IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 11, 2020, accepted August 21, 2020, date of publication September 1, 2020, date of current version September 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3020799

Short-Term Industrial Load Forecasting Based on Ensemble Hidden Markov Model

YUANYUAN WANG[1], (Member, IEEE), YANG KONG[1], XIAFEI TANG[1], XIAOQIAO CHEN[2], YAO XU[3], (Senior Member, IEEE), JUN CHEN[1], SHANFENG SUN[1], YONGSHENG GUO[1], AND YUHAO CHEN[1]

[1]School of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha 410114, China
[2]Department of Computing and Mathematical Science, California Institute of Technology, Pasadena, CA 91106, USA
[3]Pacific Gas and Electric Company, San Francisco, CA 94105, USA

Corresponding author: Xiafei Tang (11875391@qq.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 51777014, in part by the Science and Technology Projects of Hunan Province under Grant 2018GK2057, in part by the Department of Education of Hunan Province of China under Grant 18A124, in part by the Changsha Science and Technology Project under Grant kq1901104, and in part by the Hunan Graduate Research and Innovation Project under Grant CX20190686.

ABSTRACT Short-term load forecasting (STLF) for industrial customers has been an essential task to reduce the cost of energy transaction and promote the stable operation of smart grid throughout the development of the modern power system. Traditional STLF methods commonly focus on establishing the non-linear relationship between loads and features, but ignore the temporal relationship between them. In this paper, an STLF method based on ensemble hidden Markov model (e-HMM) is proposed to track and learn the dynamic characteristics of industrial customer's consumption patterns in correlated multivariate time series, thereby improving the prediction accuracy. Specifically, a novel similarity measurement strategy of log-likelihood space is designed to calculate the log-likelihood value of the multivariate time series in sliding time windows, which can effectively help the hidden Markov model (HMM) to capture the dynamic temporal characteristics from multiple historical sequences in similar patterns, so that the prediction accuracy is greatly improved. In order to improve the generalization ability and stability of a single HMM, we further adopt the framework of Bagging ensemble learning algorithm to reduce the prediction errors of a single model. The experimental study is implemented on a real dataset from a company in Hunan Province, China. We test the model in different forecasting periods. The results of multiple experiments and comparison with several state-of-the-art models show that the proposed approach has higher prediction accuracy.

INDEX TERMS Short-term load forecasting, industrial customers, hidden Markov model, ensemble learning.

I. INTRODUCTION

Short-term load forecasting (STLF) is a key technology for smart grid [1]. Accurate STLF at system level aims to assist in power system infrastructure planning and system operation, while accurate STLF at demand side can be essentially useful for demand response (DR) [2], [3].

On the demand side, industrial customers that have a huge impact on smart grid consume a large proportion of electricity energy. Illustrated by the example of China, the electricity consumption of industrial customers accounts for about 70% of the total electricity consumption of the whole society [4]. For industrial customers, under the rules of China's electricity market, the electricity price during peak periods is almost twice as high as that during off-peak periods [5]. In China, industrial customers have to plan and purchase electric load quota in advance before consuming electricity. Actually, the required electric load quota is often determined by experience. This method leads to the problem of excessive or insufficient demand plan, which result in unnecessary wastage. Therefore, accurate STLF can guide industrial customers to determine reasonable production plans and electric power purchase plans, which can improve energy efficiency and reduce production cost during peak periods.

The associate editor coordinating the review of this manuscript and approving it for publication was Sotirios Goudos.

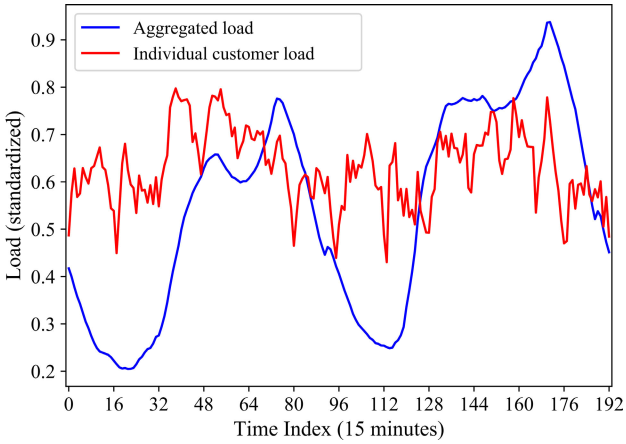


FIGURE 1. Compare the dynamic characteristics of the load curve at the aggregated level and the industrial customer level during 2 days.

Meanwhile, for Chinese suppliers and energy service companies, effectively STLF for individual industrial customers helps identify potential customers to participate in the DR plan and promote stable operation of the power system.

Regarding STLF methodologies, many approaches had been reported in the literature to address this problem, including time series models [6], [7], machine learning models [8]–[10] and deep learning models [11]–[13]. However, very few of them have confronted with individual industrial customers directly. Compared with the load at the system or substation level, it is often more difficult to forecast loads of individual industrial customers, and the prediction error will increase significantly, which is mainly due to the *effect of load aggregation* on forecasting performance [14]. For industrial customers, there are two main reasons for this dilemma. On the one hand, due to customers' random behaviors, the electricity load of individual customers is generally more volatile and therefore difficult to predict [14]. As the aggregated load increases, the loads at the system or substation level become smoother and therefore easier to predict, as shown in Fig. 1. On the other hand, industrial load will be affected by many complex factors, which may vary greatly from customer to customer [15].

In this paper, it aims to address the issues on STLF for the individual industrial customer. First, important factors are selected and added to the sliding time window to characterize the dynamics of the consumer's energy consumption patterns. Then, an industrial customer load forecasting framework based on ensemble learning and hidden Markov model (HMM) is proposed. Among them, with sliding time windows as input, a similarity measurement strategy of log-likelihood space is designed to help HMM capture dynamic information from multiple similar historical sequences. Furthermore, the Bagging algorithm is adopted to train multiple HMMs in parallel on different subsets and combine their prediction results. The proposed method is tested on a real dataset from a company in Hunan Province, China, and it outperforms other load forecasting approaches in different forecasting periods. The main contributions of this paper are as follows:

- **HMM prediction approach combining a novel time series data mining strategy.** A novel similarity measurement strategy of log-likelihood space is proposed, which enables the HMM to better capture the dynamic temporal characteristics of similar electricity consumption patterns. The sliding time window method is introduced to facilitate the HMM to utilize the dynamic temporal characteristics of multivariate time series. Experiments show that the novel strategy can effectively improve the prediction accuracy of the HMM.
- **Ensemble prediction framework integrating multiple HMMs.** To overcome the low generalization ability and instability of a single HMM, multiple HMMs are integrated into the Bagging algorithm framework to obtain an ensemble HMM (e-HMM) prediction method. The forecasting performance is determined by calculating the average of all base predictors. In addition, we introduce in detail the parameter optimization method of a single HMM. Experiments show that the forecasting performance of the integrated model is better than that of a single model.
- **Considerable overall accuracy comparing the proposed forecasting framework with the state-of-art methods.** The proposed method is tested in different forecasting periods on a real dataset. Different error metrics, namely average absolute percentage error (MAPE), root mean square error (RMSE) and average absolute error (MAE), are employed to prove the effectiveness of the proposed method. By comparing with the state-of-the-art models, we find that e-HMM outperforms all compared methods in terms of the three error metrics.

The rest of this paper is organized as follows: Section II reviews the related work of short-term industrial load forecasting. Section III is a brief introduction to the HMM and Bagging algorithm. Section IV introduces the dataset, feature selection and feature preprocessing in this paper. Section V includes the log-likelihood similarity measurement strategy, the hyperparameter selection of HMM and the e-HMM framework. Section VI is the experiment analysis. Section VII concludes the paper and discusses future work.

II. LITERATURE REVIEW OF STLF FOR INDUSTRIAL LOADS

Effective load forecasting techniques for industrial customers are gaining increasing interest. In the existing literature, the work of Domingo *et al.* [16] is the earliest example that focuses on load forecasting for individual industrial customers. They presented a neuro-fuzzy system with artificial neural networks (ANN) as well as time series process. However, the more commonly used metric of MAPE was not reported, which makes it hard to compare with other works. It is thereby unsuitable to serve as a benchmark for experimental comparisons. Li *et al.* [15] employed support vector regression (SVR) and random forest (RF) for forecasting loads of single industrial customer and found that the impact of holidays on forecasting accuracy was great, but detailed micro-indicators were not given. Ge *et al.* [17] adopted least squares support vector machine (LSSVM) to predict industrial load with several power consumption patterns, and then summarized the prediction results of all patterns. These consumption patterns were obtained by k-means clustering method from the historical load data of an industrial customer. However, how to determine the optimal number of consumption patterns is critical to predict results. In addition, the approach did not consider other influencing factors, which also limits the accuracy of the model. In the work of Wang *et al.* [18], several ensemble learning methods were introduced to improve the generalization ability and stability of a single ANN for aggregated load of industrial customers. Multiple experimental results showed that the ensemble learning method can improve the prediction accuracy of industrial loads. Their method is practical and we follow the idea of ensemble learning in our work to predict the load for a single industrial customer. In the load forecasting tasks of [19]–[21], the Bagging algorithm [22] was employed to improve the overall generalization ability of predictors, and obtained more accurate results than single predictor.

However, the common disadvantage of the above models is that the ability to mine the temporal relationships among continuous time series data is still insufficient [23]. Specifically, these methods only establish the non-linear relationship between loads and features, but ignore the temporal relationship between them, which limits their prediction accuracy. The customer's load sequences and corresponding features have dynamic non-linear characteristics, and their changes are a continuous process. The current load depends not only on the current features, but also on the previous loads and features. Clearly, it is critical to better mine the temporal relationship among the load sequences and corresponding features to characterize the consumption patterns of industrial customers for improving the accuracy of load forecasting.

With the development of deep learning, some researchers began to adopt deep learning algorithms, such as LSTM and CNN, to establish prediction models for industrial load [13], [24], [25]. Ungureanu *et al.* [24] used LSTM, ANN and RF to predict the hourly load of a single industrial customer within 27 days. The best performing model is LSTM, with a MAPE of 17.1%. Jiao *et al.* [25] employed LSTM to forecast the load of non-residential customers (including industrial customers). They classified each customer's electricity consumption patterns and analyzed the time correlation to enhance the LSTM's ability to capture dependencies between sequences. Compared with several machine learning models, LSTM performs best (MAPE ranges from 16.93% to 53.82%) in industrial customer cases. In [13], CNN was used to predict the loads of a single building. The results showed that the prediction accuracy of CNN was not only better than that of SVR, but also comparable to that of ANN and LSTM. However, these deep learning methods need to adjust multiple hyperparameters and are increasingly hard to train as the number of layer increases [26].

The HMM [27] is not only a nonlinear machine learning model with simpler model structure than many deep learning models, but also can mine the information from the time series, which have been effectively applied in many fields, such as speech recognition [28], stock prediction [29], equipment fault diagnosis [30], household appliance modeling [31], load data mining [32], load forecasting [33]–[35], etc.. In [33], researchers proposed a method that uses latent variables constructed by HMM to capture the electricity consumption behavior of household customers, and combined with a conditional Gaussian mixture model (CGMM) to improve the prediction accuracy. However, HMM is mainly used in this paper to generate latent variables to reflect customer consumption patterns, rather than forecast loads. In [34], researchers employed HMM to execute day ahead forecasts for data center load to assist in scheduling of available resources. Similarly, a discrete HMM was implemented on the data set provided by the New York Independent System Operator (NYISO) for 24 hours ahead load forecast [35]. Also, [34] and [35] both focused on system level forecasts and ignored the similarity of multiple load sequences in the consumption pattern sequences. They only used the transformation of consumption patterns between adjacent moments to predict, which makes the hidden information not be mined effectively.

In our work, we focus on the forecasts for individual industrial customer based on e-HMM. The reasons for choosing HMM are as follows [27], [32]: (a) its structure is simpler than that of neural networks; (b) fewer hyperparameters; (c) unique way to model dynamic temporal characteristics; (d) can learn customers' behaviors from their load series, transform the behaviors into states [33].

III. BRIEF THEORETICAL BACKGROUND

A. HIDDEN MARKOV MODEL

The HMM is a doubly stochastic model, in which the probability of the load value is conditioned on a small number of discrete (''hidden'') states representing the customer's random behavior, with Markovian transitions between them. Let $\mathbf{o}_t = (y, x^{(1)}, x^{(2)}, \ldots, x^{(d)})$ denotes a multivariate random observation vector for time t. The time subscripts of these variables are omitted for ease of presentation. Among them, y is the load value, x is the feature, and d is the dimension of the feature. Let Z_t denotes the hidden behavior state for time t. The interval between the two observation vectors can be naturally defined by the resolution of smart meters deployed in customers. Let $\mathbf{o}_{1:t}$ denotes the observation sequence $\{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_t\}$, and $Z_{1:t}$ denotes the hidden state sequence $\{Z_1, Z_2, \ldots, Z_t\}$.

In this paper, two conditional independence assumptions are made for load data to satisfy the rules of HMM [35]. The first assumption is that the observation vector $\mathbf{o}_t$ at time t is independent of all other variables before time t, conditional on the customer's behavior state Z_t at time t,

$$P\left(\mathbf{o}_t \middle| Z_{1:t}, \mathbf{o}_{1:t-1}\right) = P\left(\mathbf{o}_t \middle| Z_t\right) \tag{1}$$

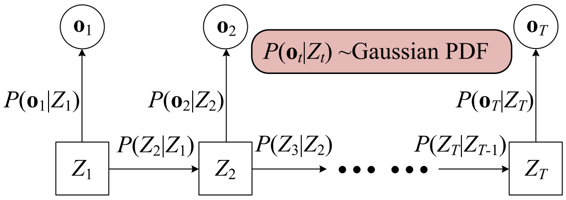


FIGURE 2. Graphical model representation of a continuous hidden Markov model.

The second assumption is that the hidden state process is first-order Markov process. The process means that the probability of the hidden state Z_t at time t depends only on the state Z_{t-1} at time $t-1$,

$$P\left(Z_t | Z_{1:t-1}\right) = P\left(Z_t | Z_{t-1}\right) \tag{2}$$

Fig. 2 is a directed graph of HMM with the conditional independence assumptions. To avoid the errors caused by vector quantization of continuous variables, continuous HMM [27] is adopted in this paper, where the probability of the observed value emitted by the state is represented by the probability density function (PDF).

Here are three elements that describe an HMM via those assumptions:

$$\begin{cases} \mathbf{A} = \left\{P\left(Z_t = q_i | Z_{t-1} = q_j\right)\right\} \\ \mathbf{B} = \left\{P\left(\mathbf{o}_{t-1} | Z_{t-1} = q_j\right)\right\} \\ \mathbf{\Pi} = \left\{P\left(Z_1 = q_i\right)\right\} \end{cases} \tag{3}$$

where $\mathbf{A}$ is the transition probability matrix that represents the change between the hidden states, i.e. q_i and q_j; $\mathbf{B}$ is the observation probability matrix that represents the relation between the hidden state and the observation; $\mathbf{\Pi}$ is the initial vector that represents probability of a certain state.

B. BAGGING ALGORITHM

Bagging is a method to construct independent base learners [36]. The base learner can be a classifier or predictor. This method relies on resampling from the original training data to ensure the independence of base learners. Given the training data of N samples, in each iteration, the new training subset will be generated via uniform sampling. The sample size of each subset is the same as that of the original training set. Some samples may be resampled repeatedly, while others may not. If the predictor is unstable, i.e. a small change in the training data leads to a large change in the generated predictor, then Bagging will result in a diverse hypothesis set. Oppositely, Bagging may not be any better than the base predictor.

IV. CUSTOMER LOAD DATA AND INPUT FEATURES PROCESSING

A. DATASET

The dataset, collected from a printed circuit board industry in Hunan province, China, contains a total of 365 days of records of smart meters from March 1, 2018 to March 1, 2019 with the resolution of 15 minutes. There are 35,040 records in this dataset.

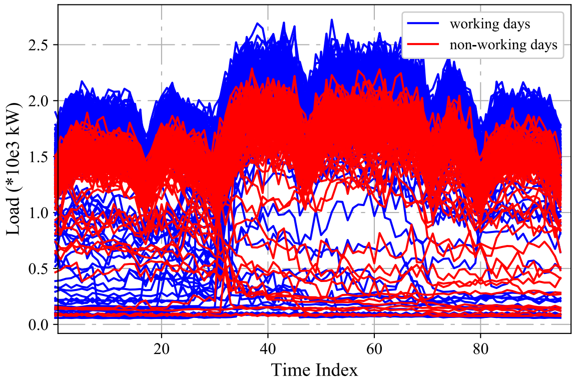


FIGURE 3. Daily load curves (365 days) of a printed circuit board industry consumer.

Fig. 3 shows the daily load curve of this customer. The following conclusions can be drawn from Fig. 3:

(1) Generally, the law and trend of load fluctuations are similar, regardless of working days or non-working days. The daily load power is usually between 1000 kW and 2500 kW. However, there is a large gap in the load value of the same sampling point on different days, which indicates that the customer's consumption pattern is affected by the date type.

(2) The daily load curve fluctuates greatly with 6 peaks and 6 valleys in a day. The largest peak period of the day usually occurs from 7:30 to 11:30 or from 13:30 to 16:30.

(3) The load curve for a few days do not follow any trend at all. Instead, they present chaotic characteristics. This shows that only carrying out load modeling according to general rules will lead to large errors in load prediction on special dates.

B. MISSING VALUE AND OUTLIER PROCESSING

Missing values account for 0.419% of the dataset. For the missing data, linear interpolation is used to process these data. Moreover, when the missing value exceeds 10% per day, all data for that day will be eliminated.

Due to the influence of random factors (such as unexpected events or drastic temperature changes), outliers sometimes appear in the load data. These outliers will disturb the regularity of the whole data sequence and affect the prediction accuracy. Therefore, it is necessary to correct the outliers. In this paper, Grubbs criterion [37] is used to detect outliers, while linear interpolation is used to correct outliers.

C. FEATURE SELECTION

Some electrical variables (i.e., current, voltage, power factor, etc.) recorded by smart meters can be regarded as inputs of the prediction model. The load to be predicted is determined by the electrical variables, previous loads, and exogenous variables (usually temperature, date type, seasonal patterns) [9]. In this paper, we consider three kinds of features: electrical feature, date feature and meteorological feature.

(1) For electrical features, we select all seven features collected from smart meters, including load, reactive power, active energy, reactive energy, current, voltage and power factor.

TABLE 1. Grey correlation degree between electrical features and load.

Electrical Variable	Correlation degree
Reactive power	**0.998**
Active energy	**0.922**
Reactive energy	**0.912**
Current	**0.974**
Voltage	0.666
Power factor	0.729

These electrical variables form a multivariate time series in the same system, and there are always complex correlations between the multivariate time series.

In the multivariate time series, correlation is used to describe the degree of relationship between two random variables (also called features). Correlation analysis is very important to reduce redundant features, decrease computational complexity and improve prediction performance [38], [39]. Typical correlation analysis methods include principal component analysis (PCA) [40], canonical correlation analysis (CCA) [41], Granger causality analysis [42], Pearson correlation coefficient [43], mutual information (MI) [44], and grey relational analysis (GRA) [45]. However, PCA and CCA extract original variables to obtain new variables. The physical meaning of these new variables is difficult to explain. Granger causality analysis only supports qualitative description of variables. Pearson correlation coefficient can provide a quantitative description of variables. But the method cannot measure the nonlinear relationship between variables. MI overcomes the shortcomings of Pearson correlation coefficient, and it has no requirement of variable distribution. However, MI is computationally complex and requires more computation time for the higher-dimensional input variable. Compared with the aforementioned correlation analysis methods, the main advantages of GRA are [45]: (a) can provide qualitative and quantitative description of variables; (b) can measure the linear and nonlinear relationships between variables; (c) there are no special requirements for the size and distribution of variables. Therefore, GRA has been applied to many fields [46]. In order to select features that have significant impact on the loads as input features, GRA is utilized to calculate the correlation between the electrical features and the load.

Among them, the load sequence is regarded as the reference sequence. The grey correlation degree can be defined according to the distance in n-dimensional space [45]. In the process of variable sequence, if the trends of two variables are consistent, the correlation degree between them will be higher. On the contrary, the correlation degree will be lower [47]. When the correlation degree is greater than 0.7, it indicates certainly correlation. when the correlation degree is greater than 0.8, it indicates highly correlation [45]. Table 1 shows the results of the grey correlation degree between the above electrical variables and the load.

As can be seen from Table 1, when the grey correlation degree is higher than 0.8, the variable has a high correlation with the load. Thus, reactive power, active energy, reactive energy, and current are selected as the input features, while voltage and power factor are deleted.

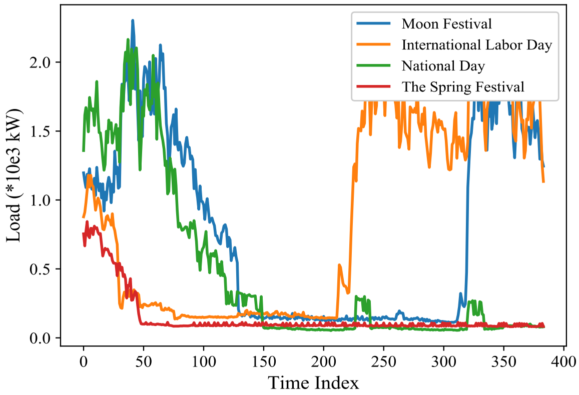


FIGURE 4. Load changes during the different holidays.

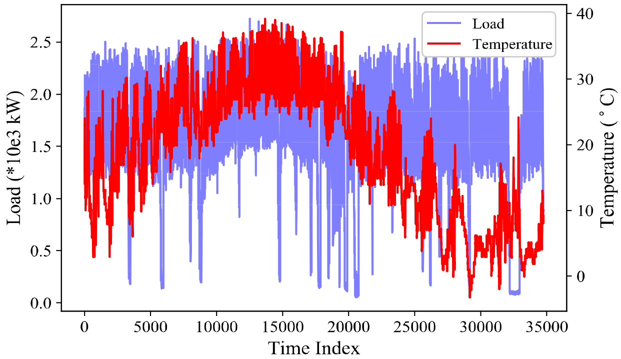


FIGURE 5. The impact of temperature on industrial customer load.

(2) For date features, from the above analysis in Fig. 3, we find that some data with lower daily load are not abnormal data. Just because the date type is a holiday and the customer's electricity consumption throughout the day is much lower than usual. As shown in Fig. 4, we profile the load changes during the Moon Festival, International Labor Day, National Day and the Spring Festival, and we can see that the customer's load is also affected by different holidays. Therefore, the date feature (i.e. working days or non-working days, and different holidays) should be considered in the industry customers load forecasting.

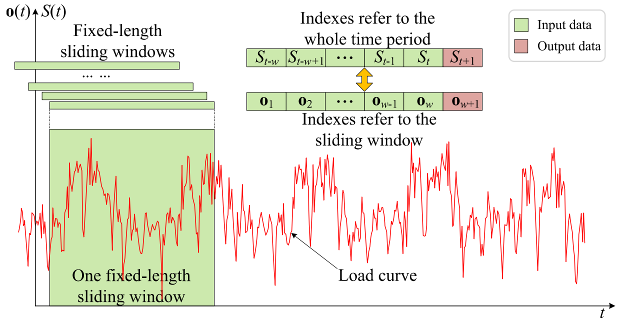
(3) For meteorological features, we acquire public temperature data in Hunan Province provided by the National Oceanic and Atmospheric Administration (NOAA), as shown in Fig. 5. From Fig. 5, we can see that the temperature has little effect on the customer load. The gray correlation degree between temperature and load is calculated, and the result is 0.713. Because most of the electric energy consumed by the customer is used to produce products, not air conditioners. This indicates that some common exogenous variables applied to the aggregated load may not be applicable to the industrial loads.

D. FEATURE PREPROCESSING

In this paper, the features selected via Section IV-C for industrial customer load forecasting are shown in Table 2.

TABLE 2. Features and preprocessing method.

Type of feature	Feature name	Symbol	Preprocessing Method
Electrical features	History load	y	Reconstructed by sliding time window approach and normalized by Min-Max normalization method
	Reactive power	Q	
	Active energy	P_E	
	Reactive energy	Q_E	
	Current	I	
Date features	Working day or non-working day	W_d	One-Hot Encoding
	Holiday	H_d	



$o(t)$ $S(t)$
Fixed-length sliding windows
... ...
Indexes refer to the whole time period
S_{t-w} S_{t-w+1} ⋯ S_{t-1} S_t S_{t+1}
o_1 o_2 ⋯ o_{w-1} o_w o_{w+1}
Indexes refer to the sliding window
🟩 Input data
🟥 Output data
One fixed-length sliding window
Load curve
t

FIGURE 6. A sliding time window approach.

Moreover, Table 2 illustrates the preprocessing methods for different types of features.

Since the load data has time series characteristics, all the electrical features in Table 2 changed over time. In order to better make the prediction model look back to the past to extract dynamic temporal characteristics, we adopt a sliding time window method [48] to generate shifted learning data and group different data vectors in historical moments. Fig. 6 depicts the construction of the sliding time window and shift space.

In this task, a multivariate time series $S(t) = \{S_1, S_2, \ldots, S_T\}$ that spans through the whole time period is given. For example, S_t represents a data vector at time t, this vector contains load data and feature data. $\mathbf{o}_t$ is the observation vector also contains load data and feature data at time t in the reference system of the sliding time window. The input data are presented to the prediction model as regression vectors consisting of fixed time-lagged data associated with a window size of length w which slides over the time series. Given the fixed-length view of past values, the goal of the predictor is to predict the load value at the next moment. Therefore, the prediction problem is turned into a supervised learning problem. Thus, given the input sequence at discrete time t defined as $\{\mathbf{o}_{t-w}, \mathbf{o}_{t-w+1}, \ldots, \mathbf{o}_{t-1}, \mathbf{o}_t\} = \{S_{t-w}, S_{t-w+1}, \ldots, S_{t-1}, S_t\}$, the predictor needs to estimate the load value $\mathbf{o}_{t+1}(y) = S_{t+1}(y)$ corresponding to the vector at time $t+1$. To simplify the notation, we represent the input and output in the reference system of the sliding time window, rather than the entire time series. By following this approach, the input sequence of discrete time t becomes $\{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_{w-1}, \mathbf{o}_w\}$. The corresponding output is $\mathbf{o}_{t+1}(y) = \mathbf{o}_{w+1}(y)$. Once the data in the old window has been learned, the new window is generated by deleting the oldest data $\mathbf{o}_1$ and adding the latest data $\mathbf{o}_{t+1}$, while remaining the window length at w. The next forecast value will be $\mathbf{o}_{t+2}(y)$. Similarly, this input and output method will be presented to different prediction model later. For convenience, the univariate sequence (load curve) of the time series is shown in Fig. 6. In fact, the observation sequence is a multivariate time series.

V. ENSEMBLE HIDDEN MARKOV MODEL FOR STLF

A. SIMILARITY MEASUREMENT STRATEGY FOR HMM PREDICTOR

Given the fixed number of hidden states K (details shown in Section V-B), we learn the parameters $\boldsymbol{\Theta} = (\mathbf{A}, \mathbf{B}, \boldsymbol{\Pi})$ of the HMM by searching for the parameters to best fit the observed data. To do this, we chose the commonly used Maximum Likelihood (ML) criterion. Specifically, search for $\boldsymbol{\Theta}$ to maximize the conditional probability of the observed data. The result of this conditional probability function is called log-likelihood:

$$L(\boldsymbol{\Theta}) = \sum_{Z_{1:T}} P(Z|\mathbf{o}_{1:T}, \boldsymbol{\Theta}) \log\left[P\left(o_{1:T}, Z|\hat{\boldsymbol{\Theta}}\right)\right] \quad (4)$$

We use the commonly used Expectation Maximization (EM) algorithm to iteratively calculate the maximum likelihood in Formula (4). The HMM parameter fitting technique is well known in the statistical literatures. Readers can refer to the standard references [27] and [31] for details. Therefore, there is no need to repeat here.

As previously mentioned in Section III-A and Section IV-D, let $\mathbf{o}_t = (y, Q, P_E, Q_E, I, W_d, H_d)$ denotes the observation vector at time t. In this paper, the width of the fixed-length time window is set to 96, i.e. the previous 96 observation vectors are used as input features of the next load. In addition, the sequence within each time window is also used to calculate the log-likelihood.

Specifically, the similarity measurement strategy of log-likelihood space is divided into the following steps:

•**Step 1**: Use the historical observation data (total length is T) $\mathbf{o}_{1:T}$ as the input of the HMM, and then the optimal parameter $\boldsymbol{\Theta}_e$ of HMM for this sequence can be obtained by Formula (4).

•**Step 2**: Put the data in the updated sliding time window into the set $\{\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_T\} = \{\mathbf{o}_{1:w}, \mathbf{o}_{2:w+1}, \ldots, \mathbf{o}_{T-w+1:T}\}$.

•**Step 3**: Use the fitted HMM from **Step 1** to calculate the probability of the observed sequence in $\mathbf{W}_i$, and use its natural logarithm (to the base e) to obtain the log-likelihood set $\{l_1, l_2, \ldots l_T\}$. The formula for l_i is as follows

$$\begin{aligned} l_i &= \ln\left[\sum_{Z_{1:w}} P(\mathbf{W}_i, Z|\boldsymbol{\Theta}_e)\right] \\ &= \ln\left[\sum_{Z_{1:w}} P(\mathbf{W}_i|Z, \boldsymbol{\Theta}_e) P(Z|\boldsymbol{\Theta}_e)\right] \end{aligned} \quad (5)$$

•**Step 4**: Calculate the distance between all l_i and l_T by Formula (6), and the log-likelihood similarity vector

$\mathbf{V}_\mathrm{L} = (\mathrm{D}_1, \mathrm{D}_2, \ldots, \mathrm{D}_{T-1})$ can be calculated by:

$$D(i, T) = \sqrt{(l_i - l_T)^2} \tag{6}$$

•**Step 5**: Rank the elements in $\mathbf{V}_\mathrm{L}$ from small to large, and select the top n values to consider the data under the most similar n windows.

•**Step 6**: Record and extract the index of the sliding window corresponding to the top n values in **Step 5** on the whole time period to obtain the load forecast value at time $T + 1$ by Formula (7).

$$y_{T+1} = y_T + \frac{1}{n}\sum_{i=1}^{n} (\mathbf{W}_i(y_{w+1}) - \mathbf{W}_i(y_w)) \tag{7}$$

In order to solve the problems of high computational complexity and prediction accuracy of the forecasting model, in this paper, the number of similar windows is $n = 5$.

B. HYPERPARAMETERS OPTIMIZATION

Determining the most appropriate parameters of the HMM is a complex task since such parameters interact with each other in a highly nonlinear manner. The number of hidden states expressed by K is an important hyperparameter for HMM. In this paper, Bayesian Information Criterion (BIC) and cross-validation log-likelihood described in literature [49] are used to evaluate the quality of the fitted HMM. $E_{BIC.K}$ and $E_{CV.K}$ represent the scores of the above two methods, respectively:

$$\begin{cases} E_{BIC.K} = -2L(\mathbf{\Theta}_K) + p\log N \\ E_{CV.K} = -\dfrac{\sum_{i=1}^{fold} L_i(\mathbf{\Theta}_K)}{N} \end{cases} \tag{8}$$

where $\mathbf{\Theta}_K$ is the estimated maximum likelihood parameter vector found by EM on the training data for a model with K hidden states; $L(\mathbf{\Theta}_K)$ and $L_i(\mathbf{\Theta}_K)$ are the log-likelihood calculated by Formula (4); p is the number of parameters in the K-state model; N is the total number of observation samples used to train the model; *fold* is the number of folds in cross-validation.

A composite score is used, which is a weighted combination of the two scores:

$$E_{CS.K} = \frac{E_{ij}\sqrt{\sigma(E_j)}}{\sum_{j=1}^{2}\sqrt{\sigma(E_j)}} \tag{9}$$

where E_{ij} is the i-th sample of the j-th score (i.e. E_j is E_{BIC} or E_{CV}), $\sigma(E_j)$ is the variance of the score. Note that E_j in Formula (9) needs to be normalized to the range of (0, 1).

In this paper, samples between March 2018 and January 2019 (15-minute resolution) are used for training the load forecasting model ($N = 96 \times 336$). Four-fold cross-validation is adopted, i.e. leaving a quarter of data as test data once (84 consecutive days). For different K, we use the EM algorithm 10 times from different random starting positions in the parameter space to calculate Formula (8) and Formula (9) to avoid convergence to the local maximum. The performance evaluations of HMMs with different K are given in Table 3 for $K = 2, 3, \ldots, 10$.

TABLE 3. Performance evaluations of HMMs with different numbers of hidden state.

State K	$E_{BIC.K}$	$E_{CV.K}$	$E_{CS.K}$
2	7.52×10^8	58.14	0.341
3	6.37×10^8	62.74	0.186
4	6.41×10^8	59.51	0.126
5	$\mathbf{6.21\times10^8}$	55.92	**0.014**
6	9.68×10^8	**55.23**	0.647
7	9.67×10^8	62.38	0.794
8	8.39×10^8	63.92	0.588
9	9.67×10^8	72.18	0.998
10	8.94×10^8	66.67	0.747

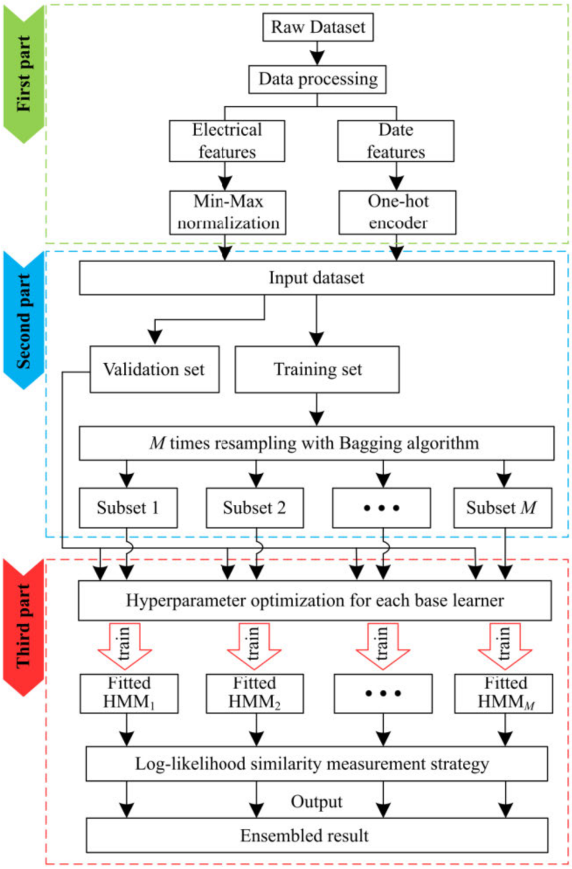


FIGURE 7. Framework of ensemble HMM.

It can be seen from Table 3 that the optimal result obtained by BIC is $K = 5$, while the optimal result obtained by cross-validation is $K = 6$. Combining the above two judgments, the reasonable result obtained by the comprehensive score is $K = 5$. Therefore, $K = 5$ is selected as the hyperparameter for the non-integrated HMM (underlying model and Enhanced model).

C. ENSEMBLE HMM LOAD FORECASTING

The framework of the proposed ensemble hidden Markov model (e-HMM) is shown in Fig. 7, which consists of three parts. The first part is the preprocessing of data and features, which is described in Section IV.

The second part is the sampling of the input data. In order to ensure the continuity of the series during sampling, we cannot directly resample each sampling point as a unit, so we adopt the following method. First, the training set of the input data is divided into many small data sequences (sample size is 96 × 7), called "slices", i.e. each data sequence contains all samples for 7 days. Then, taking these "slices" as the units, the raw dataset is resampled M times using Bagging algorithm to obtain multiple data subsets whose sample size does not exceed the input data.

The third part is model training and prediction. First, the resampled subset is used to train the parameters of a single HMM predictor (i.e. base learner). Specifically, the EM algorithm is used to train the parameters $\boldsymbol{\Theta}$ of the base learner to better fit the data. The two indicators, BIC and cross-validation method, are combined to determine the appropriate number of hidden states K. Then, the fitted HMM is used to calculate the log-likelihood of the data within the sliding time window. The similar log-likelihood measurement strategy is used to find similar window observations, then these similar observations is used to calculate future load values. Finally, the results of all base learners are aggregated and their average value is taken to obtain the final load forecasting result.

D. PERFORMANCE METRICS

The widely adopted error metrics are used to evaluate the load forecasting performance of the model. Specifically, we use MAPE, RMSE and MAE. The above error metrics are briefly described as follows:

$$\mathrm{MAPE} = \frac{1}{N}\sum_{t=1}^{N}\frac{|y_t - \hat{y}_t|}{y_t} \times 100\% \tag{10}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2} \tag{11}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{t=1}^{N}|y_t - \hat{y}_t| \tag{12}$$

where N is the number of samples; y_i is the real load at time t; $\hat{y}_i$ is the predicted load at time t.

VI. EXPERIMENTS ANALYSIS

A. EXPERIMENTAL DATA

In this experiment, the dataset is collected from a printed circuit board industry in Hunan province, China. The time span of the data is 362 days and the sampling rate is 15 minutes. The data are split into three parts: the training set (from 1-March-2018 to 8-November-2018), the validation set (9-November-2018 to 31-January-2019), and the test set (1-February-2019 to 28-February-2019). For the proposed method, the training set and the verification set are combined together to be used as the hyperparameter K optimization.

B. THE BENCHMARKS

In order to verify the performance of the proposed prediction method, we chose 2 classic machine learning algorithms (SVR and RF) and 2 deep learning algorithms (CNN and LSTM) as the benchmarks because their prediction performance was proved to be excellent in previous work [8], [9], [12], [13]. The implementations of SVR and RF are based on the scikit-learn package. CNN and LSTM are realized by the Keras deep learning package. Other HMM-based models are implemented in the hmmlearn package. All experiments are completed based on the Python 3.7 programming language. The hardware is a personal computer with Intel core i7-9700k and 16GB of memory.

For SVR, grid search method is used to select two important hyperparameters: penalty parameter and kernel coefficient. For RF, most of the parameters are default values, except for slightly adjusting the number of base estimator in the previous work. Grid search method and some rules of thumb are adopted to select some important hyperparameters of CNN and LSTM, and other parameters are default values. The hyperparameters of each model used in the experiment are set as follows:

•**SVR**: Two kernel functions, linear and RBF, are used. The optimal penalty parameter of the two SVRs is 2.0, and kernel coefficient of the RBF kernel is 0.001. Other parameters are default values.

•**RF**: Decision tree is used as a base estimator. In order to obtain a balance between the diversity of a single tree and the speed of the algorithm, the maximum feature is set to 50% of the total number of features. According to the suggestion of [25], the number of trees in the random forest is set to {100, 300, 500}, which is expressed by RF-100, RF-300 and RF-500 respectively. The maximum depth and other parameters are the default values.

•**CNN**: Previous work has shown that hidden features can be extracted by the designed one-dimensional (1D) convolutional layers [13]. Therefore, the number of convolutional layers in this experiment is 1. The optimal number of filters is 128, and the optimal kernel size is 5. The number of fully connected layers is 4 by rules of thumb, which is 2 more layers than the fully connected layers in [13]. The number of neurons in each layer is also set to 100/100/100/50 by grid search method, and the search range of the number of neurons in each layer is {50, 100}. In our experiments, the pooling layer caused information loss and reduced accuracy, so the pooling layer is not added. After parameters tuning, the epoch is 200, and the batch size is 256. Adam is the optimizer in the training process. The loss function is MAE.

•**LSTM**: According to the consistent findings in [11], multiple layers generally perform better than single layers (usually 1 to 3 hidden layers in load forecasting tasks [11], [12], [24], [25]), and the number of hidden nodes should be sufficient. The LSTM architecture is selected according to the recommendations in [12]. Specifically, the number of hidden layers is 3. Then, the order of hidden nodes is 50/100/100, which is the optimal result found in the range of {50, 100} via grid search method. The optimal hidden nodes are 50/100/100 in order. The optimal dropout rate is equal to 0.1, 0.2 and 0.2. After the parameters tuning, the epoch is 150,

TABLE 4. Load forecasting results of different models in different forecast periods.

Algorithm/ Scenario	Day ahead (1-February-2019)			Week ahead (1-February-2019 to 7-February-2019)			Month ahead (1-February-2019 to 28-February-2019)		
	MAPE (%)	RMSE (kW)	MAE (kW)	MAPE (%)	RMSE (kW)	MAE (kW)	MAPE (%)	RMSE (kW)	MAE (kW)
SVR-linear kernel	8.99	130.73	102.66	5.90	129.34	103.15	44.79	128.84	109.34
SVR-RBF kernel	10.09	140.42	112.70	6.05	130.65	104.65	64.84	150.77	126.74
RF-100 estimators	7.02	109.81	84.93	5.09	115.69	89.96	16.80	102.31	76.35
RF-300 estimators	6.89	108.86	82.99	5.03	114.64	88.83	14.39	101.12	74.06
RF-500 estimators	6.88	108.95	82.95	5.04	114.45	88.97	13.81	100.88	73.57
CNN	6.83	112.66	84.87	5.49	128.14	97.52	10.79	112.50	78.12
LSTM	6.82	106.80	80.98	5.11	116.83	90.87	9.90	99.06	69.91
Underlying HMM	9.36	144.04	111.85	17.31	305.91	524.99	35.64	290.14	156.70
Enhanced HMM	4.69	66.93	53.02	6.27	156.64	110.80	17.38	102.87	76.79
e-HMM	**3.13**	**46.37**	**35.56**	**4.98**	**112.62**	**88.11**	**7.07**	**69.39**	**35.59**

and the batch size is 256. Adam is used as the optimizer. The loss function is MAE.

•**Underlying HMM**: The number of hidden states is equal to 5, which is explained in detail in Section V-B. The log-likelihood similarity measurement strategy is not used, i.e., the number of similar windows is $n = 1$.

•**Enhanced HMM**: The number of hidden states is equal to 5. The log-likelihood similarity measurement strategy is used, and the number of similar windows is $n = 5$, which is explained in detail in Section V-A.

•**e-HMM**: Considering the number of training samples, the sampling iteration in this paper is set to 50, which is the recommendation on the number of bagging iterations in [19]. The base learner adopts the log-likelihood similarity measurement strategy, and the number of similarity windows is $n = 5$. The number of hidden states K of different base learners is determined by BIC and cross validation.

C. EXPERIMENTAL RESULTS AND DISCUSSION

We perform predictions for the individual industrial customer in different forecasting periods, as shown in Table 4. In general, the forecasts for a single customer are not as accurate as the forecasts for aggregated loads. Because the prediction errors of individual customer could not be offset by the diversity of different users. According to the reports in previous studies [17], [24], [25], the prediction accuracy will decrease significantly with the decrease of the aggregation level. Therefore, the results shown in Table 4 are reasonable. From Table 4, we can make the following observations:

(1) In different forecasting periods, the prediction performance of machine learning algorithms, such as SVR, RF and non-integrated HMM, is generally lower than that of deep learning algorithms. Among them, compared with SVR and HMM, the prediction accuracy of RF is higher, and sometimes even slightly better than that of CNN and LSTM. Because the ensemble learning architecture improves the generalization of a single decision tree.

(2) The prediction performance of the underlying HMM is the lowest among all algorithms. Because this model only considers the historical information in a single sliding window, it cannot accurately track future changes. This also indicates that the underlying HMM is an unstable predictor.

(3) The enhanced HMM uses the log-likelihood similarity measurement strategy proposed in this paper to make the model consider the historical information in multiple similar sliding windows. Obviously, the prediction accuracy of single HMM is greatly improved.

(4) The e-HMM model shows the highest prediction accuracy in different prediction periods. On the one hand, this is due to the ensemble learning framework. On the other hand, the hyperparameter of a single predictor are reasonably determined. Therefore, each base learner can be trained more efficiently under different training sets.

Fig. 8 shows statistical plots for daily, weekly and monthly prediction errors, with a time step of 15 minutes. In each box, the red horizontal central line is the median, the edges are 25th and 75th percentiles, and outliers are plotted using '+' sign. From Fig. 8, we can get the following information:

(1) For daily and weekly predictions, the overall error variation using e-HMM is the least, followed by the Enhanced HMM, CNN, and RF-500. This can be seen from the maximum percentage error on the y-axis of each graph. In the daily forecast, the maximum percentage error of RF-500 is about 28%, the Enhanced HMM is 18%, and the e-HMM is 15%. In the weekly forecast, the maximum percentage error of CNN is 28%, the Enhanced HMM is 27%, and the e-HMM is 22%.

(2) In the monthly forecast, the maximum percentage error of all methods is significantly larger than that of the daily and weekly forecasts. There are two main reasons for this result: (a) the sample resolution is too different from the time window length, and the adaptability of the model will be reduced. (b) the customer's random activity will change over time, and the expansion of time window will increase the random factors.

(3) Compared with other methods, the median of e-HMM is the least, which can be seen from the horizontal lines of different forecast periods. With smaller spread (the range

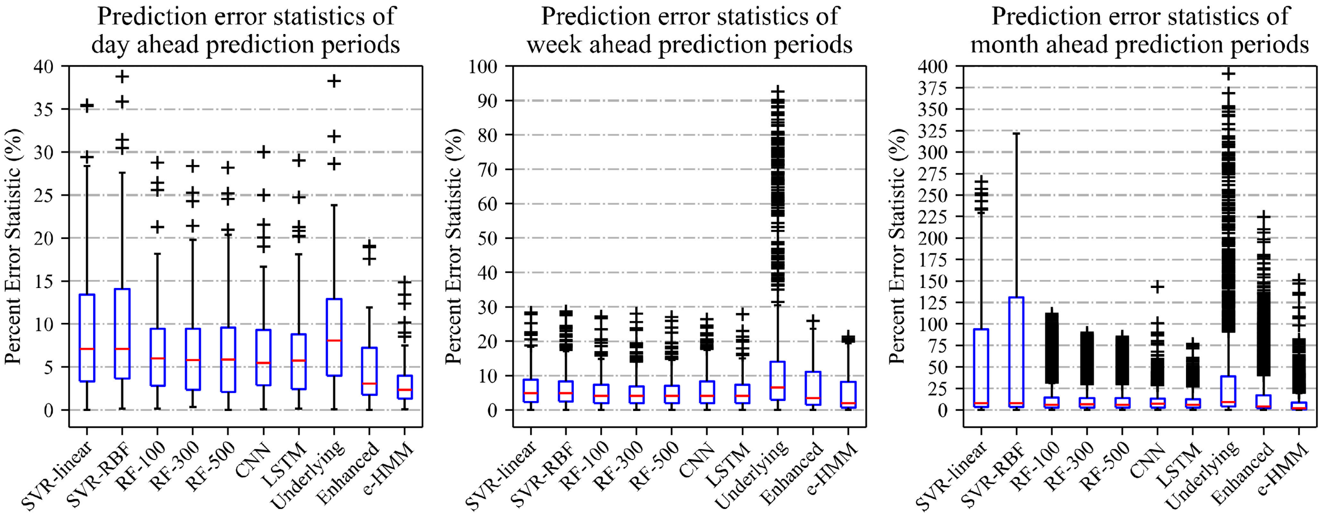


FIGURE 8. Comparison of day ahead, week ahead and month ahead prediction error statistics using different models.

of interquartile quartile), the prediction performance of the model is more stable. The spread of day ahead and month ahead of e-HMM are smaller than that of other models, while the spread of week ahead is significantly higher than that of other models. This is because the proposed model estimates the power consumption trend via the similarity of abstract electrical behaviors of users within the predicted time scale. For this customer's load sequence, the weekly behavioral similarity is lower. Moreover, the outliers of the predicted results cannot be ignored. Fewer outliers can indicate that external factors have less influence on the model. However, the e-HMM generated more outliers in the monthly prediction than daily prediction and weekly prediction. This is because: (a) with the increase of time scale, the random behavior will increase; (b) e-HMM does not have memory cell structure like LSTM, leading to the gradual accumulation of outliers. Nevertheless, the overall forecasting performance of the e-HMM is still better than the other models.

We plot the prediction curves of different models one day in advance, as shown in Fig. 9. Clearly, e-HMM performs better than any other comparison algorithms, followed by the enhanced HMM, then LSTM and CNN. The results of curve analysis are consistent with the previous error analysis. This result benefits from three aspects: (a) the log-likelihood measurement strategy enables the HMM to mine more information; (b) the Bagging ensemble framework improves the stability of a single HMM predictor and makes the prediction results smoother; and (c) the hyperparameter selection strategy makes the training of HMM more efficient. The relatively large prediction errors of LSTM are mainly due to the following two points: (a) the forecasting accuracy tends to drop significantly as the level of aggregation decreases (i.e. the effect of load aggregation). Literature [24] and [25] showed that the range of MAPE obtained by LSTM for industrial load forecasting was between 16.93%-53.82%; (b) the input of the reference sequence profile becomes a dominant feature of the LSTM, which may cause large errors. In other words, if the LSTM correctly decomposed the periodic features, its accuracy would be high; but if there was an error in the feature, the prediction errors would increase due to the memory cell structure.

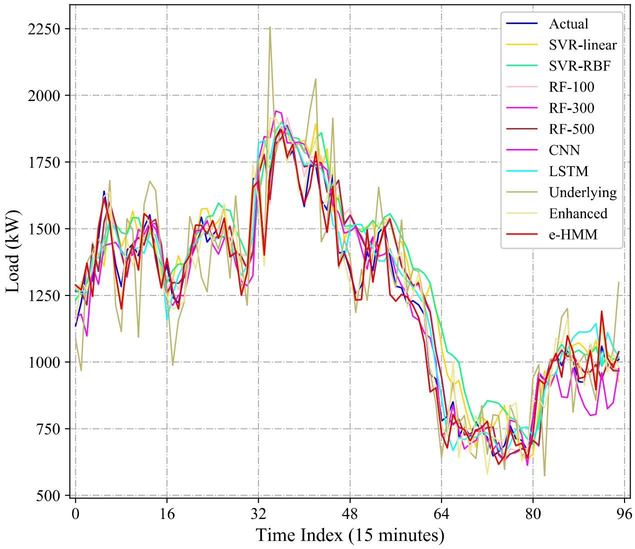


FIGURE 9. The day ahead forecasting load curve using different models.

VII. CONCLUSION AND FUTURE WORK

Accurate STLF could help industrial customers forecast their future load variations, guide the industrial customers to adjust their production planning in advance, avoid the electricity peak and save the electricity costs. While, for the power system, STLF of industrial customers could promote the stable operation of the smart grid.

This paper tries to address the STLF problem for individual customers based on ensemble hidden Markov Model (e-HMM). In this paper, the important features are obtained via GRA to reduce redundant features and improve model efficiency, while the sliding time window method provides a dynamic description of the time process for multiple time series. Then, a novel log-likelihood similarity measurement strategy is proposed, which is proven to effectively improve the prediction performance of HMM. What's more, a reasonable hyperparameter selection strategy improves the efficiency of model training. Last, the e-HMM framework based on the Bagging ensemble learning is proposed, which improves the generalization ability and stability of a single HMM and further reduces the prediction errors. Experimental results show that the overall prediction accuracy of the proposed method is superior to several state-of-the-art models. In addition, we find that the proposed method is sensitive to the prediction time span. Although the method shows lower overall errors in different forecasting periods, as the prediction time increases, the prediction error is more likely to deviate from the normal range. In other words, the proposed method performs better in daily forecasts.

We expect the proposed method to be test in other types of customers, such as commercial customers and residents. As for future work, more real data sets will be tested to prove the reliability of this method, and the impact of the combination of different features on the model will be discussed.

REFERENCES

[1] K. Zor, O. Timur, and A. Teke, "A state-of-the-art review of artificial intelligence techniques for short-term electric load forecasting," in *Proc. 6th Int. Youth Conf. Energy (IYCE)*, Jun. 2017, pp. 1–7.

[2] A. Ghasemi, H. Shayeghi, M. Moradzadeh, and M. Nooshyar, "A novel hybrid algorithm for electricity price and load forecasting in smart grids with demand-side management," *Appl. Energy*, vol. 177, pp. 40–59, Sep. 2016.

[3] 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.

[4] *National Bureau of Statistics of the People's Republic of China, China Energy Statistical Yearbook 2018*, China Stat. Press, Beijing, China, 2019, pp. 124–125.

[5] Y. Yang, M. Wang, Y. Liu, and L. Zhang, "Peak-off-peak load shifting: Are public willing to accept the peak and off-peak time of use electricity price?" *J. Cleaner Prod.*, vol. 199, pp. 1066–1071, Oct. 2018.

[6] C.-M. Huang, C.-J. Huang, and M.-L. Wang, "A particle swarm optimization to identifying the ARMAX model for short-term load forecasting," *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 1126–1133, May 2005.

[7] J. C. López, M. J. Rider, and Q. Wu, "Parsimonious short-term load forecasting for optimal operation planning of electrical distribution systems," *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 1427–1437, Mar. 2019.

[8] A. Lahouar and J. Ben Hadj Slama, "Random forests model for one day ahead load forecasting," in *Proc. IREC 6th Int. Renew. Energy Congr. (IREC)*, Mar. 2015, pp. 1–6.

[9] E. Ceperic, V. Ceperic, and A. Baric, "A strategy for short-term load forecasting by support vector regression machines," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4356–4364, Nov. 2013.

[10] N. Ding, C. Benoit, G. Foggia, Y. Bésanger, and F. Wurtz, "Neural network-based model design for short-term load forecast in distribution systems," *IEEE Trans. Power Syst.*, vol. 31, no. 1, pp. 72–81, Jan. 2016.

[11] 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.

[12] K. Yan, W. Li, Z. Ji, M. Qi, and Y. Du, "A hybrid LSTM neural network for energy consumption forecasting of individual households," *IEEE Access*, vol. 7, pp. 157633–157642, 2019.

[13] K. Amarasinghe, D. L. Marino, and M. Manic, "Deep neural networks for energy load forecasting," in *Proc. IEEE 26th Int. Symp. Ind. Electron. (ISIE)*, Aug. 2017, pp. 1483–1488.

[14] R. Sevlian and R. Rajagopal, "A scaling law for short term load forecasting on varying levels of aggregation," *Int. J. Electr. Power Energy Syst.*, vol. 98, pp. 350–361, Jun. 2018.

[15] Q. Li, L. Zhang, and F. Xiang, "Short-term load forecasting: A case study in Chongqing factories," in *Proc. 6th Int. Conf. Inf. Sci. Control Eng. (ICISCE)*, Dec. 2019, pp. 892–897.

[16] A. Domingo Gundin, C. Garcia, A. Yannis Dimitriadis, E. Garcia, and G. Vega, "Short-term load forecasting for industrial customers using FASART and FASBACK neuro-fuzzy Systems," in *Proc. Power Syst. Comput. Conf. (PSCC)*, Jun. 2002, pp. 1–7.

[17] Q. Ge, C. Guo, H. Jiang, Z. Lu, G. Yao, J. Zhang, and Q. Hua, "Industrial power load forecasting method based on reinforcement learning and PSO-LSSVM," *IEEE Trans. Cybern.*, early access, May 6, 2020, doi: 10.1109/TCYB.2020.2983871.

[18] L. Wang, S.-X. Lv, and Y.-R. Zeng, "Effective sparse AdaBoost method with ESN and FOA for industrial electricity consumption forecasting in China," *Energy*, vol. 155, pp. 1013–1031, Jul. 2018.

[19] A. S. Khwaja, M. Naeem, A. Anpalagan, A. Venetsanopoulos, and B. Venkatesh, "Improved short-term load forecasting using bagged neural networks," *Electr. Power Syst. Res.*, vol. 125, pp. 109–115, Aug. 2015.

[20] E. M. de Oliveira and F. L. Cyrino Oliveira, "Forecasting mid-long term electric energy consumption through bagging ARIMA and exponential smoothing methods," *Energy*, vol. 144, pp. 776–788, Feb. 2018.

[21] X. Dong, L. Qian, and L. Huang, "A CNN based bagging learning approach to short-term load forecasting in smart grid," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Aug. 2017, pp. 1–6.

[22] S. Ali, S. S. Tirumala, and A. Sarrafzadeh, "Ensemble learning methods for decision making: Status and future prospects," in *Proc. Int. Conf. Mach. Learn. Cybern. (ICMLC)*, Jul. 2015, pp. 211–216.

[23] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, "Deep EHR: A survey of recent advances in deep learning techniques for electronic health record (EHR) analysis," *IEEE J. Biomed. Health Informat.*, vol. 22, no. 5, pp. 1589–1604, Sep. 2018.

[24] S. Ungureanu, V. Topa, and A. Cziker, "Industrial load forecasting using machine learning in the context of smart grid," in *Proc. 54th Int. Univ. Power Eng. Conf. (UPEC)*, Sep. 2019, pp. 1–6.

[25] R. Jiao, T. Zhang, Y. Jiang, and H. He, "Short-term non-residential load forecasting based on multiple sequences LSTM recurrent neural network," *IEEE Access*, vol. 6, pp. 59438–59448, 2018.

[26] 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.

[27] Z. Ghahramani, "An introduction to hidden Markov models and Bayesian networks," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 15, no. 1, pp. 9–42, Feb. 2001.

[28] W. Hu, G. Tian, Y. Kang, C. Yuan, and S. Maybank, "Dual sticky hierarchical Dirichlet process hidden Markov model and its application to natural language description of motions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 10, pp. 2355–2373, Oct. 2018.

[29] P. Somani, S. Talele, and S. Sawant, "Stock market prediction using hidden Markov model," in *Proc. IEEE 7th Joint Int. Inf. Technol. Artif. Intell. Conf. (ITAIC)*, Dec. 2014, pp. 89–92.

[30] Q. Huang, L. Shao, and N. Li, "Dynamic detection of transmission line outages using hidden Markov models," *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 2026–2033, May 2016.

[31] W. Kong, Z. Y. Dong, D. J. Hill, J. Ma, J. H. Zhao, and F. J. Luo, "A hierarchical hidden Markov model framework for home appliance modeling," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3079–3090, Jul. 2018.

[32] S. Lu, G. Lin, H. Liu, C. Ye, H. Que, and Y. Ding, "A weekly load data mining approach based on hidden Markov model," *IEEE Access*, vol. 7, pp. 34609–34619, 2019.

[33] D. Zhou, M. Balandat, and C. Tomlin, "A Bayesian perspective on residential demand response using smart meter data," in *Proc. 54th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2016, pp. 1212–1219.

[34] A. Bajracharya, M. R. A. Khan, S. Michael, and R. Tonkoski, "Forecasting data center load using hidden Markov model," in *Proc. North Amer. Power Symp. (NAPS)*, Sep. 2018, pp. 1–5.

[35] S. Henselmeyer and M. Grzegorzek, "Short-term load forecasting with discrete state hidden Markov models," *J. Intell. Fuzzy Syst.*, vol. 38, no. 2, pp. 2273–2284, Feb. 2020.

[36] Z. Zhou, "Bagging," in *Ensemble Methods: Foundations and Algorithms*, 1st ed. Boca Raton, FL, USA: CRC Press, 2012, pp. 47–57.

[37] X. Guangcheng, C. Wenli, L. Xingzhi, Z. Ke, Z. Bo, and S. Hongliang, "Research and application of verification error data processing of electricity meter based on Grubbs criterion," in *Proc. 4th Int. Conf. Smart Grid Electr. Autom. (ICSGEA)*, Aug. 2019, pp. 13–17.

[38] B. Song, X. Zhou, H. Shi, and Y. Tao, "Performance-indicator-oriented concurrent subspace process monitoring method," *IEEE Trans. Ind. Electron.*, vol. 66, no. 7, pp. 5535–5545, Jul. 2019.

[39] B. Song, H. Yan, H. Shi, and S. Tan, "Multisubspace elastic network for multimode quality-related process monitoring," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 5874–5883, Sep. 2020.

[40] A. P. Kale and S. Sonavane, "PF-FELM: A robust PCA feature selection for fuzzy extreme learning machine," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1303–1312, Dec. 2018.

[41] T.-K. Kim and R. Cipolla, "Canonical correlation analysis of video volume tensors for action categorization and detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 8, pp. 1415–1428, Aug. 2009.

[42] E. Siggiridou and D. Kugiumtzis, "Granger causality in multivariate time series using a time-ordered restricted vector autoregressive model," *IEEE Trans. Signal Process.*, vol. 64, no. 7, pp. 1759–1773, Apr. 2016.

[43] J. Xu, B. Tang, H. He, and H. Man, "Semisupervised feature selection based on relevance and redundancy criteria," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 9, pp. 1974–1984, Sep. 2017.

[44] J. M. Leiva-Murillo and A. Artes-Rodríguez, "Information-theoretic linear feature extraction based on kernel density estimators: A review," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1180–1189, Nov. 2012.

[45] M. Han, R. Zhang, T. Qiu, M. Xu, and W. Ren, "Multivariate chaotic time series prediction based on improved grey relational analysis," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 10, pp. 2144–2154, Oct. 2019.

[46] T. Zhao, S. Wang, J. Zuo, X. Duan, and X. Wang, "Performance evaluation of smart meters based on grey relational analysis," in *Proc. 10th Int. Conf. Intell. Hum.-Mach. Syst. Cybern. (IHMSC)*, Aug. 2018, pp. 312–315.

[47] S. Liu, L. Tao, N. Xie, and Y. Yang, "On the new model system and framework of grey system theory," in *Proc. IEEE Int. Conf. Grey Syst. Intell. Services (GSIS)*, Aug. 2015, pp. 1–11.

[48] J.-S. Chou and T.-K. Nguyen, "Forward forecast of stock price using sliding-window Metaheuristic-optimized machine-learning regression," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3132–3142, Jul. 2018.

[49] A. W. Robertson, S. Kirshner, and P. Smyth, "Downscaling of daily rainfall occurrence over northeast Brazil using a hidden Markov model," *J. Climate*, vol. 17, no. 22, pp. 4407–4424, Nov. 2004.

YUANYUAN WANG (Member, IEEE) received the B.S. and M.Sc. degrees in electrical engineering from the Changsha University of Science and Technology, Changsha, China, in 2004 and 2007, respectively, and the Ph.D. degree in electrical engineering from the College of Electrical Engineering, Guangxi University, Guangxi, China, in 2012.

She is currently an Associate Professor with the College of Electrical and Information Engineering, Changsha University of Science and Technology. Her research interests include machine learning, power system load forecasting, and power system protection and control.

YANG KONG is currently pursuing the M.S. degree in electrical engineering with the School of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha, China. His current research interests include machine learning and its applications in load forecasting.

XIAFEI TANG received the B.E. degree in electrical engineering and automation from Beihang University, in 2006, and the Ph.D. degree in electrical and electronic engineering from The University of Manchester, U.K., in 2012.

After that, she joined the Changsha University of Science and Technology, Changsha, China in 2013. She is working as a Lecturer with the School of Electrical and Information Engineering, Changsha University of Science and Technology. Her research interests include risk assessment of power systems, the stability analysis of power systems, nonlinear systems control, and disturbance rejection.

XIAOQIAO CHEN is currently pursuing the Ph.D. degree with the Department of Computing and Mathematical Science, California Institute of Technology, Pasadena, CA, USA. Her research interests include computer and applied mathematics, and power system load forecasting.

YAO XU (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the Changsha University of Science and Technology, China, in 2006 and 2009, respectively, and the Ph.D. degree in electrical engineering from The University of Tennessee, Knoxville, TN, USA, in 2014.

She is currently a Transmission Planning Engineer with Pacific Gas and Electric Company, San Francisco, CA, USA. Her research interests include power markets, renewable energy integration, utility application of power electronics, and power system control.

JUN CHEN is currently pursuing the M.S. degree in electrical engineering with the School of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha, China. His research interests include deep learning and its application in load forecasting.

SHANFENG SUN is currently pursuing the M.S. degree in electrical engineering with the School of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha, China. His research interests include deep learning and its application in load forecasting.

YONGSHENG GUO is currently pursuing the M.S. degree in electrical engineering with the School of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha, China. His research interests include machine learning and its applications in power systems and data mining.

YUHAO CHEN is currently pursuing the M.S. degree in electrical engineering with the School of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha, China. Her research interests include deep learning and its applications in power systems and data mining.