, and BPTT (Back Propagation Through Time) algorithm is adopted in LSTM layer.

C. Hybrid Ensemble Strategy

Given that the performance of individual network fluctuates in some cases, ensemble learning usually performs better. In this context, we propose a hybrid ensemble strategy called BRSB (Bagging, Random Subspace and Boosting) to further improve the accuracy and robustness of forecasting. During the training stage, the first two methods increase the diversity of base learner by introducing data randomness with bootstrap sampling, and the random feature subspace will be selected from data attributes, which is not only expected to enhance diversity but also reduce the computational cost due to decreasing the number of input attributes.

The whole process of the proposed ensemble approached based on hybrid BRSB strategy is described in Algorithm 1. By applying the bootstrap sampling with T_1 rounds, T_1 groups of training data subsets are obtained, the number of samples in each group are the same as in the original dataset. Then, we randomly select d dimensional input variables with T_2 times from entire feature space of each training set. Consequently, $T_1 \times T_2$ training sets are created to train base learners h_{t_1,t_2} independently, where $t_1 \in T_1, t_2 \in T_2$. Based on the prediction error in the last training round, Boosting method is used to enhance the accuracy of the model by adjusting the data distribution with re-sampling. The data subset in round t_1-th can be represented as

$$D'_{t_1} = \left\{ (x_i, y_i) | (x_i, y_i) \in D, \frac{1}{t_1} \sum_{j=1}^{t_1} \alpha_j (\varepsilon_j^{(x_i)} - \bar{\varepsilon}_j) > \xi \right\} \tag{4}$$

where

$$\varepsilon_j^{(x_i)} = \frac{|h_j^*(x_i) - y_i|}{y_i}, \bar{\varepsilon}_j = \frac{1}{N} \sum_{i=1}^{N} \varepsilon_j^{(x_i)}, \alpha_j = \ln \frac{1 - \bar{\varepsilon}_j}{\bar{\varepsilon}_j},$$

D is the whole training dataset, α_j is the aggregation coefficient of the j-th individual learner h_j^* according to its mean absolute percentage error $\bar{\varepsilon}_j$ on D. Then, the samples (x_i, y_i) with error $\varepsilon_j^{(x_i)}$ that exceed the threshold ξ are appended into the training dataset for the next round of bootstrap sampling.

After training multiple individual learners by different feature subspaces in round t_1, the ensemble pruning is conducted and then an optimal learner with lowest validation error will be retained. Given $t_1 \in T_1$, we can get an individual learner

$$h_{t_1}^* = \underset{h_{t_1,t_2}}{\arg\min} \sum_{(x,y) \in V} L(y, h_{t_1,t_2}(x)), \tag{5}$$

where $t_2 \in T_2$, V is the validation dataset that created by randomly choosing data from original samples at a fixed ratio 5%, and obviously V should never be included in the training dataset. h_{t_1,t_2} represents any picked item from the trained individual learners. Then, the base learners with poor performance are filtered out, which will produce a pruned ensemble model with smaller size, and less computing and memory cost.

Algorithm 1: BRSB Ensemble Algorithm.

```
Input: Dataset $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i), \dots, (x_m, y_m)\}$;
       The LSTM base learner;
       The number of learning rounds for Bagging, $T_1$;
       The number of learning rounds for Random Subspace, $T_2$;
       The feature subset size, $d$.
Output: $H(x) = \sum_{t_1=1}^{T_1} w_{t_1} h_{t_1}^*(x)$
Process:
for $t_1 = 1$ to $T_1$
    $D'_0 = []$
    $D_{t_1} = BS(D, D'_{t_1})$//Bootstrap sampling
    for $t_2 = 1$ to $T_2$
        $F_{t_2} = RSFS(D_{t_1}, d)$//Randomly select feature subspace
        $D_{t_1,t_2} = Map_{F_{t_2}}(D_{t_1})$// Projecting to feature subspace
        $h_{t_1,t_2} = \text{LSTM}(D_{t_1,t_2})$
    end for
    $h_{t_1}^* = EP_{1 \le t_2 \le T_2}(h_{t_1 \times t_2})$//Ensemble pruning
    $D'_{t_1} = RS(D, \xi)$//Re-sample the poorly predicted samples
end for
```

The outputs of all base learners are combined by a weight that calculated from α_j. For the selected learner $h_{t_1}^*$ in the t_1-th round bootstrap sampling, we can define the weight

$$w_{t_1} = \frac{\alpha_{t_1}}{\sum_{j=1}^{T_1} \alpha_j}, \tag{6}$$

using this weight coefficient, we iteratively combine a set of LSTM models using weighted additive combination. This weighted additive model is different from the additive model in traditional Boosting method, the weight for each learner is related to the model errors in all training rounds but not just current training round. Finally, we get an ensemble learning network with the consideration of both overall and peak demand forecasting performance.

IV. Assessment on Open Dataset

In order to assess the performance of our model, we chose an open dataset and compared the forecasting value with published results of short-term load demand forecasting in Ref. [31]. The open dataset is retrieved from Australian Energy Market Operator (AEMO), it contains half-hourly electricity load data during 2013 recorded in New South Wales, Australia. In the experiments in Ref. [31], the AEMO dataset was partitioned into 12 smaller monthly datasets and targeted for 1–24 h ahead forecasting. For each monthly dataset, the first 50% data were used for training and the remaining 50% was used for testing. The full feature set contains 2 days historical value. The results of 1 h, 12 h, and 24 h ahead forecasting were published.

For the fairness of assessment, we used the same training and testing data as in Ref. [31]. The published results combined with

TABLE I
THE NRMSE OF DEMAND FORECASTING ON OPEN DATASET AEMO

	Horizon=1h							Horizon=12h							Horizon=24h						
	Per.	sARIMA	ANN	SVR	RF	RVFL	Ours	Per.	sARIMA	ANN	SVR	RF	RVFL	Ours	Per.	sARIMA	ANN	SVR	RF	RVFL	Ours
1	0.160	0.218	0.134	0.056	0.044	**0.018**	**0.015**	0.160	0.245	0.138	0.114	**0.108**	0.117	**0.062**	0.161	0.265	0.340	0.150	**0.118**	0.120	**0.074**
2	0.121	0.190	0.042	0.063	0.058	**0.025**	**0.023**	0.121	0.153	0.195	0.128	0.114	**0.113**	**0.077**	0.124	0.140	0.188	0.132	0.113	**0.109**	**0.091**
3	0.118	0.172	0.088	0.069	0.047	**0.028**	**0.028**	**0.118**	0.194	0.146	0.215	0.132	0.156	**0.061**	**0.115**	0.246	0.256	0.165	0.177	0.173	**0.089**
4	0.113	0.178	0.049	0.040	0.045	**0.026**	**0.028**	0.108	0.180	0.083	0.114	**0.076**	0.086	**0.059**	0.098	0.204	0.082	**0.074**	0.075	0.085	**0.077**
5	0.103	0.181	0.097	0.090	0.083	**0.041**	**0.041**	**0.103**	0.133	0.128	0.135	0.104	0.137	**0.099**	**0.104**	0.136	0.196	0.126	0.118	0.153	**0.098**
6	0.101	0.194	0.059	0.073	0.070	**0.030**	**0.029**	0.099	0.189	0.152	0.103	**0.098**	0.109	**0.088**	**0.098**	0.208	0.167	0.145	0.116	0.128	**0.092**
7	0.095	0.195	0.081	0.049	0.050	**0.032**	**0.025**	0.091	0.134	0.087	0.096	**0.081**	0.122	**0.049**	0.091	0.133	0.113	0.100	**0.086**	0.119	**0.054**
8	0.121	0.216	0.070	0.067	0.053	**0.027**	**0.025**	0.121	0.234	0.130	0.115	**0.097**	0.102	**0.072**	0.123	0.256	0.154	0.152	0.116	**0.114**	**0.073**
9	0.122	0.200	0.076	0.046	0.056	**0.027**	**0.031**	0.127	0.204	0.129	0.115	**0.096**	0.100	**0.066**	0.135	0.222	0.117	**0.092**	0.097	0.104	**0.068**
10	0.139	0.217	0.049	0.064	0.040	**0.025**	**0.022**	0.139	0.233	0.237	0.120	**0.108**	0.115	**0.064**	0.139	0.283	0.202	0.129	**0.114**	0.118	**0.078**
11	0.120	0.194	0.098	0.054	0.044	**0.026**	**0.028**	0.126	0.210	0.175	0.105	**0.098**	0.103	**0.056**	0.131	0.245	0.234	**0.108**	0.115	0.110	**0.065**
12	0.152	0.188	0.170	0.073	0.059	**0.023**	**0.022**	0.152	0.199	0.205	**0.140**	0.147	0.149	**0.084**	**0.149**	0.235	0.207	0.188	0.151	0.162	**0.109**

our results are all provided in Table I. In the table, for each time horizon, the first six columns represent the published results, and the last column gives the results of our model. The row number represents the sequence number of monthly dataset. From the results, we can see that RVFL, the model proposed in Ref. [31], outperforms the comparison models except ours while horizon = 1 h, but if the horizon is 12 h or 24 h, RVFL can't always get the smallest NRMSE on each monthly dataset. However, our model always gets smaller NRMSE on each monthly dataset than all the other models, whether the horizon is 1 h, 12 h, or 24 h. From the statistical results on the AEMO dataset, we can conclude that our proposed model has good performance, both accuracy and robustness, in short-term load demand forecasting.

V. CASE STUDY

In this section, we will provide a case study that applies the proposed model to the power demand forecasting in a practical steel plant. Under the premise of satisfying the power demand in production, controlling the maximum power demand effectively to reduce the power demand bill has become an urgent problem in engineering. To solve this problem, we collect the power demand related data from a data acquisition workstation in the central substation that connects to power grid, and then predict the future power demand according to the historical and current data. At last, the predicted power demand will send to a demand control module.

A. Data Collection and Analysis

The real-time power demand related data in this case study are collected form a practical steel plant and, which consists of three-phase electric current, three-phase active power and three-phase demand active power of the entire plant. The data are collected at a frequency of 60 seconds. Table II gives the original attributes in the dataset. As shown in the table, there are two observations about the power demand. Present power demand, the value for the last completed interval, which is used to compare with the maximum demand limit. Recent power demand, which is the real time value measured by the field data acquisition meters, and which represents the trend in the near future. The full dataset and the description can be obtained from $https://github.com/YuanSiping/IPPD$.

TABLE II
ORIGINAL ATTRIBUTES IN DATA COLLECTION

Attributes	Description
Date	YYYY/MM/DD HH:MM:SS
T_DEM	Three-phase total power demand for present interval
A/B/C_CUR	Real-time current of phase A/B/C
A/B/C/T_ACT	Three-phase total real-time active power of phase A/B/C
S_DEM	Three-phase recent power demand updated every second
Hour	0 $\sim$ 23 corresponds to 24 hour of the day
Month	1 $\sim$ 12 corresponds to January $\sim$ December
Day of Week	1 $\sim$ 7 corresponds to Monday $\sim$ Sunday

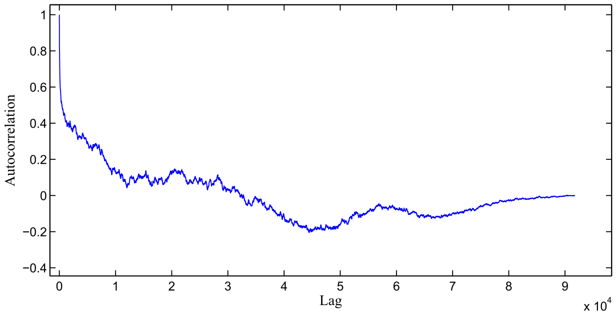


Fig. 6. The autocorrelation curve of power demand.

As described in Section III-A, the exploratory data analysis is conducted between the present power demand and the original attributes that are shown in Table II. A visible curve of autocorrelation function is shown in Fig. 6. We find that there is strong autocorrelation at the early lags of the power demand time series, which indicates that the data has a strong tendency without obvious periodicity. The closer near the check point, the higher the correlation appears, which inspired us to choose the actual power demand of the nearest few minutes as the input.

Furthermore, through the EDA method, although the ultra-short-term power demand has weak periodicity, it is still related to hour, day of the week, and month. Finally, as is shown in Fig. 7, the input feature space consists of an input sequence of 52 dimensional variables, the sequence length is n. The 52 dimensional variables consist of present power demand T_DEM, three-phase current A/B/C_CUR, three-phase active power A/B/C_ACT,

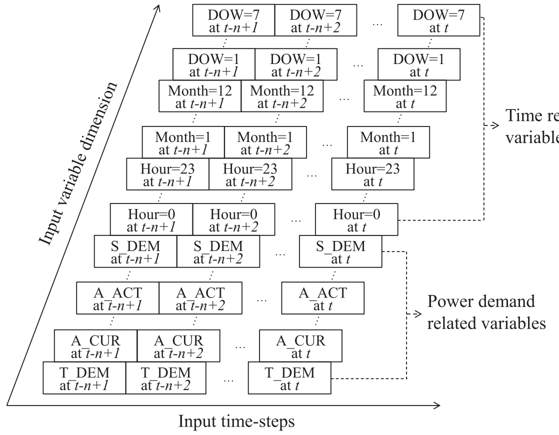


Fig. 7. The original input feature space.

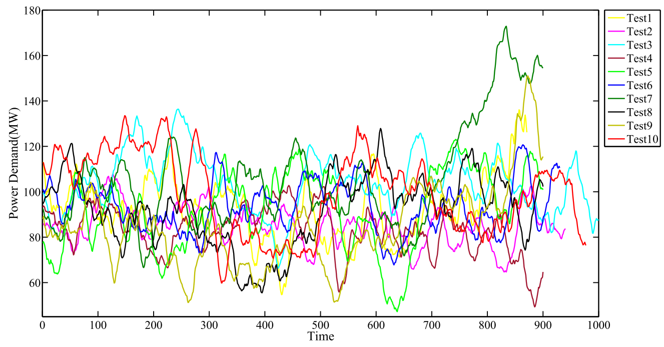


Fig. 8. The ten testing load curves.

total active power T_ACT, recent power demand S_DEM, hour labels, day-of-week labels and month labels. The time related labels are one-hot encoded, and the sizes of one-hot encoded vector for hour, day-of-week and month variables are 24, 7 and 12, respectively. The reconstructed feature space for single LSTM network is set as 25 dimensions, which consists of the present power demand and other 24 dimensional variables that randomly chosen using randomized subspace method. According to the strong tendency of this power demand, the input time-steps of model is determined to be 9 based on the values of enumerations 1 through 15, and output time-steps is set to 5, which is required by the power demand control strategy described in Section II.

Ten testing datasets, which are randomly selected from the collected raw data and not included in the training dataset, are chosen to assess the proposed model. The demand change patterns of the testing datasets are shown as Fig. 8. It can be seen that the fluctuation of this power demand are frequent and intensive, which reflects the strong volatility and high nonlinearity of industrial ultra-short-term power demand. During the experiment, we acquire 164626 data points to train model, and use the ten datasets to test model.

B. Evaluation Metrics

TABLE III
THE OPTIMAL PARAMETERS OF THE COMPARISON MODELS

Methods	Optimal parameters
SVR	C=2.0, $kernel$=rbf
DBN	n_h=[150,300,200], a_h=sigmoid, a_o=linear
Seq2Seq	n_e=150, n_d=[250,150], d=[0.2,0.2,0.3], a_o=linear
LSTM	n_h=[50,100], d=[0.2,0.2], a_o=linear
RFR	$ntree$=100, $mtry$=5
XGBoost	max_{depth}=6, eta=0.1, $objective$=reg:linear

*C–penalty coefficient; n_h–number of hidden layer units; a_h–activation function of hidden layer; a_o–activation function of output layer; n_e–number of encoder units; n_d–number of decoder units; d–weight of dropout; $ntree$–number of trees; $mtry$–number of variables randomly sampled at each split; $\max_{depth}$–maximum depth of the trees; eta–learning rate.

To evaluate the overall forecasting accuracy of the proposed model, three commonly used evaluation metrics are adopted in this work, which are Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Normalized Root Mean Square Error (NRMSE). In addition, we find that peak demand forecasting is very important to maximum power demand control. We know that the maximum power demand during a settlement period comes from the peak power demand in local time periods, and the fitting effect of the local maximum value could definitely improve the peak demand forecasting accuracy in a specified settlement period. Therefore, a new criterion of error evaluation to analyze the forecasting performance for peak power demand is proposed, which is named Peak Absolute Percentage Error (PAPE) and expressed as

$$\mathrm{PAPE}(y,\hat{y}) = \frac{T}{n}\sum_{i=1}^{\frac{n}{T}} \frac{|\hat{y}_i^{\mathrm{peak}} - y_i^{\mathrm{peak}}|}{y_i^{peak}}, \qquad (7)$$

where T is a measurement cycles of local maximum power demand, which in this evaluation is set to 15 minute; n is the total time interval of the test dataset; y_i^{peak} is the local maximum power demand in the i-th cycle; and $\hat{y}_i^{peak}$ is the predicted value of corresponding maximum demand.

C. Experimental Results and Analysis

During the experimental procedure, several classical time series forecasting models—including SVR with Radial basis function (RBF) kernel, DBN with three layers of Restricted Boltzmann Machine (RBM), LSTM-based Seq2Seq, standard LSTM, RFR with 100 regression trees, and XGBoost with linear regression objective function—are employed to compare with our proposed model. The optimal parameters of the methods are shown in Table III. It should be noted that among the models, LSTM, LSTM-based Seq2Seq, DBN, RFR and our proposed model output all 5 time-steps forecasting value once and for all, SVR forecasts the values step by step—it means that the previous output value is considered as the input of next time-step, and Xgboost trains a separate model for each time-step.

1) Overall Power Demand Forecasting: In this section, we provide the overall demand forecasting to assess the proposed ensemble model in the case of multivariate inputs. The overall forecasting errors of each test dataset, including the MAPE of

TABLE IV
THE OVERALL FORECASTING PERFORMANCE METRICS OF THE MODELS

Methods	MAPE-1		MAPE-2		MAPE-3		MAPE-4		MAPE-5		NRMSE		MAE	
	Test1	Test2	Test1	Test2	Test1	Test2	Test1	Test2	Test1	Test2	Test1	Test2	Test1	Test2
SVR	5.484	4.190	6.034	4.532	6.551	4.869	7.065	5.197	7.539	5.509	9.188	9.308	5.884	4.356
DBN	3.961	2.361	4.650	2.873	5.297	3.359	5.905	3.793	6.482	4.188	7.482	6.574	4.845	3.019
Seq2Seq	2.467	2.730	2.588	2.535	**2.953**	2.624	**3.462**	2.872	**4.012**	3.238	**4.382**	5.582	**2.853**	2.543
LSTM	2.137	1.862	**2.564**	**2.103**	3.092	**2.422**	3.580	**2.607**	4.147	**2.941**	4.459	**4.967**	2.884	**2.249**
RFR	2.639	3.015	3.557	3.100	3.342	3.214	5.069	3.369	5.021	3.603	6.694	6.599	4.323	3.010
XGBoost	**1.937**	**1.531**	2.795	2.297	3.638	3.018	4.383	3.743	5.035	4.411	5.026	5.065	3.244	2.574
Ours	**1.403**	**1.132**	**1.962**	**1.549**	**2.584**	**1.952**	**3.234**	**2.361**	**3.875**	**2.784**	**3.664**	**3.947**	**2.395**	**1.800**

Methods	MAPE-1		MAPE-2		MAPE-3		MAPE-4		MAPE-5		NRMSE		MAE	
	Test3	Test4	Test3	Test4	Test3	Test4	Test3	Test4	Test3	Test4	Test3	Test4	Test3	Test4
SVR	4.513	4.356	4.500	4.769	4.536	5.169	4.614	5.556	4.731	5.928	7.883	8.902	4.496	5.246
DBN	4.257	4.579	4.553	5.112	4.936	5.712	5.281	6.281	5.616	6.832	8.435	10.571	4.935	4.500
Seq2Seq	1.841	3.318	2.059	3.008	2.460	3.009	2.922	3.220	3.386	3.600	4.536	6.313	2.591	2.616
LSTM	2.073	1.920	2.283	2.235	2.662	**2.624**	**2.768**	**2.939**	**3.125**	**3.351**	4.797	**4.971**	2.698	**2.147**
RFR	2.833	3.228	2.978	3.409	3.157	3.668	3.399	3.980	3.643	4.339	6.285	6.941	2.958	2.926
XGBoost	**1.149**	**1.504**	**1.759**	**2.208**	**2.365**	2.864	2.931	3.509	3.488	4.136	**4.261**	5.400	**2.394**	2.283
Ours	**1.036**	**1.215**	**1.477**	**1.777**	**1.942**	**2.343**	**2.357**	**2.873**	**2.760**	**3.435**	**3.494**	**4.444**	**1.960**	**1.893**

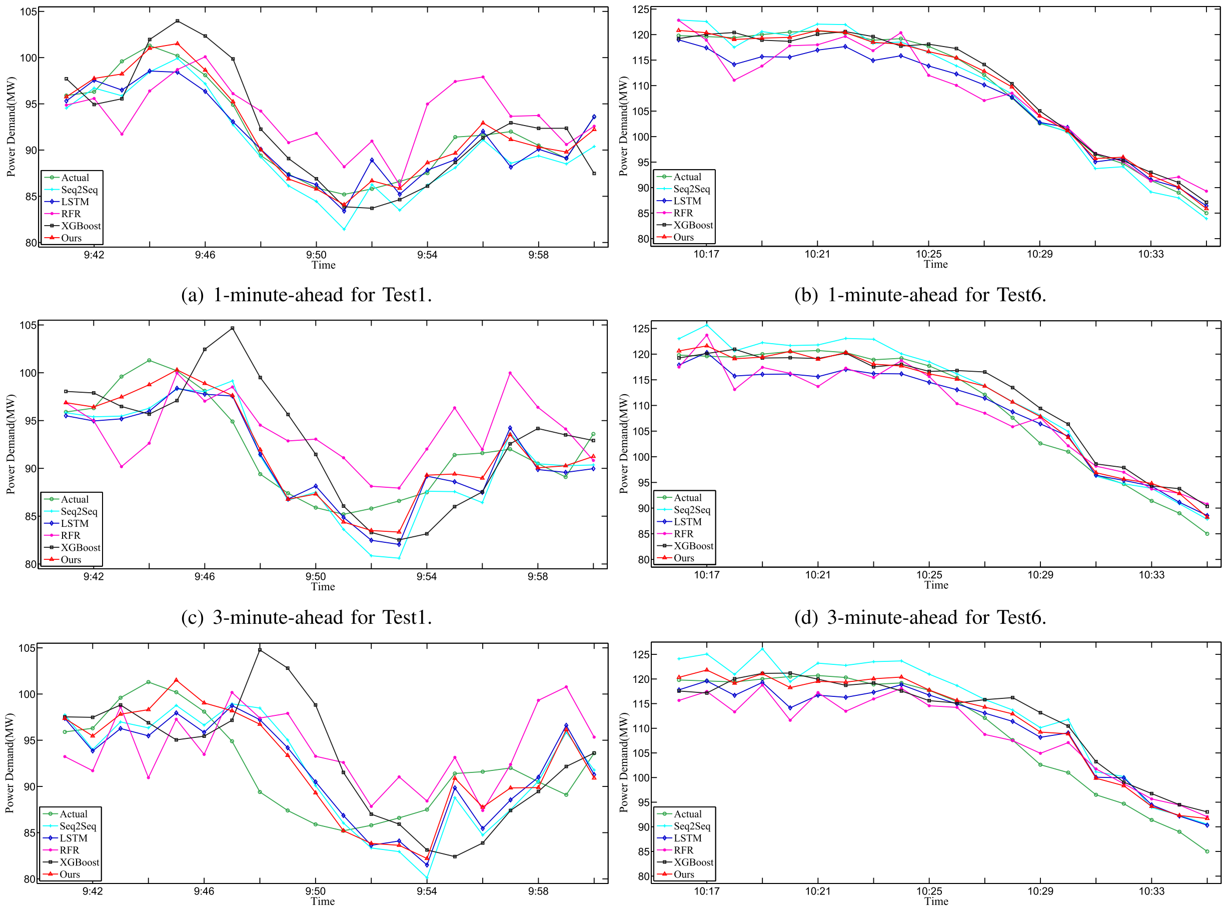
Methods	MAPE-1		MAPE-2		MAPE-3		MAPE-4		MAPE-5		NRMSE		MAE	
	Test5	Test6	Test5	Test6	Test5	Test6	Test5	Test6	Test5	Test6	Test5	Test6	Test5	Test6
SVR	5.986	5.126	6.560	5.507	7.118	5.887	7.666	6.264	8.194	6.649	9.834	12.799	6.013	5.355
DBN	5.499	5.570	5.890	5.819	6.432	6.198	6.972	6.584	7.512	6.969	9.040	12.703	5.585	5.703
Seq2Seq	2.713	1.666	2.705	1.950	**3.027**	**2.451**	**3.489**	**3.039**	**4.027**	3.673	**4.614**	5.543	**2.809**	2.359
LSTM	2.198	2.136	2.600	2.554	3.131	3.080	3.526	3.562	4.106	4.122	4.632	**4.858**	2.845	2.902
RFR	3.442	2.792	3.633	3.028	3.971	3.309	4.399	3.660	4.881	4.031	5.811	7.294	3.504	3.091
XGBoost	**1.590**	**1.297**	**2.392**	**1.914**	3.196	2.504	3.966	3.105	4.700	**3.672**	4.631	5.515	2.833	**2.313**
Ours	**1.278**	**1.136**	**1.904**	**1.591**	**2.557**	**2.081**	**3.225**	**2.605**	**3.895**	**3.121**	**3.790**	**4.645**	**2.302**	**1.947**

Methods	MAPE-1		MAPE-2		MAPE-3		MAPE-4		MAPE-5		NRMSE		MAE	
	Test7	Test8	Test7	Test8	Test7	Test8	Test7	Test8	Test7	Test8	Test7	Test8	Test7	Test8
SVR	5.107	5.486	5.534	5.911	5.960	6.332	6.386	6.752	6.800	7.148	7.035	9.136	6.004	5.481
DBN	5.072	7.437	5.221	7.626	5.491	7.968	5.788	8.290	6.129	8.612	6.251	13.180	5.432	7.061
Seq2Seq	1.492	3.193	**1.715**	3.321	**2.089**	3.762	**2.558**	4.334	**3.076**	4.960	**2.619**	7.319	**2.225**	3.528
LSTM	2.996	5.091	3.080	5.221	3.340	5.456	3.320	5.622	3.661	5.955	4.128	7.616	3.605	5.078
RFR	3.469	3.665	3.607	3.868	3.831	4.108	4.109	4.426	4.464	4.766	6.246	5.225	4.610	5.027
XGBoost	**1.226**	**1.477**	1.793	**2.242**	2.346	**2.961**	2.933	**3.663**	3.534	**4.283**	2.882	**4.502**	2.438	**2.625**
Ours	**1.061**	**1.224**	**1.524**	**1.826**	**1.985**	**2.408**	**2.474**	**2.990**	**2.992**	**3.554**	**2.421**	**3.761**	**2.054**	**2.165**

Methods	MAPE-1		MAPE-2		MAPE-3		MAPE-4		MAPE-5		NRMSE		MAE	
	Test9	Test10	Test9	Test10	Test9	Test10	Test9	Test10	Test9	Test10	Test9	Test10	Test9	Test10
SVR	7.041	5.443	7.467	5.563	7.897	5.972	8.326	6.365	8.757	6.743	7.678	9.516	6.156	5.510
DBN	5.720	5.747	6.053	6.055	6.519	6.466	7.008	6.843	7.502	7.206	6.422	9.628	5.314	6.059
Seq2Seq	3.479	1.921	3.615	1.973	4.050	**2.315**	4.658	**2.795**	5.394	**3.341**	5.846	**4.034**	3.460	**2.385**
LSTM	2.016	2.120	2.435	2.539	**2.887**	3.069	**3.406**	3.559	**3.979**	4.123	**3.064**	4.533	**2.400**	2.915
RFR	2.856	2.807	3.078	2.950	3.402	3.220	3.832	3.568	4.282	3.983	6.139	5.497	3.870	3.206
XGBoost	**1.474**	**1.294**	**2.234**	**1.940**	2.991	2.557	3.749	3.179	4.505	3.785	3.183	4.124	2.484	2.462
Ours	**1.242**	**1.007**	**1.805**	**1.500**	**2.383**	**2.002**	**2.982**	**2.524**	**3.615**	**3.054**	**2.563**	**3.320**	**1.991**	**1.959**

each time-step and the average NRMSE and MAE for all time-steps, are shown in Table IV, in which the minimum errors of the contrast models and the errors of our model are emphasized in bold text. From the table, it is obvious to see that among the individual models, LSTM and Seq2Seq have lower prediction error than SVR and DBN, even some state-of-the-art ensemble methods, e.g., RFR and XGBoost, that have been widely verified in many scenes, also have larger forecasting errors than LSTM in some conditions. This may be due to the excellent time sequence processing capability of LSTM and Seq2Seq, and the weak base learners of RFR and XGBoost that based on decision tree. These results indicate that the existing methods have weak adaptability and robustness for different scenarios. However, our proposed model basically outperforms all the other models on each test datasets, whichever metric is considered. Consequently, it can be concluded that our LSTM based hybrid ensemble learning model is effective and robust to solve the demand forecasting problem in this paper.

In addition, Test1 and Test6 are selected as representatives to show the fitting results of the model. Fig. 9 plots the fitting curves in a short period for several good models, as well as the actual power demand curve. In the legend, the three sub-figures in each column correspond to the 1, 3, and 5 minutes ahead forecasting, respectively. It can be seen that our model fits the actual power demand curve better than the other models, on the whole. For each test dataset, along the prediction time horizon, i.e., from 1 to 5 minutes, the forecasting error propagates and becomes larger, this is mainly due to the high variability of ultra-short industrial power load, which lead to the difficulty to forecast accurately. Nevertheless, we have to say our proposed model performs the best to forecast the power demand, whichever dataset or time horizon is considered.



(a) 1-minute-ahead for Test1.

(b) 1-minute-ahead for Test6.

(c) 3-minute-ahead for Test1.

(d) 3-minute-ahead for Test6.

(e) 5-minute-ahead for Test1.

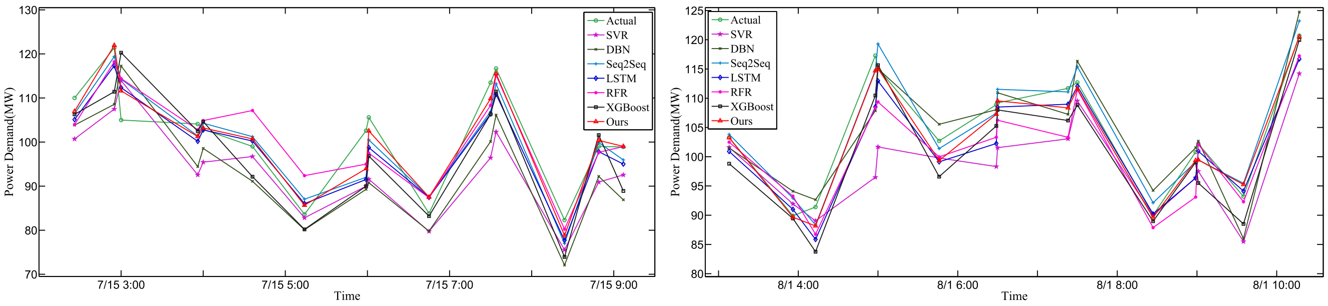
(f) 5-minute-ahead for Test6.

Fig. 9. The overall power demand forecasting results of Test1 and Test6.

TABLE V
THE PEAK DEMAND FORECASTING PERFORMANCE METRICS OF THE MODELS

Methods	PAPE-1					PAPE-3					PAPE-5				
	Test1	Test2	Test3	Test4	Test5	Test1	Test2	Test3	Test4	Test5	Test1	Test2	Test3	Test4	Test5
SVR	5.399	3.492	3.703	3.496	5.276	6.867	4.222	3.542	4.556	6.617	7.980	4.849	3.834	5.047	7.422
DBN	5.138	2.011	2.741	3.467	3.734	6.925	3.128	3.338	4.133	3.973	8.503	4.124	3.781	4.758	4.947
Seq2Seq	2.506	2.916	1.950	3.129	2.342	**3.225**	**2.827**	2.485	3.238	2.711	**4.035**	**3.013**	**3.075**	3.776	**3.667**
LSTM	2.468	2.305	2.735	2.212	2.665	3.674	3.086	3.486	3.079	3.251	4.425	3.331	3.958	**3.735**	4.440
RFR	3.072	3.073	2.515	3.002	3.624	3.832	3.333	3.184	3.430	3.583	4.974	3.544	3.370	4.156	4.324
XGBoost	**1.531**	**1.233**	**0.933**	**1.408**	**1.190**	3.473	3.095	**2.481**	**2.690**	**2.601**	5.295	4.476	3.912	4.282	4.723
Ours	**1.438**	**1.042**	**0.925**	**1.157**	**1.010**	**2.432**	**2.016**	**2.277**	**2.064**	**2.114**	**3.591**	**2.720**	**3.223**	**3.334**	**3.991**
Methods	PAPE-1					PAPE-3					PAPE-5				
	Test6	Test7	Test8	Test9	Test10	Test6	Test7	Test8	Test9	Test10	Test6	Test7	Test8	Test9	Test10
SVR	3.941	4.571	4.210	4.898	4.718	4.687	5.763	5.045	5.335	4.988	5.237	6.151	6.130	6.597	5.247
DBN	4.116	4.097	6.071	4.233	4.289	4.343	4.119	5.637	4.394	4.361	4.677	3.927	6.276	5.694	4.427
Seq2Seq	1.371	1.223	3.193	3.403	1.972	**2.383**	**1.668**	3.444	3.647	**2.527**	**2.957**	**2.653**	**3.603**	4.900	**3.524**
LSTM	2.558	3.461	5.407	2.037	2.512	3.273	3.764	6.048	**2.648**	3.474	4.245	4.019	6.257	**3.846**	4.321
RFR	3.092	4.167	3.912	2.674	3.666	3.092	4.567	4.266	3.840	3.666	3.833	5.471	5.376	4.614	4.500
XGBoost	**1.102**	**1.024**	**1.323**	**1.186**	**1.171**	2.560	2.157	**2.938**	2.890	2.542	3.549	3.521	4.138	4.832	3.706
Ours	**0.970**	**0.824**	**1.324**	**1.137**	**1.036**	**2.037**	**1.738**	**2.367**	**2.406**	**2.112**	**2.924**	**2.620**	**2.959**	**4.117**	**3.297**

2) Peak Demand Forecasting: In this section, we use the customized metric PAPE to assess the peak demand forecasting performance of the proposed model. The 1, 3, and 5 minutes ahead errors of different models are shown in Table V. It can be still seen that our model outperforms other models with minimum errors and robust performance on all ten datasets. Except for our model, none of the other models can forecast well on each test dataset. This can be expected as our method can reduce redundant data or noise interference by hybrid ensemble strategy, and it trains LSTM network sequentially by



(a) The fitting effect of Test1.

(b) The fitting effect of Test6.

Fig. 10. The 5-minute-ahead peak demand forecasting results of Test1 and Test6.

TABLE VI
THE PERFORMANCE METRICS OF OUR MODEL USING DIFFERENT LOSS FUNCTION

Methods	MAPE-1		MAPE-3		MAPE-5		NRMSE		MAE		PAPE-1		PAPE-3		PAPE-5	
	Test1	Test2	Test1	Test2	Test1	Test2	Test1	Test2	Test1	Test2	Test1	Test2	Test1	Test2	Test1	Test2
Eq. (3)	1.403	**1.132**	2.584	1.952	3.875	2.784	**3.664**	**3.947**	**2.395**	1.800	**1.438**	**1.042**	**2.432**	**2.016**	**3.591**	**2.720**
Eq. (8)	**1.401**	1.173	**2.558**	**1.934**	**3.834**	**2.711**	3.698	3.952	2.405	**1.785**	1.654	1.344	2.770	2.311	3.610	2.913
Methods	MAPE-1		MAPE-3		MAPE-5		NRMSE		MAE		PAPE-1		PAPE-3		PAPE-5	
	Test3	Test4	Test3	Test4	Test3	Test4	Test3	Test4	Test3	Test4	Test3	Test4	Test3	Test4	Test3	Test4
Eq. (3)	**1.036**	**1.215**	**1.942**	2.343	2.760	3.435	**3.494**	4.444	**1.960**	1.893	**0.925**	**1.157**	**2.277**	**2.064**	**3.223**	**3.334**
Eq. (8)	1.065	1.217	1.943	**2.314**	**2.751**	**3.393**	3.526	**4.408**	1.968	**1.875**	1.119	1.234	2.400	2.229	3.344	3.476
Methods	MAPE-1		MAPE-3		MAPE-5		NRMSE		MAE		PAPE-1		PAPE-3		PAPE-5	
	Test5	Test6	Test5	Test6	Test5	Test6	Test5	Test6	Test5	Test6	Test5	Test6	Test5	Test6	Test5	Test6
Eq. (3)	**1.278**	**1.136**	2.557	**2.081**	3.895	**3.121**	3.790	**4.645**	2.302	**1.947**	**1.010**	**0.970**	**2.114**	**2.037**	**3.991**	**2.924**
Eq. (8)	1.285	1.169	**2.536**	2.118	**3.872**	3.161	**3.761**	4.735	**2.297**	1.991	1.159	1.218	2.131	2.207	4.037	3.121
Methods	MAPE-1		MAPE-3		MAPE-5		NRMSE		MAE		PAPE-1		PAPE-3		PAPE-5	
	Test7	Test8	Test7	Test8	Test7	Test8	Test7	Test8	Test7	Test8	Test7	Test8	Test7	Test8	Test7	Test8
Eq. (3)	**1.061**	**1.224**	1.985	**2.408**	2.992	**3.554**	**2.421**	**3.761**	**2.054**	**2.165**	**0.824**	**1.324**	**1.738**	**2.367**	**2.620**	**2.959**
Eq. (8)	1.095	1.332	**1.978**	2.444	**2.980**	3.558	2.437	3.827	2.068	2.217	1.044	1.507	1.804	2.546	2.765	3.355
Methods	MAPE-1		MAPE-3		MAPE-5		NRMSE		MAE		PAPE-1		PAPE-3		PAPE-5	
	Test9	Test10	Test9	Test10	Test9	Test10	Test9	Test10	Test9	Test10	Test9	Test10	Test9	Test10	Test9	Test10
Eq. (3)	1.242	**1.007**	2.383	2.002	**3.615**	**3.054**	2.563	**3.320**	**1.991**	**1.959**	**1.137**	**1.037**	**2.406**	**2.112**	**4.117**	**3.297**
Eq. (8)	**1.225**	1.042	**2.365**	**1.999**	3.615	3.077	**2.559**	3.346	1.994	1.983	1.219	1.168	2.560	2.179	4.223	3.464

Boosting to adjust training data distribution. For each Boosting, a multi-round Random Subspace algorithm is executed on a training subset obtained by Bagging, so each LSTM model is as independent as possible. Adopting ensemble pruning to select optimal model learned from a feature subspace for each Bagging, the selected model inputs most suitable feature. In summary, these results indicate that our proposed model has the best accuracy and robustness over other models.

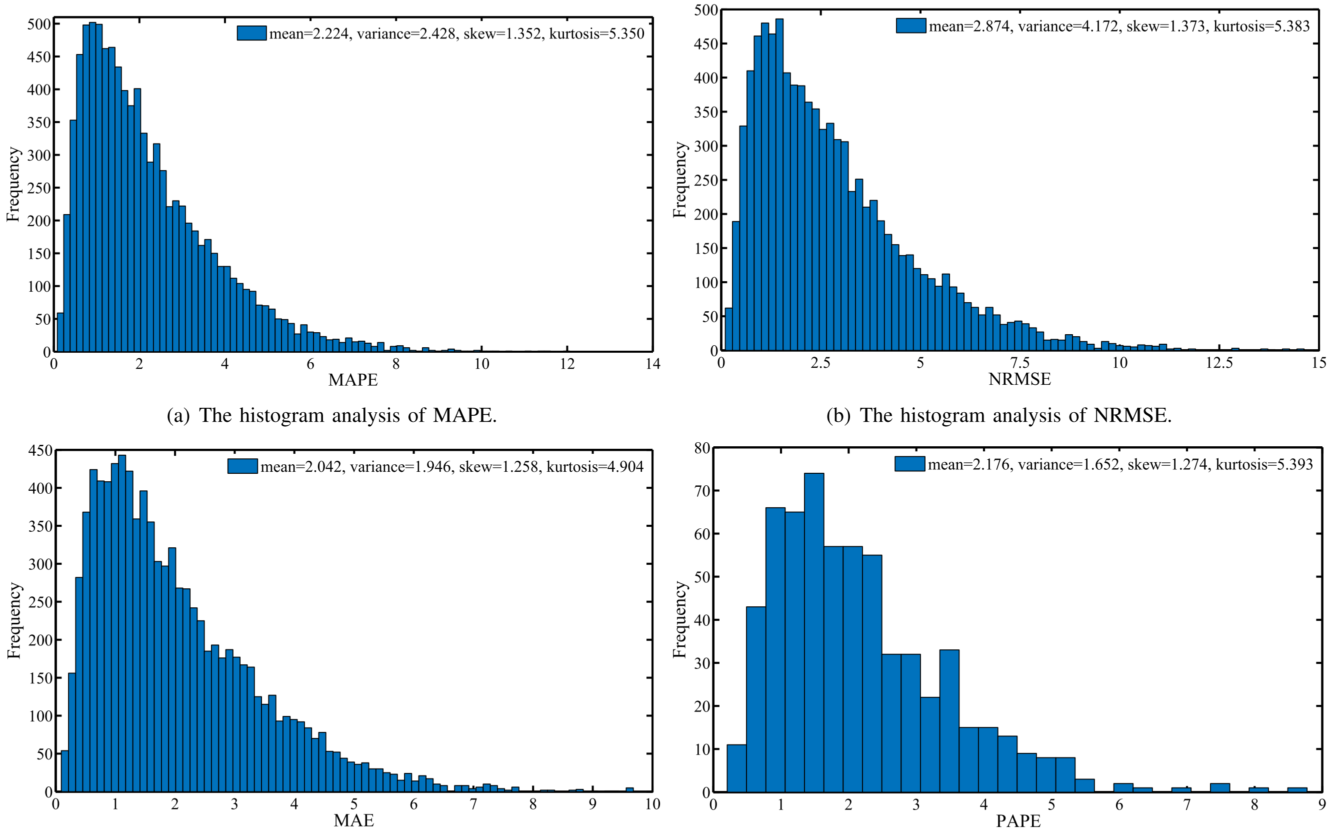
Next, we still choose the Test1 and Test6 as representatives and analyze the 5 minutes ahead peak demand forecasting performance of the models on all the test datasets, the results are shown in Fig. 10. In this figure, the peak demand values are sampled in every 30 minutes, so they are unevenly distributed along the horizontal axis. We can see that facing the strong volatility of industrial power load, our model fits the actual peak demand best on each dataset. Therefore, the results illustrate that proposed LSTM based hybrid ensemble strategy enables our model to have high accuracy and good generalization ability for peak load forecasting, which is very a key issue for maximum demand control in engineering.

In (3), we have described that the first part of the loss function is used to minimize the bias of model, and the second part is designed to minimize variance of model. Furthermore, to analyze the effect of the improved loss function for the peak demand forecasting, we trained our model using a basic MSE loss function

$$L(y, \hat{y}) = \underset{0<i\leq N}{mean}(y_i - \hat{y}_i)^2, \tag{8}$$

and compared the MAPE, NRMSE, MAE and PAPE metrics of our model trained by the basic loss function as (8) and the improved loss function as (3), respectively, the results are shown in Table VI. From the table we can observe that (3) is better than (8) in getting smaller PAPE error robustly, under the premise that the overall forecasting error is slightly smaller or comparable, this may be due to that (3) not only requires the overall average accuracy, but also reduces the peak demand forecasting error in each batch of training. From these results we can conclude that our method benefits from the improved loss function, and has excellent adaptability for the maximum demand control application that focuses on peak demand.

3) Histogram Analysis for the Errors: Furthermore, in order to describe the distribution of errors more intuitively, we conduct histogram analysis for the MAPE, NRMSE, MAE and PAPE errors on ten testing datasets from one to five minutes forecasting horizons. As it is shown in Fig. 11, we plot a histogram for



(a) The histogram analysis of MAPE.

(b) The histogram analysis of NRMSE.

(c) The histogram analysis of MAE.

(d) The histogram analysis of PAPE.

Fig. 11. The frequency statistical histograms of MAPE, NRMSE, MAE, and PAPE errors.

each kind of error. Several error statistics of histogram analysis, including mean, variance, skewness, kurtosis, are computed and shown in the figure.

From the figures we can see that both the mean and variance are relatively small for each kind of error, the mean is between 2.000 and 2.900, and the variance is between 1.600 to 2.500—except 4.172 for the NRMSE error, these metrics indicate the error distribution is concentrated and close to the mean. In addition, the error exhibits a certain positive skew and positive kurtosis distribution, and it can be found from the histograms that most of the errors are distributed on the left side of mean, which results indicate that the error distribution tends to be smaller than mean. From the error distribution we can conclude that our method is generally accurate and robust to meet the requirement of ultra-short-term power demand forecasting task in this work.

VI. Conclusion

In this work, a novel deep ensemble learning model was proposed to forecast ultra-short-term industrial power demand. Benefiting form the proposed hybrid ensemble strategy and the improved loss function, our model is superior to several state-of-the-art models in the commonly used accuracy metrics—MAPE, NRMSE and MAE—and the customized peak demand forecasting accuracy metric PAPE. Either on open dataset or the practical dataset, our model can always obtain the smallest forecasting errors, which represents our model is accurate and robust to multi-step forecasting problem in this paper. In next step, the model performance, especially peak demand forecasting accuracy for longer time horizon, needs to be further improved to support more advanced power demand control. Online adaptive learning could be another issue to be studied in further research.

References

[1] P. D. Diamantoulakis, V. M. Kapinas, and G. K. Karagiannidis, “Big data analytics for dynamic energy management in smart grids,” *Big Data Res.*, vol. 2, no. 3, pp. 94–101, 2015.

[2] M. Q. Raza and A. Khosravi, “A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings,” *Renew. Sustain. Energy Rev.*, vol. 50, pp. 1352–1372, 2015.

[3] 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, 2008.

[4] C. Hsu and C. Chen, “Applications of improved grey prediction model for power demand forecasting,” *Energy Convers. Manage.*, vol. 44, no. 14, pp. 2241–2249, 2003.

[5] G. Cerne, D. Dovzan, and I. Skrjanc, “Short-term load forecasting by separating daily profiles and using a single fuzzy model across the entire domain,” *IEEE Trans. Ind. Electron.*, vol. 65, no. 9, pp. 7406–7415, Sep. 2018.

[6] J. Zheng, C. Xu, Z. Zhang, and X. Li, “Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network,” in *Proc. IEEE Conf. Inf. Sci. Syst.*, doi: 10.1109/CISS.2017.7926112, 2017, pp. 1–6.

[7] H. S. Hippert, C. E. Pedreira, and R. C. Souza, “Neural networks for short-term load forecasting: A review and evaluation,” *IEEE Trans. Power Syst.*, vol. 16, no. 1, pp. 44–55, Feb. 2001.

[8] S. H. Ling, F. H. Leung, H. Lam, and P. K. Tam, “Short-term electric load forecasting based on a neural fuzzy network,” *IEEE Trans. Ind. Electron.*, vol. 50, no. 6, pp. 1305–1316, Dec. 2003.

[9] H. Jiang, Y. Zhang, E. Muljadi, J. J. Zhang, and D. W. Gao, “A short-term and high-resolution distribution system load forecasting approach using support vector regression with hybrid parameters optimization,” *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3341–3350, Jul. 2018.

[10] C. Cecati, J. Kolbusz, P. Rozycki, P. Siano, and B. M. Wilamowski, “A novel rbf training algorithm for short-term electric load forecasting and comparative studies,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 10, pp. 6519–6529, Oct. 2015.

[11] A. Almalaq and G. Edwards, “A review of deep learning methods applied on load forecasting,” in *Proc. IEEE Int. Conf. Mach. Learn. Appl.*, 2017, pp. 511–516.

[12] D.-C. Park, D.-M. Woo, and S.-S. Han, “Electric load forecasting using adaptive multiresolution-based bilinear recurrent neural network,” in *Proc. Image Signal Process., Congr.*, 2008, vol. 4, pp. 393–397.

[13] V. Mansouri and M. E. Akbari, “Efficient short-term electricity load forecasting using recurrent neural networks,” *J. Artif. Intell. Elect. Eng.*, vol. 3, no. 9, pp. 46–54, 2014.

[14] S. Kumar, L. Hussain, S. Banarjee, and M. Reza, “Energy load forecasting using deep learning Approach-LSTM and GRU in spark cluster,” in *Proc. IEEE Int. Conf. Emerg. Appl. Inf. Technol.*, doi: 10.1109/EAIT.2018.8470406, 2018, pp. 1–4.

[15] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, and Y. Xu, “Short-term residential load forecasting based on resident behaviour learning,” *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 1087–1088, Jan. 2018.

[16] D. L. Marino, K. Amarasinghe, and M. Manic, “Building energy load forecasting using deep neural networks,” in *Proc. IEEE Ind. Electron. Soc., IECON 42nd Annu. Conf.*, 2016, pp. 7046–7051.

[17] L. Kuan *et al.*, “Short-term chp heat load forecast method based on concatenated lstms,” in *Proc. IEEE Chin. Autom. Congr.*, 2017, pp. 99–103.

[18] W. He, “Load forecasting via deep neural networks,” *Procedia Comput. Sci.*, vol. 122, pp. 308–314, 2017.

[19] C. Fan, J. Wang, W. Gang, and S. Li, “Assessment of deep recurrent neural network-based strategies for short-term building energy predictions,” *Appl. Energy*, vol. 236, pp. 700–710, 2019.

[20] A. Dedinec, S. Filiposka, A. Dedinec, and L. Kocarev, “Deep belief network based electricity load forecasting: An analysis of Macedonian case,” *Energy*, vol. 115, pp. 1688–1700, 2016.

[21] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, and J. He, “Short-term load forecasting with deep residual networks,” *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3943–3952, Jul. 2019, doi: 10.1109/TSG.2018.2844307.

[22] A. Lahouar and J. B. Slama, “Day-ahead load forecast using random forest and expert input selection,” *Energy Convers. Manage.*, vol. 103, pp. 1040–1051, 2015.

[23] H. Zheng, J. Yuan, and L. Chen, “Short-term load forecasting using EMD-LSTM neural networks with a Xgboost algorithm for feature importance evaluation,” *Energies*, vol. 10, no. 8, pp. 1168–1188, 2017.

[24] X. Qiu, L. Zhang, Y. Ren, P. N. Suganthan, and G. Amaratunga, “Ensemble deep learning for regression and time series forecasting,” in *Proc. IEEE Comput. Intell. Ensemble Learn., Symp.*, 2014, pp. 1–6.

[25] X. Dong, L. Qian, and L. Huang, “Short-term load forecasting in smart grid: A combined cnn and k-means clustering approach,” in *Proc. IEEE Big Data Smart Comput. (BigComp), Int. Conf.*, 2017, pp. 119–125.

[26] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. London, U.K.: Chapman and Hall/CRC, 2012.

[27] G. Suryanarayana, J. Lago, D. Geysen, P. Aleksiejuk, and C. Johansson, “Thermal load forecasting in district heating networks using deep learning and advanced feature selection methods,” *Energy*, vol. 157, pp. 141–149, 2018.

[28] G. M. U. Din and A. K. Marnerides, “Short-term power load forecasting using deep neural networks,” in *Proc. IEEE Int. Conf. Comput., Netw. Commun.*, 2017, pp. 594-598.

[29] A. Ahmad, N. Javaid, M. Guizani, N. Alrajeh, and Z. A. Khan, “An accurate and fast converging short-term load forecasting model for industrial applications in a smart grid,” *IEEE Trans. Ind. Informat.*, vol. 13, no. 5, pp. 2587–2596, Oct. 2017.

[30] S. N. Fallah, R. C. Deo, M. Shojafar, M. Conti, and S. Shamshirband, “Computational intelligence approaches for energy load forecasting in smart energy management grids: State of the art, future challenges, and research directions,” *Energies*, vol. 11, no. 3, pp. 596–626, 2018.

[31] Y. Ren, P. N. Suganthan, N. Srikanth, and G. A. Amaratunga, “Random vector functional link network for short-term electricity load demand forecasting,” *Inf. Sci.*, vol. 367, pp. 1078–1093, 2016.