

Received June 13, 2019, accepted June 27, 2019, date of publication July 1, 2019, date of current version July 18, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2926137

# Multi-Scale Convolutional Neural Network With Time-Cognition for Multi-Step Short-Term Load Forecasting

ZHUOFU DENG<sup>1</sup>, BINBIN WANG<sup>1,2</sup>, YANLU XU<sup>3</sup>, TENGTEG XU<sup>1</sup>, CHENXU LIU<sup>1</sup>, AND ZHILIANG ZHU<sup>1</sup>

<sup>1</sup>Software College, Northeastern University, Shenyang 110169, China

<sup>2</sup>Deep NEU Technology Company Ltd., Shenyang 110169, China

<sup>3</sup>Economic Research Institute, State Grid Liaoning Electric Power Supply Company Ltd., Shenyang 110015, China

Corresponding author: Binbin Wang (neubob414@163.com)

This work was supported by the State Grid Corporation of China under Grant 2018YF-51 and Grant 2019YF-40.

**ABSTRACT** Electric load forecasting has always been a key component of power grids. Many countries have opened up electricity markets and facilitated the participation of multiple agents, which create a competitive environment and reduce costs to consumers. In the electricity market, multi-step short-term load forecasting becomes increasingly significant for electricity market bidding and spot price calculation, but the performances of traditional algorithms are not robust and unacceptable enough. In recent years, the rise of deep learning gives us the opportunity to improve the accuracy of multi-step forecasting further. In this paper, we propose a novel model multi-scale convolutional neural network with time-cognition (TCMS-CNN). At first, a deep convolutional neural network model based on multi-scale convolutions (MS-CNN) extracts different level features that are fused into our network. In addition, we design an innovative time coding strategy called the periodic coding strengthening the ability of the sequential model for time cognition effectively. At last, we integrate MS-CNN and periodic coding into the proposed TCMS-CNN model with an end-to-end training and inference process. With ablation experiments, the MS-CNN and periodic coding methods had better performances obviously than the most popular methods at present. Specifically, for 48-step point load forecasting, the TCMS-CNN had been improved by 34.73%, 14.22%, and 19.05% on MAPE than the state-of-the-art methods recursive multi-step LSTM (RM-LSTM), direct multi-step MS-CNN (DM-MS-CNN), and the direct multi-step GCNN (DM-GCNN), respectively. For 48-step probabilistic load forecasting, the TCMS-CNN had been improved by 3.54% and 6.77% on average pinball score than the DM-MS-CNN and the DM-GCNN. These results show a great promising potential applied in practice.

**INDEX TERMS** Short-term load forecasting, probabilistic load forecasting, multi-step, multi-scale convolution, time cognition, deep learning.

## I. INTRODUCTION

Load forecasting plays an essential role for energy management and distribution management in power grids. It is a necessary part in order to ensure the balance between generation and demand. Operators of power grid need high-accuracy power load forecasting to maintain the safety and stability of power supply. Accurate load forecasting becomes more challenging due to the continuous development of the power grids and the increasing complexity of grid

management [1], [2]. Many countries have opened up electricity markets and facilitated the participation of multiple agents, which creates a competitive environment and reduces costs to consumers. In the electricity market, load forecasting has become one of the most important tasks for electricity market entities. Electricity load forecasting categories can be simply summarized as follows: very short-term load forecasting (VSTLF), short-term load forecasting (STLF), medium-term load forecasting (MTLF), and long-term load forecasting (LTLF). The cut-off horizons for these four categories are one day, two weeks, and three years respectively [3]. STLF gives great significances to power system in

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

providing strategies, reliability analysis, interchange evaluation, security assessment, and spot price calculation [4], [5], which brings a higher accuracy requirement.

In the last decades, short-term load forecasting algorithms have been widely studied. The major forecasting models are divided into three categories: traditional, machine-learning, and artificial intelligent models. The traditional models are studied and used frequently due to their fast computing speed and robustness [6]. Some common ones include linear regression [7], [8] and auto regressive integrated moving average (ARIMA) [9], [10]. Others, such as exponential smoothing [11] and multiple linear regression [12], [13], have also attracted the interests of relevant researchers. In dealing with the linear forecasting problem, these models are effective as a result of studying the qualitative relationship between electric load and its influencing factors. However, the linear regression methods are not competent in nonlinear problems like relationship between weather and load. Also, there are some other confusing problems. For examples, ARIMA has difficulty in selecting the corresponding order in two processes auto regressive (AR) and moving average (MA) etc.

Machine-learning models mainly include support vector regression (SVR) [14], [15], regression trees [16], random forest (RF) [17], gradient boosting regression trees (GBRT) [18], Kalman filtering [19], [20], and gray forecasting [21]. Gray forecasting can just only deal with the data type with exponent increase trend [22]. SVR can effectively extract non-linear features and avoid over-fitting, so it shows good performance in STLF. However, SVR is not stable enough to outliers, and the setting of training parameters involves many skills and difficulties, which produces a terrible training process. RF and GBRT are derived from decision tree, which has better robustness to outliers, fewer parameter settings, and higher forecasting accuracy. However, the decision tree algorithm does not serve an acceptable performance in the case of large load growth.

Then, more intelligent and automated artificial neural network (ANN) [23] methods are developed to overcome the non-linear and complex relationship in electric time series, such as back propagation neural network (BPNN), extreme learning machine (ELM), and radial basis function (RBF). Recently, deep learning has become one of the preferred technology in many research fields. Deep learning borrows spirits from ANN and boost the power of ANN via deepening its layers and leveraging its structures. These learning methods have been widely implemented to solve natural language processing and speech/image recognition problems. Deep belief networks (DBN), made up from multiple layers of restricted Boltzmann machines, was used in the problem of 24 hour ahead electricity consumption forecasting, and applied to available historical real data describing the electricity consumption in the Republic of Macedonia [24]. The layer-by-layer unsupervised training procedure is followed by fine-tuning of the parameters by using a supervised back-propagation training method. There are two most widely adopted models: recurrent neural networks (RNN) and

convolutional neural networks (CNN). Long short term memory (LSTM) [25], an improved gate-based RNN, is suitable for processing and predicting important events with relatively long intervals and delays in time series [26]. LSTM cannot only resolve the gradient disappearance problem that exists in the RNN structure, but also enhances the ability of long-term memory achieving the state-of-the-art precision in STLF [27], [28]. Nevertheless, LSTM has some probabilities to lose important information in training process. At the same time, it does not hold the capability of the parallel computing that results in a time-consuming and computing resource wasting behavior. The basic architecture of CNN is designed for the image recognition, but through careful structural design, CNN-based networks can still achieve top-level accuracy in sequence processing [29]–[31]. Gated CNN (GCNN) and temporal CNN are reported being practiced in load forecasting [32].

However, most researches only pay attention to one-step load forecasting. On the contrary, multi-step load forecasting gives more contributions to practical applications, such as electricity market bidding and spot price calculation. Bonetto and Rossi [33] proposed an original forecasting technique based on non-linear autoregressive (NAR) neural networks. This architecture allows parallel and efficient training and is also lightweight at runtime. Yan *et al.* [34] proposed a multi-step forecasting strategy with a one-dimension convolutional neural network to predict the load quantity of a relatively longer span for electricity market bidding. Although their work improves the ability of parallel computing saving time and resources, the accuracy of this model is not satisfied enough. Cai *et al.* [35] presented a direct multi-step model based on GCNN for multi-step load forecasting. As a result, this model achieves the state-of-the-art performance. Besides, it gets rid of the shortcomings of the accumulated errors that exist over the recursive multi-step process since the forecasting is obtained directly from the prior neighbor data. Nonetheless, these researches ignore the time feature that is indispensable in multi-step load forecasting.

In order to solve problems mentioned above, this paper proposes the multi-scale convolutional neural network with time-cognition (TCMS-CNN) for multi-step forecasting of power aggregation load. With the multi-scale convolution, the ability of CNN have been improved by extracting complex and significant features of power load sequences. In addition, the temporal cognition of our deep model is strengthened by the periodic representation of a special time mark. At last, we propose a novel framework using deep learning, which combines sufficient and discriminative features to extract potential law in the dataset providing an excellent result. The main contributions of the paper can be summarized as follows.

- A one-dimensional multi-scale convolution is introduced, which extracts the intrinsic relationship of the load sequence from different locations. Besides, MS-CNN is proposed with multiple residual blocks to increase the depth and improve the ability of

feature representation. Through comparison experiments with the state-of-the-art methods, MS-CNN effectively improves the forecasting accuracy at first.

- A novel strategy of a sophisticated periodical coding is proposed that enable our deep model born with a better capability of time-cognition. Experiments demonstrated that our network achieve considerable accuracy compared with currently popular algorithms.
- The TCMS-CNN model unifies two innovations above, which combines the multi-scale and time-cognition features to form an end-to-end multi-step predictive deep learning model. Complete experiments, including point load forecasting and probabilistic load forecasting, were performed to demonstrate the effectiveness of the proposed method. By competing with the state-of-the-art models, we find that TCMS-CNN can serve more accurate results and show excellent stability in multi-step forecasting, giving strong generalization in electricity market bidding and spot price calculation.

To the best of our knowledge, this is the first paper that presents periodically coded deep neural network combining multi-scale convolution for multi-step load forecasting. At the same time, this is the first paper to carry out multi-step probabilistic load forecasting. These innovations increase the learning ability of our novel deep learning network to discriminative features and implement end-to-end training and predicting. The purpose of our work is to optimize a multi-step load forecasting model based on deep neural network to improve accuracy in and facilitate production practice.

This paper will be structured as follows: Section II defines the problem and describes the details of our proposed method. Section III reports the experimental results. Section IV further discusses some insights as well as problems of the proposed method. The conclusions are drawn in Section V.

## II. METHODOLOGY

### A. PROBLEM FORMULATION

This paper focuses on STLF, which is considered a kind of time series prediction problem. Historical load ( $x^L$ ) and holiday data ( $x^H$ ) are selected as input, since in practice, they are highly relevant to load forecasting and acquired easily for business. The problem is to construct mapping relationships between historical load sequences and future load sequences, i.e.,

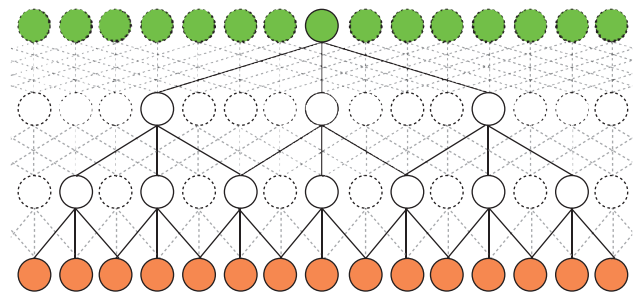
$$Y = f(X), \tag{1}$$

where  $X = [x_1, x_2, \dots, x_{N_i}]$  is the input sequence,  $Y = [y_{N_i+1}, y_{N_i+2}, \dots, y_{N_i+N_o}]$  represents the output prediction, and  $x_t = [x_t^L, x_t^H]$ .  $N_i$  tells the length of the input sequence and  $N_o$  the length of the output sequence. When  $N_o$  is equal to 1, it is single-step forecasting, and if  $N_o$  is greater than 1, it is a multi-step forecasting problem. In point load forecasting,  $y_t$  is a scalar. While in probabilistic load forecasting,  $y_t$  is a vector with length  $q$ , denoting the estimated  $q$  quantiles at step  $t$ . Based on the data set adopted in this paper, the forecast

horizon is set to 48 steps(hours), covering the hourly load forecast for the next 2 days.

### B. MS-CNN

Recently, CNN-based models have begun to receive attentions in sequence processing, and have achieved the state-of-the-art results in sequence tasks such as speech synthesis, language modeling, and machine translation. A simple convolutional neural network can only extract features in neighborhoods, which reflects limited relationships and cannot fit the target of load forecasting. Consequently, we introduce the multi-scale convolution to fuse extensive receptive fields information by stacking multiple layers of dilated convolution with various scales [36]. Dilated convolution cannot only reduce the size of learning parameters in deep network, but extract the multi-level and significant relationships of different position fused into our neural network.

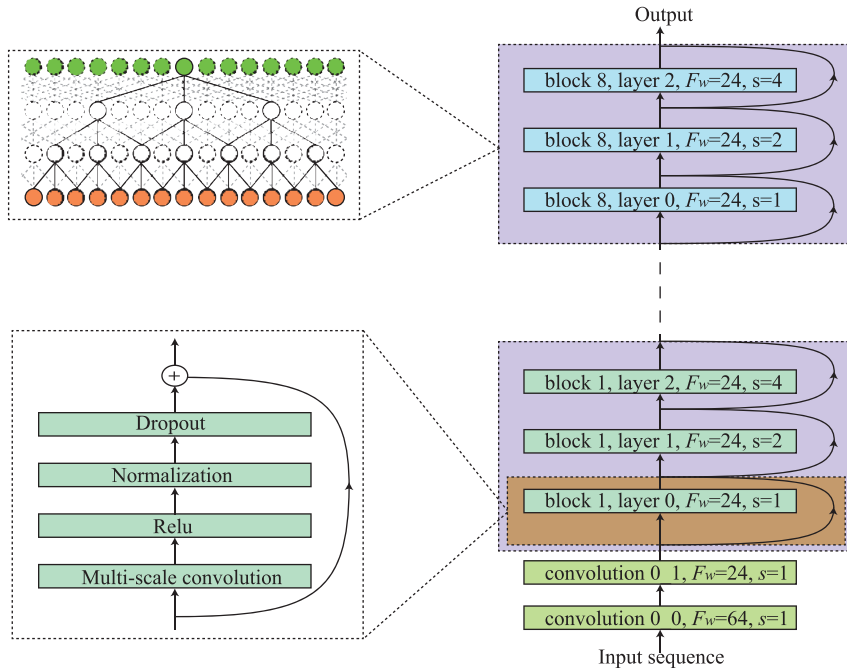


**FIGURE 1.** This figure includes three layers of multi-scale convolutions in a block, resulting in a receptive field size of 15. The dilation rate  $s$  from bottom to top is 1, 2, and 4, that is  $s = 2^l$  in the  $l$ -th layer.

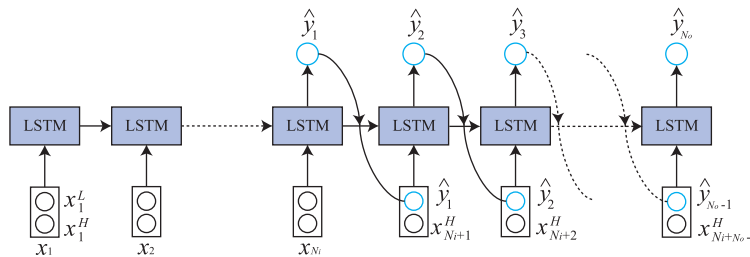
As shown in Fig. 1, we define a series of blocks, each of which contains a sequence of  $L$  convolutional layers. The activations in the  $l$ -th layer and  $j$ -th block are given by  $S^{(j,l)} \in \mathbb{R}^{F_w \times T}$ , where  $T$  is the length of input sequence that keeps equivalent in every layer. Also, the number of filters  $F_w$  preserve the consistence, which enables us to combine activations from different layers later. Meanwhile, each layer consists of a set of dilated convolutions with a specific rate parameter  $s$  and related non-linear operations. We denote the non-linear activation function, the normalization and the dropout operation together as  $g(\cdot)$ . In our work, we pay no attention to causal relationships in load forecasting due to it does not fit time series problems well. Convolutions are applied over three time steps,  $t, t-s$ , and  $t+s$ , so the complete equations is as follows. The filters are parameterized by  $W = \{W^{(1)}, W^{(2)}, W^{(3)}\}$  with  $W^{(i)} \in \mathbb{R}^{F_w \times F_w}$  and bias vector  $b \in \mathbb{R}^{F_w}$ . Let  $\hat{S}_t^{(j,l)}$  be the result of the dilated convolution at time  $t$  and  $S_t^{(j,l)}$  be the result after adding the residual connection such that

$$\hat{S}_t^{(j,l)} = g \left( W^{(1)} S_{t-s}^{(j,l-1)} + W^{(2)} S_t^{(j,l-1)} + W^{(3)} S_{t+s}^{(j,l-1)} + b \right), \tag{2}$$

$$S_t^{(j,l)} = S_t^{(j,l-1)} + V \hat{S}_t^{(j,l)} + e. \tag{3}$$



**FIGURE 2.** The network structure consists of multiple multi-scale convolutional blocks, each of which contains multiple residual layers with different scales. Each residual layer contains a dilation convolution, a ReLU activation function, a batch normalization operation, a dropout operation, and a residual connection.



**FIGURE 3.** The recursive multi-step method includes a variety of models such as ARIMA, SVR, and RNN. LSTM is illustrated in this instance. A single-step forecasting model are trained firstly, then we run prediction  $N_o$  times to get all  $N_o$  forecasts. In prediction phase, the result of the previous step forecasting is integrated into the next step input. For example, if we want to get  $\hat{y}_2$ , we must predict  $\hat{y}_1$  first and integrate  $\hat{y}_1$  into input of next step.

Let  $V \in \mathbb{R}^{F_w \times F_w}$  and  $e \in \mathbb{R}^{F_w}$  be a set of weights and biases for the residual, where parameters  $\{W, b, V, e\}$  are different for each layer. The dilation rate increases for consecutive layers within a block such that  $s_l = 2^l$ . This enables the structure to extract relationships from more scales rather than just neighborhood and increase the receptive field by a substantial amount without drastically increasing the number of parameters.

Fig. 2 describes our proposed multi-scale convolutional neural network (MS-CNN) architecture that holds 8 blocks, each of which have 3 dilated convolution layers with different dilation rate  $s$ . This mechanism combines multi-scale features and makes a deeper network that is demonstrated an effective model backbone for load forecasting in section 3.

### C. CONVOLUTIONAL NEURAL NETWORK WITH TIME-COGNITION

Currently, multi-step load forecasting methods can be divided into two categories: direct multi-step and recursive multi-step [35]. The recursive multi-step method at beginning forecasts the first step, and then integrates the predicted value into the input sequence for the next step forecasting, as illustrated in Fig. 3. The prediction error of every step is accumulated to influence next step severely. Conversely, the direct multi-step method feeds the input sequence into the model and obtains all steps to be forecasted.

However, the current direct multi-step method does not take into account the temporal relationships between multiple forecasted load points, which makes the neural network

Days of the week	Nature coding	One-hot coding	Periodic coding
Sunday	0	[1 0 0 0 0 0 0]	[ 0.00 1.00 ]
Monday	1	[0 1 0 0 0 0 0]	[ 0.78 0.62 ]
Tuesday	2	[0 0 1 0 0 0 0]	[ 0.98 -0.22 ]
Wednesday	3	[0 0 0 1 0 0 0]	[ 0.43 -0.90 ]
Thursday	4	[0 0 0 0 1 0 0]	[ -0.43 -0.90 ]
Friday	5	[0 0 0 0 0 1 0]	[ -0.98 -0.22 ]
Saturday	6	[0 0 0 0 0 0 1]	[ -0.78 0.62 ]

**FIGURE 4.** Three coding methods are illustrated for comparison by coding seven days of one week. Natural coding uses integers from 0 to 6 to represent days of a week, while the one-hot coding converts the natural code into seven one-hot vectors. The periodic coding uses the sin and cos functions to convert the natural coding into 7 vectors, each of which contains 2 values. Natural coding tells the time mark using semantic description lacking of period. The same problem also exists in one-hot coding. With our proposed periodic coding, the uniqueness and periodicity are preserved effectively that enables the neural network with a well-trained time-cognition.

model lose prior knowledge resulting in a precision to be improved. Traditionally, there are two ways to encode time: natural coding and one-hot coding [37], which ignore the cyclic characteristic in electricity behavior and are not suitable in load forecasting. For example, 0 Sunday and 6 Saturday seems too far but a neighborhood actually. Therefore, we propose a kind of periodic coding to mark each step of the input sequence and predicted ones. Periodic coding emphasizes the uniqueness of the moment in one period and provides periodicity description. In details, periodic coding depends on a unique markup through the sin and cos functions. For example, if the natural encoding to one day in a week is  $n^{dw}$ , then the periodic encoding converts  $n^{dw}$  to a vector  $[p^{dwsin}, p^{dwcoss}]$  by the following formula:

$$\begin{cases} p^{dwsin} = \sin \frac{2\pi n^{dw}}{T^{dw}} \\ p^{dwcoss} = \cos \frac{2\pi n^{dw}}{T^{dw}} \end{cases} \quad (4)$$

where  $T^{dw}$  describes the length of a period, 7 days in a week. In load forecasting with time-cognition, let  $n^{hd}$ ,  $n^{dw}$ , and  $n^{dy}$  denote the hour location in one day and day in one week or year with  $[p^{hd\sin}, p^{hd\cos}], [p^{dwsin}, p^{dwcoss}]$ , and  $[p^{dysin}, p^{dycoss}]$  respectively. The complete periodic coding for one point in the year is followed by (5) and Fig. 4 gives an example three coding approaches for comparison.

$$p_t = [p_t^{hd\sin}, p_t^{hd\cos}, p_t^{dwsin}, p_t^{dwcoss}, p_t^{dysin}, p_t^{dycoss}] \quad (5)$$

### D. TCMS-CNN

TCMS-CNN model is a hybrid network of MS-CNN and time-cognition models. Periodically encoded marks are concatenated with the load sequence fed into the MS-CNN.

Meanwhile, a fully connected network learns the periodic codes of the forecasted multi-step position to increase extra features for prediction. Both of strategies strengthen the hybrid network’s capability on time-cognition periodically. Then, the outputs of two subnetworks are fused and fed to another fully connected layer to learn and predict the final result. The entire framework of TCNS-CNN is described in detail in Fig. 5. TCMS-CNN reveals temporal characteristics of the input sequence with multi-scale convolutions to extract more potential features, which provides an advanced prediction for electricity market bidding. The implementation can be roughly divided into three stages: data preparation, model training, and forecasting, which are shown in Fig. 6.

## III. RESULTS

### A. DATASET

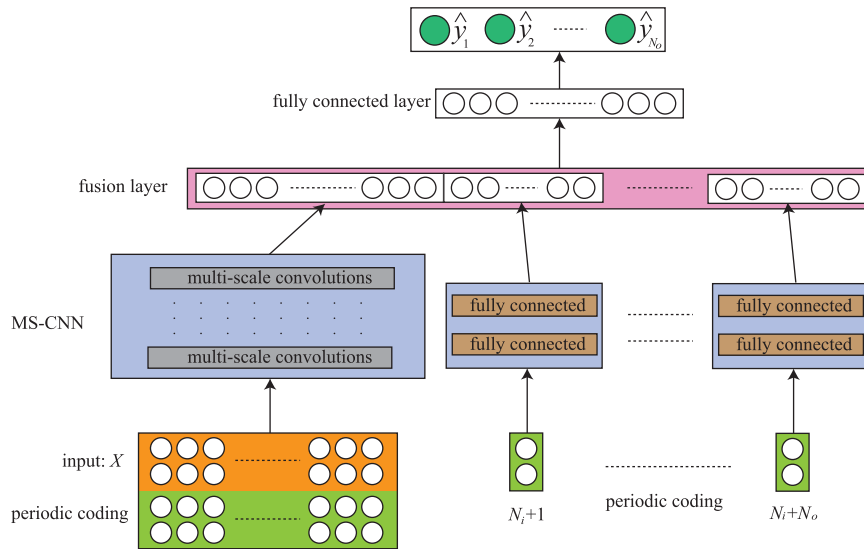
We use Ireland’s 2014–2018 load dataset and the granularity is hourly [38]. Load data are the total electricity consumption in this country. It is widely known that there is a strong nonlinear correlation between load and weather characteristics such as temperature and humidity. In the experiments, we compared our proposed MS-CNN and TCMS-CNN with a wide range of state-of-the-art models, some of which are linear models. For the sake of fairness and clear contrast, only load, holiday, and time data are used for all experiments. We intercepted a two-week hourly load profile, starting at zero o’clock on Tuesday, as shown from the Fig. 7, and the daily and weekly loads have significant periodicity, which also belongs to a common character of most power system loads.

Fig. 8 shows a box plot to collect the statistics of the electric load distribution in 24 hours of a day (0:00 AM - 24:00 PM) over our dataset. We observe that the load value of each hour has different maximums, minimums, medians, and quartiles, which proves a great fluctuation caused by dataset’s periodicity leads to difficulties of learning the rules of load forecasting. Our dataset contains 31,128 pieces that are divided into training set, validation set, and testing set by 60%, 20%, and 20%.

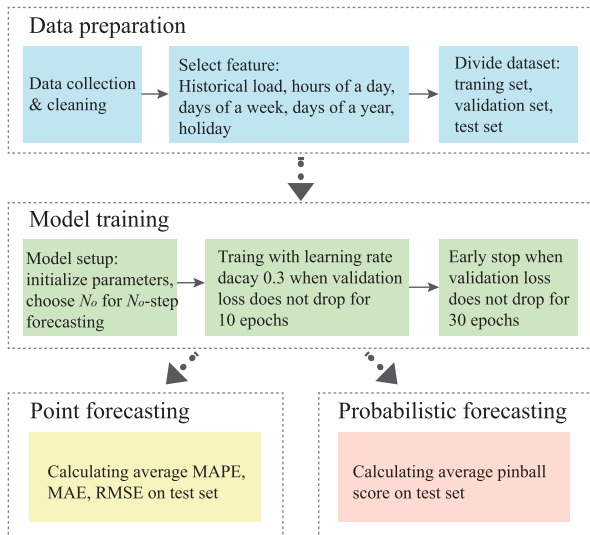
### B. EXPERIMENT SETUP

All experiments were conducted on a cloud server with two NVIDIA P4 computing cards and the CPU with 8 cores. The implementations of SARIMAX and SVR are based on the StatsModels and scikit-learn packages respectively. Other neural network-based models are realized by the Keras framework with Tensorflow [39] backend.

The experiments consist of three sections. At first, MS-CNN is compared with SARIMAX, DBN, SVR, LSTM, and residual neural networks (ResNet) at the single-step forecasting. Moreover, based on the multi-step MS-CNN, time coding methods including natural coding, one-hot coding, and our proposed periodic coding are compared. Finally, we evaluated our proposed TCMS-CNN to compare the performance with the state-of-the-art methods like recursive multi-step LSTM (RM-LSTM), direct multi-step



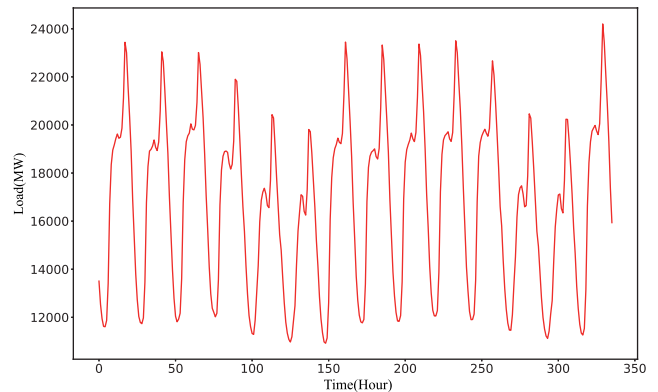
**FIGURE 5.** Two subnetworks MS-CNN and fully connected networks build up the baseline of TCMS-CNN for multi-scale and periodical coding features fusion. The subnetwork on the left is MS-CNN, and the input data sequences contain historical load, holiday, and periodical coding. Subnetworks on the right hold two full connection layers, inputs of which are periodic encoding of many predicted steps. The representation vectors output from the two sub-networks are concatenated as inputs of the top-level fully connected layer for generating loads at the predicted steps. This framework ensures the model obtain sufficient characteristics, which enhanced the understanding of dataset.



**FIGURE 6.** Flowchart for implementation of multi-step load forecasting.

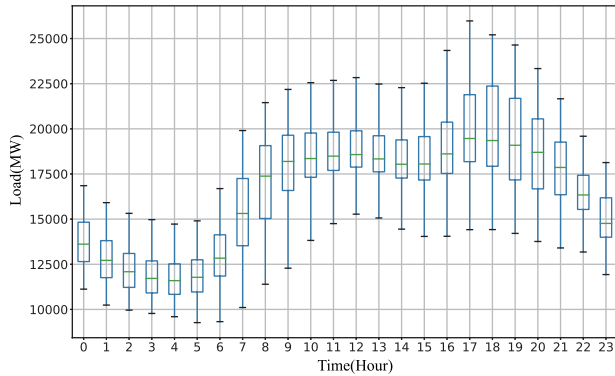
MS-CNN (DM-MS-CNN), and the direct Multi-step GCNN(DM-GCNN).

Global parameters settings are followed. Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors model (SARIMAX) is an extended model of ARIMA that adds periodicity and uses external information to enhance the predictive ability of the model. The standard SARIMAX model follows the notation of SARIMAX( $p, d, q$ )( $P, D, Q$ ) $S$ , where  $p$  = non-seasonal auto-regressive (AR) order,  $d$  = non-seasonal differencing,  $q$  = non-seasonal moving



**FIGURE 7.** This figure shows a typical 2-week load profile with significant periodicity. The valley of the week’s load is on Saturday and Sunday, whereas the peak appears on weekdays, and there is a similar changing trend every week. Similarly, the daily load trend also shows a similar situation.

average (MA) order,  $P$  = seasonal AR order,  $D$  = seasonal differencing,  $Q$  = seasonal MA order, and  $S$  = time span of the repeating seasonal pattern. SVR is an important application branch of Support Vector Machine (SVM) and has been widely used in regression problems in recent years. In SVR parameters, 'kernel' specifies the kernel type to be used in the algorithm. 'degree' denotes degree of the polynomial kernel function. 'gamma' is kernel coefficient and 'coef0' independent term in kernel function. 'tol' and 'C' represent the tolerance for stopping criterion and penalty parameter of the error term respectively. 'Shrinking' tells whether to use the shrinking heuristic. The parameters of SARIMAX and



**FIGURE 8.** The 24-hour power load box plot shows the distribution of load in our dataset. It demonstrates there are remarkable changes in quartiles of different times, which grows influenced by electricity behaviors of holidays.

**TABLE 1.** The explanations of global parameters setting in experiments of SARIMAX and SVR models.

SARIMAX		SVR	
$p$	3	kernel	'rbf'
$d$	0	degree	None
$q$	1	gamma	'auto'
$P$	1	coef0	0.0
$D$	1	tol	1e-3
$Q$	0	C	0.1
$S$	24	epsilon	0.0
		shrinking	True
		max_iter	None
		input_length	20

SVR are shown in Table 1. We use grid search over all possible combinations of parameter values within a predefined range of values to get the parameters of SARIMAX, and the  $S$  is 24 clearly. For SVR, most parameters are defaults except that  $C$  and  $input\_length$  are chosen by cross validation.

There are several infrastructure neural networks including DBN, LSTM, ResNet, MS-CNN, and GCNN, and they share the same parameters in different experiments. The parameters of the DBN consist of two parts: the parameters of the unsupervised training phase and the parameters of the supervised fine tuning phase. The parameters of DBN are shown in 2, and the others in Table 3.

**TABLE 2.** Two phase parameters of DBN.

Parameters	DBN Phase 1	DBN Phase 2
Depth	5	5
Hidden neural	20+40+60+50+40	20+40+60+50+40
Batch size	128	128
Input length	8	8
Training method	Contrastive Divergence	MSE+SGD
Start learning rate	0.0005	0.005
Learning rate decay	None	0.3
Training stop	200 epoch	early stopping

### C. EVALUATION METRICS

The point forecasting performances are evaluated using three performance metrics [40], including root mean square

**TABLE 3.** Shared parameters of neural network models and related strategies selected.

Parameters	LSTM	ResNet	MS-CNN	GCNN
Depth	3	34	34	6
Hidden neural	128	None	None	None
Kernel size	None	8	8	6,3,3 + 1,1,1
Kernel number	None	24	24	8,5,1 + 8,5,1
Batch size	128	128	128	128
Input length	240	240	240	240
Dropout rate	0.05	0.05	0.05	0.05
Loss function	MSE	MSE	MSE	MSE
Optimizing method	AMSGrad	AMSGrad	AMSGrad	AMSGrad
Start learning rate	0.01	0.01	0.01	0.01
Learning rate decay	0.3	0.3	0.3	0.3
Training stop	early stopping	early stopping	early stopping	early stopping

error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). Smaller values of their outputs mean higher forecasting accuracy. The formulations of the above three metrics are as follows:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2} \quad (6)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i| \quad (7)$$

$$MAPE = \frac{1}{N} \sum_{i=1}^N \frac{|\hat{y}_i - y_i|}{y_i} \times 100 \quad (8)$$

where  $y_i$  is the ground truth of testing sample,  $\hat{y}_i$  represents the prediction, and  $N$  is number of testing samples.

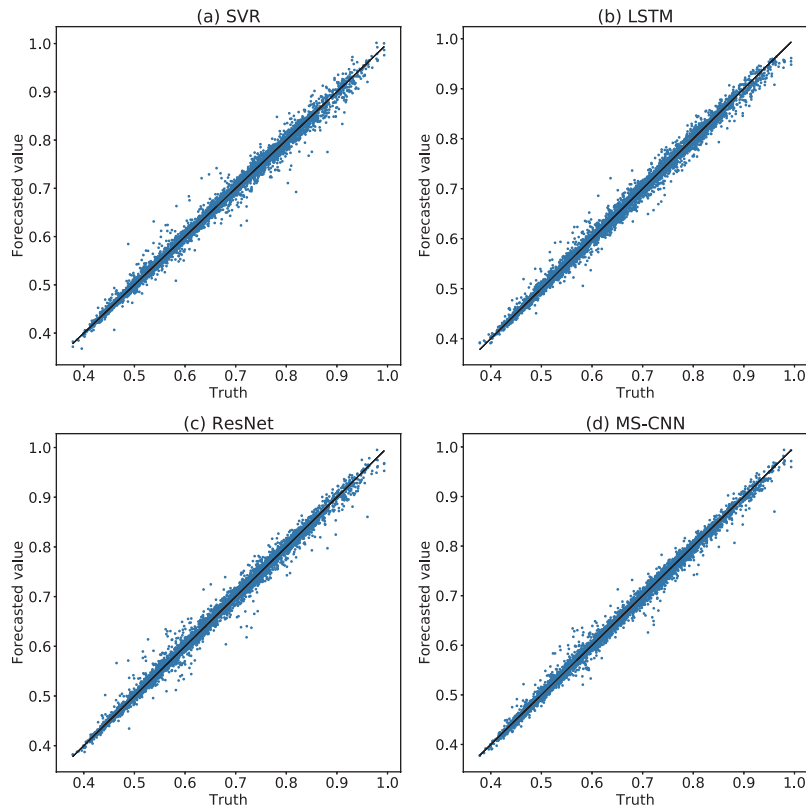
For probabilistic load forecasting evaluation, there are three commonly used attributes: reliability, sharpness, and resolution. Reliability refers to how close the predicted distribution is to the actual one. Sharpness refers to how tightly the predicted distribution covers the actual one. Resolution refers to how much the predicted interval varies over time. Measures like Kolmogorov-Smirnov statistic, Cramer-von Mises statistic, and Anderson-Darling statistic are only be used to assess the unconditional coverage of a probabilistic forecast, but not evaluate its sharpness or resolution. In this paper, the performance of the probabilistic forecasts are evaluated by the average of the total pinball score, which is a comprehensive measure considering not only reliability but also sharpness and resolution. The pinball score for one quantile is calculated as follows:

$$Pinball\_score = \begin{cases} (1 - q) (\hat{y}_t^q - y_t) & \hat{y}_t^q \geq y_t \\ q (y_t - \hat{y}_t^q) & \hat{y}_t^q < y_t \end{cases} \quad (9)$$

where  $q$  denotes the targeted quantile,  $\hat{y}_t^q$  denotes the forecast at the  $q$ th quantile at step  $t$ .

### D. PERFORMANCE COMPARISON WITH ABLATION STUDY AT SINGLE STEP

In order to clarify the predictive performance of our proposed MS-CNN model for prediction, we compared it with the state-of-the-art models such as SARIMAX, SVR, LSTM, and



**FIGURE 9.** Scatter plots of forecasting results in comparisons of our proposed MS-CNN, SVR, LSTM, and ResNet at single step point prediction.

**TABLE 4.** Comparison results of our proposed MS-CNN, SARIMAX, SVR, DBN, LSTM, and ResNet at single step point forecasting.

Model	RMSE(MW)	MAE(MW)	MAPE(%)
DBN	388.3	273.03	1.57
SARIMAX	270.29	181.65	1.20
SVR	285.53	180.42	1.04
LSTM	272.2	187.74	1.09
ResNet	263.83	179.11	1.05
MS-CNN	252.59	167.87	0.98

ResNet at single-step forecasting accuracy. In this ablation study, all input sequences are not marked with the time stamps. The results of the comparison are shown in Table 4 and Fig. 9.

Through the results, we can observe that models based on neural networks generally achieve higher accuracy than SARIMAX. SVR’s MAPE performs better, but its RMSE becomes inferior among all models. Due to its powerful learning and fitting ability of deeper network, vanilla ResNet outperforms LSTM in three metrics giving great potential for loading forecasting. In statistics, our proposed MS-CNN achieves the best accuracy, and compared with LSTM, MS-CNN increases 7.2%, 11.58%, and 10.09% in RMSE, MAE, and MAPE respectively. It demonstrates that the multi-scale convolution helps MS-CNN extract more significant features from the input sequence providing the excellent

ability of mining valuable relationships in the dataset, as illustrated in Fig. 9.

**E. EVALUATIONS ON TIME CODING METHODS**

In this subsection, we compare our proposed time coding strategy periodical coding with common methods on multi-step forecasting performance, including natural coding, one-hot coding, and periodic coding. All experiments are implemented using the MS-CNN for the sake of fairness. For three dimension informations, hours of a day, days of a week, and days of a year, as discussed in section 2, the natural coding has 3 (1 + 1 + 1) features, one-hot coding holds 397 (24 + 7 + 366) features, and our periodic coding serves 6 (2 + 2 + 2) features. The target of multi-step load forecasting adopts next 48 hours and evaluated results are listed in Table 5.

**TABLE 5.** Comparisons of different time coding approaches including natural coding, one hot, and our proposed periodical coding on RMSE, MAE, and MAPE.

Model	RMSE(MW)	MAE(MW)	MAPE(%)
Natural Coding	1.87e3	1.5e3	8.47
One-hot Coding	1.27e3	0.93e3	6.09
Periodic Coding	1.02e3	0.65e3	3.74

Table 5 expresses that the natural coding has the lowest accuracy, and the neural network cannot utilize time features to produce an acceptable prediction. Natural coding provides



only simple representation to time, which exhibits a significant periodical relationship. One-hot coding removes the linear temporal features in the natural coding, thus improved the accuracy of load forecasting. However, it still lacks of the periodic correlation between time and load, which is difficult for generalization in our problem. The periodic coding achieved the best accuracy, indicating that it effectively learned periodic features in the training addressing problems caused by other two algorithms. In experiments, RMSE, MAE, and MAPE of periodic coding decreased by 19.69%, 30.1%, and 38.59% compared to the one-hot coding respectively.

### F. COMPARISON OF THE STATE-OF-THE-ART MULTI-STEP FORECASTING MODELS

We evaluated several the state-of-the-art multi-step models, including RM-LSTM, DM-MS-CNN, and DM-GCNN, with our proposed TCMS-CNN for 48 hours forecasting on 5976 pieces of testing data listed in Table 6, where each item represents an average MAPE of 5976 experiments on each step. RM-LSTM is derived from the recursive multi-step model. DM-MS-CNN and DM-GCNN belong to the direct multi-step model without any time coding. MAPE was chosen as the only metric to reflect the algorithm's performance. The results are depicted in Table 6. RM-LSTM yielded the best results before the fourth step but fell behind others increasingly. RM-LSTM has the predominance in single-step, whereas accumulated errors results in poor performances with growing steps. DM-MS-CNN is built up with our proposed MS-CNN for direct multi-step forecasting. DM-GCNN is another direct multi-step model with gated CNN outperforming RNN for the sequential modeling task. Their MAPE growth are slower than RM-LSTM indicating an improved performance for multi-step load forecasting. Our proposed model TCMS-CNN achieved the best accuracy over steps on average, improved by 34.73%, 14.22%, and 19.05% on average MAPE than RM-LSTM, DM-MS-CNN, and DM-GCNN respectively.

Fig. 10 gives 10 examples of four models' predictions to different group of 48 hours, where our proposed TCMS-CNN results keep a stable and closest performance to ground truth in general. Fig. 11 describes the compared methods' MAPE curve. Obviously, the MAPE curve of TCMS-CNN grows slowly with lower and steady errors manifesting a remarkable robustness in 48 hours load forecasting. Its success was owed to two points: 1) a basic convolutional neural network backbone MS-CNN playing a critical role in extracting multi-level features and fusion; 2) an effective periodical coding to integrate sequential features into our model. In conclusion, TCMS-CNN is proved to have the advanced precision and stable performance holding a promising application in practice.

### G. MULTI-STEP PROBABILISTIC FORECASTING

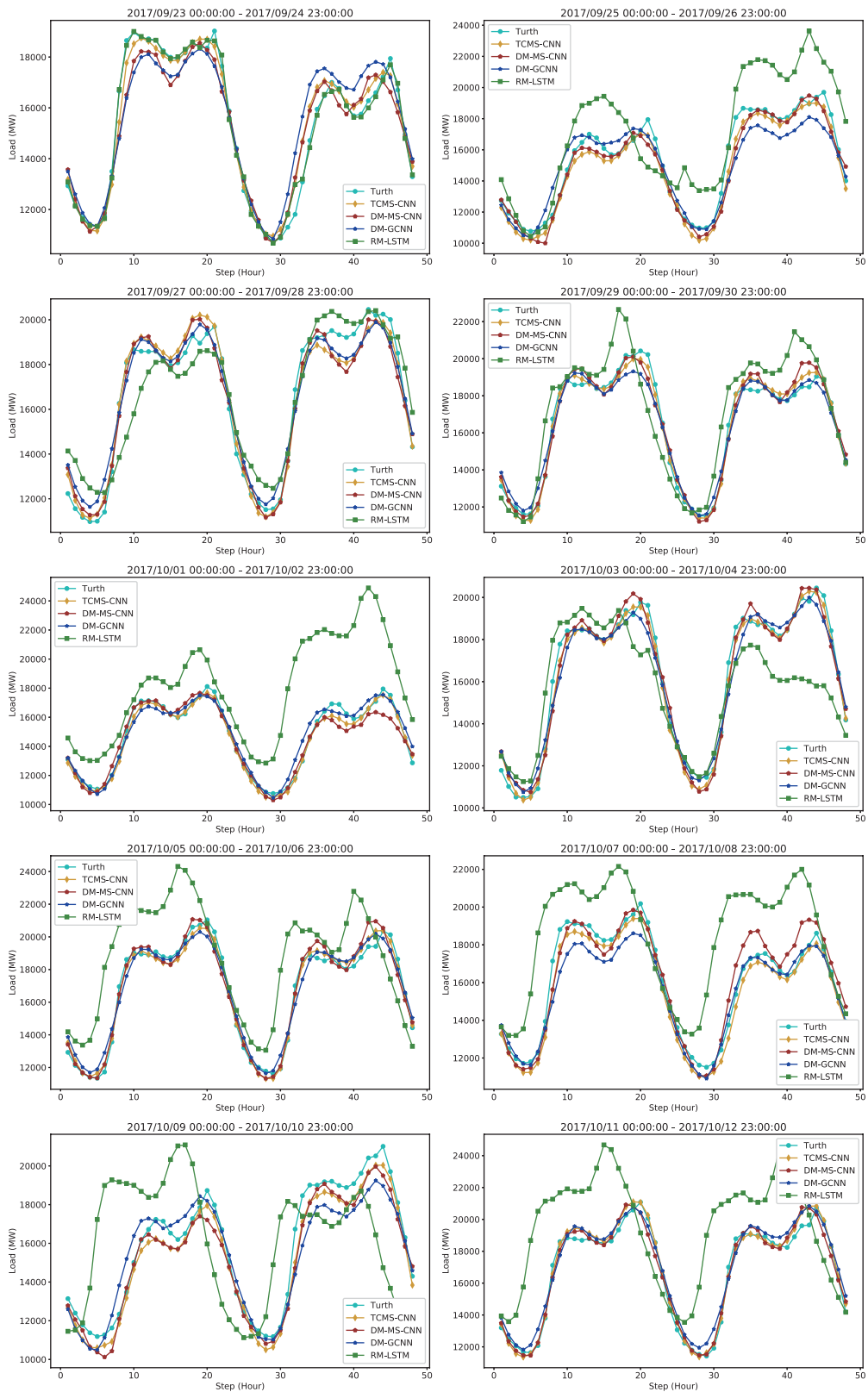
DM-MS-CNN, DM-GCNN, and TCMS-CNN are evaluated for multi-step probabilistic forecasting in this subsection.

**TABLE 6. Comparisons of multi-step forecasting models on average MAPE of 5976 pieces of testing data with 48 steps load point forecasting.**

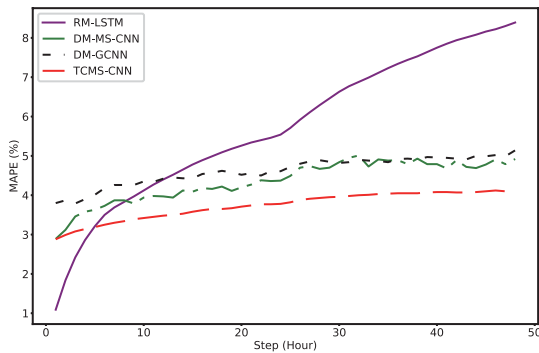
Step(Hour)	RM-LSTM	DM-MS-CNN	DM-GCNN	TCMS-CNN
1	1.09	2.89	3.80	2.88
2	1.84	3.12	3.87	2.99
3	2.42	3.46	3.79	3.08
4	2.86	3.58	3.90	3.14
5	3.21	3.63	4.01	3.19
6	3.50	3.73	4.18	3.25
7	3.69	3.87	4.26	3.30
8	3.83	3.87	4.26	3.34
9	3.97	3.79	4.26	3.39
10	4.12	3.94	4.35	3.42
11	4.27	3.98	4.34	3.45
12	4.40	3.97	4.42	3.48
13	4.52	3.94	4.45	3.50
14	4.65	4.12	4.43	3.53
15	4.78	4.09	4.43	3.58
16	4.89	4.18	4.54	3.62
17	4.99	4.16	4.57	3.65
18	5.09	4.22	4.62	3.65
19	5.18	4.11	4.58	3.67
20	5.26	4.20	4.52	3.71
21	5.34	4.27	4.56	3.74
22	5.40	4.38	4.50	3.77
23	5.46	4.36	4.61	3.77
24	5.54	4.37	4.61	3.78
25	5.71	4.49	4.72	3.82
26	5.92	4.70	4.80	3.88
27	6.11	4.74	4.86	3.91
28	6.29	4.67	4.89	3.93
29	6.46	4.70	4.85	3.95
30	6.63	4.84	4.82	3.96
31	6.77	4.95	4.84	3.98
32	6.88	5.01	4.90	4.00
33	6.99	4.73	4.88	4.01
34	7.11	4.91	4.87	4.03
35	7.23	4.88	4.84	4.04
36	7.34	4.89	4.92	4.05
37	7.44	4.79	4.93	4.05
38	7.53	4.93	4.91	4.05
39	7.64	4.79	4.97	4.07
40	7.75	4.79	4.96	4.08
41	7.85	4.68	4.95	4.08
42	7.94	4.89	4.93	4.07
43	8.01	4.72	4.91	4.07
44	8.08	4.69	5.00	4.08
45	8.16	4.78	4.99	4.10
46	8.22	4.91	5.02	4.12
47	8.30	4.79	4.99	4.10
48	8.39	4.92	5.14	4.09
Average	5.73	4.36	4.62	3.74

We adopt quantile regression with pinball loss for producing quantiles [41]. Recursive methods are not suitable for this experiment, because quantile outputs cannot be fed into input recursively. The application of AMSGrad requires the loss function to be differentiable so that the neural network can be trained using gradient descent. Common pinball loss is not differentiable everywhere, so we introduced the Huber norm to the loss function, with least change, making the loss function differentiable everywhere. The Huber norm can be viewed as a combination of the L1- and L2-norms:

$$H(y_t, \hat{y}_t^q) = \begin{cases} \frac{(\hat{y}_t^q - y_t)^2}{2\varepsilon} & 0 \leq |\hat{y}_t^q - y_t| \leq \varepsilon \\ |\hat{y}_t^q - y_t| - \frac{\varepsilon}{2} & |\hat{y}_t^q - y_t| > \varepsilon \end{cases} \quad (10)$$



**FIGURE 10.** Ten groups of 48-step hourly load forecasting results for different time, showing the performances of different models. RM-LSTM deviates from the truth drastically. Although DM-MS-CNN and DM-GCNN are close, in contrast, TCMS-CNN shows the best accuracy over the dataset.

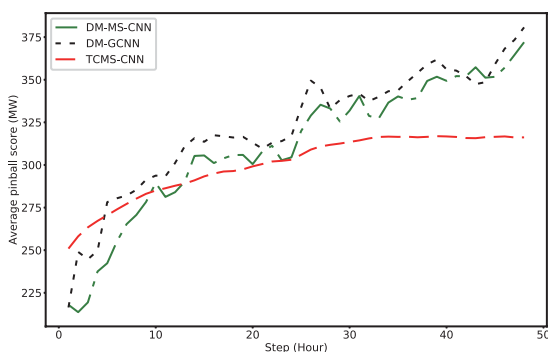


**FIGURE 11.** The MAPE curve of RM-LSTM, DM-MS-CNN, DM-GCNN, and TCMS-CNN. Obviously, our proposed TCMS-CNN serves a stable curve trend with lower errors.

where  $\varepsilon$  denotes the threshold magnitude for the L1- and L2-norms. In order to minimize the error caused by this approximation, we set  $\varepsilon$  to a very small value of 0.001. Then, we can get the approximated pinball loss:

$$L_{pinball} = \begin{cases} (1 - q) H(y_t, \hat{y}_t^q) & \hat{y}_t^q \geq y_t \\ qH(y_t, \hat{y}_t^q) & \hat{y}_t^q < y_t \end{cases} \quad (11)$$

As with the point prediction experiment, we still forecast the 48 steps probabilistic load for each sample on all 5976 test samples. Each predicted step produces 9 quantiles, from 0.1 to 0.9, so in the forecasting phase, a total of  $5976 \times 48 \times 9$  quantiles are generated. The average pinball score for each step and for all 48 steps are calculated in Table 7. As shown, the average pinball loss of TCMS-CNN on entire test set is 300.11MW, which is 3.54% and 6.77% lower compared to DM-MS-CNN and DM-GCNN respectively. The change of pinball score for each step is demonstrated in Fig. 12. Three average pinball scores curves all goes upward as the steps moving forward. Compared with DM-MS-CNN and DM-GCNN, the average pinball score curve of TCMS-CNN is less fluctuating and smoother, and the curve rises more slowly, which shows the superiority of the proposed method.



**FIGURE 12.** The average pinball score profile. The curve fluctuation of TCMS-CNN is smaller, indicating that the error difference between different steps is smaller.

Figure 13 captures a 48-step forecast curve, with the red dashed line corresponding to the predicted quantiles and the black solid line representing the actual value. Among the

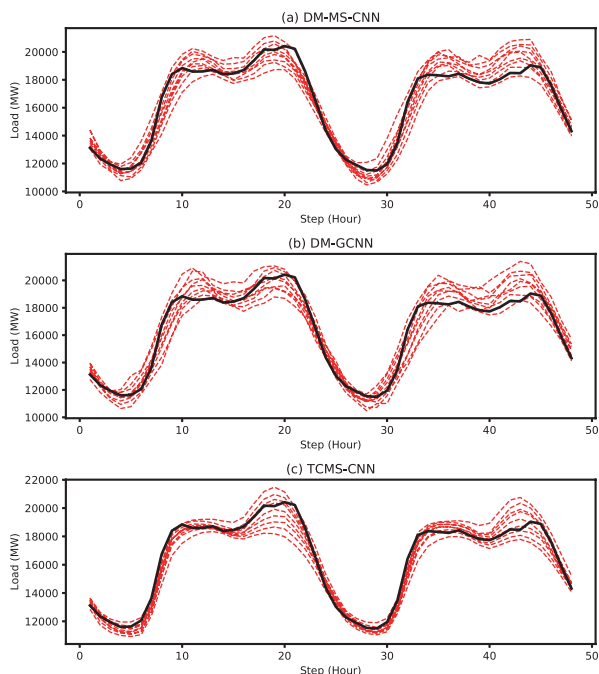
**TABLE 7.** Comparisons of three model on average pinball score of 5976 test samples with 48 steps probabilistic load forecasting.

Step(Hour)	DM-MS-CNN	DM-GCNN	TCMS-CNN
1	217.95	216.39	250.92
2	213.64	248.99	258.12
3	219.36	244.65	263.40
4	237.50	249.71	267.18
5	242.35	278.21	270.51
6	255.17	280.46	273.93
7	265.31	282.07	277.16
8	270.66	285.43	280.25
9	278.28	291.34	283.07
10	289.47	293.75	285.02
11	281.26	293.35	286.35
12	284.03	301.16	287.82
13	290.29	311.03	289.10
14	305.31	315.73	290.99
15	305.58	313.59	293.30
16	301.11	317.44	294.95
17	303.89	316.84	296.16
18	305.82	316.11	296.43
19	305.90	316.53	297.44
20	300.48	313.05	299.18
21	307.92	309.58	300.57
22	311.26	313.03	302.04
23	302.78	314.00	302.42
24	304.48	317.00	303.06
25	319.90	333.85	305.97
26	328.98	349.49	309.02
27	335.35	345.41	310.87
28	333.13	333.13	311.82
29	325.50	338.03	312.56
30	331.90	340.49	313.50
31	340.45	341.84	314.41
32	328.78	337.49	315.58
33	327.25	339.96	316.43
34	336.59	343.26	316.63
35	340.23	343.84	316.48
36	338.24	349.11	316.56
37	339.24	353.98	316.24
38	349.29	359.03	316.50
39	351.77	361.82	316.89
40	349.36	356.09	316.74
41	352.13	355.37	316.44
42	352.02	352.12	315.80
43	357.33	347.41	315.71
44	351.20	348.58	316.30
45	351.73	359.14	316.51
46	357.10	368.23	316.74
47	364.49	373.33	316.17
48	372.01	380.73	316.18
Average	311.12	321.90	300.11

three subgraphs, the red lines of TCMS-CNN are the smoothest and most compact, indicating that TCMS-CNN has achieved the best sharpness. Although the red lines of TCMS-CNN are very compact, the black line is still well wrapped in the red lines, showing that the reliability of TCMS-CNN keeps high.

**IV. DISCUSSION**

Multi-step short-term load forecasting is gaining more and more attentions in electricity market bidding and spot price calculation. The development of deep learning technology provides a pathway to improve accuracy for short-term load forecasting and a growing ability to fit the time series data.



**FIGURE 13.** A 48-step probabilistic forecasting profile, with the red dashed line corresponding to the predicted quantile and the black solid line representing the actual value.

In this paper, we give some contributions to multi-step load forecasting with deep learning.

At first, a multi-scale convolutional network (MS-CNN) is proposed to extract multi-level features fused to our model with dilated kernels. In addition, the shortcomings of traditional time coding approaches are analyzed. For strengthening the periodical description to sequential model, we design the periodical coding strategy to encode the load data for improved prediction. At last, we present the TCMS-CNN model to integrate multi-scale convolutions and periodical coding into an end to end trainable neural network, which optimizes the structure of CNN and extracts more relationships with periodical characters raising the accuracy of multi-step load forecasting. Some ablation experiments of comparison with the state-of-the-art methods were finished. For verifying the MS-CNN, we compared it with some popular models SARIMAX, SVR, LSTM, and ResNet on single-step prediction. The results demonstrated that MS-CNN has an obvious predominance 7.2%, 11.58%, and 10.09% improvement in RMSE, MAE, and MAPE to 2nd rank algorithm respectively which can be preferred as the baseline for advanced networks for load forecasting. In the second experiment, we evaluated our proposed periodical coding in comparison with natural coding and one-hot coding. Through the same neural networks, the statistics of RMSE, MAE, and MAPE gave a proof that the periodical coding outperforms remarkably than others providing a great potential in the sequential model prediction. Finally, the most advanced currently models RM-LSTM, DM-MS-CNN, and DM-GCNN had joined in our testings. 34.73%, 14.22%,

and 19.05% improvements on MAPE of point forecasting than the state-of-the-art methods RM-LSTM, DM-MS-CNN, and DM-GCNN respectively had been found on 5976 pieces of testing data for 48 steps load point forecasting. For 48 steps probabilistic load forecasting, TCMS-CNN had been improved by 3.54% and 6.77% on average pinball score than DM-MS-CNN and DM-GCNN respectively.

However, for direct multi-step forecasting, the performance of TCMS-CNN is not satisfied enough at first steps. Meanwhile, its network structure runs some complicated that results in a little time-consuming training and inference process. Future work includes improving accuracy for first steps and optimizing the network structure.

### V. CONCLUSIONS

This paper proposed TCMS-CNN for short-term multi-step load forecasting. With the multi-scale convolution, the ability of CNN have been improved by extracting complex and significant features of load sequences. In addition, the temporal cognition of our proposed model is strengthened by the periodic coding. At last, we propose a novel framework that combines sufficient and discriminative features to extract potential law in the dataset providing an excellent result. By competing with the state-of-the-art models, we find that TCMS-CNN can serve more accurate results and show excellent stability in multi-step point and probabilistic forecasting, giving strong generalization in electricity market bidding and spot price calculation.

### NOMENCLATURE

$\hat{y}_i$	forecasts
$N_i$	length of input sequence
$N_o$	forecast horizon and length of output
$y_i$	truth
ARIMA	auto regressive integrated moving average
CNN	convolutional neural networks
DBN	deep belief networks
DM-GCNN	direct Multi-step GCNN
DM-MS-CNN	direct multi-step MS-CNN
GCNN	Gated CNN
LSTM	Long short term memory
MAE	mean absolute error
MAPE	mean absolute percentage error
MS-CNN	multi-scale CNN
ResNet	residual neural network
RM-LSTM	recursive multi-step LSTM
RMSE	root mean square error
SARIMAX	Seasonal ARIMA with eXogenous
STLF	short-term load forecasting
SVR	support vector regression
TCMS-CNN	MS-CNN with time-cognition

### ACKNOWLEDGMENT

The authors would like to thank Minghao Xie, Heng Guo, Minghao Wang, Suhan Cui and Chengwei Cai for their assistances on experiments.

## REFERENCES

- [1] M. Espinoza, J. A. K. Suykens, R. Belmans, and B. D. Moor, "Electric load forecasting," *IEEE Control Syst.*, vol. 27, no. 5, pp. 43–57, Oct. 2007. [Online]. Available: <https://ieeexplore.ieee.org/document/4303474/>
- [2] T. Hong, "Energy forecasting: Past, present, and future," *Foresight Int. J. Appl. Forecasting*, vol. 32, pp. 43–48, Apr. 2014.
- [3] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," *Int. J. Forecasting*, vol. 32, no. 3, pp. 914–938, 2016.
- [4] Ö. F. Ertugrul, "Forecasting electricity load by a novel recurrent extreme learning machines approach," *Int. J. Electr. Power Energy Syst.*, vol. 78, pp. 429–435, Jun. 2016. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0142061515005578>
- [5] Z. Zhang, G. Liang, Y.-J. Dai, X.-U. Dong, and P.-X. Wang, "A short-term user load forecasting with missing data," DEStech Publications, Inc., Lancaster, PA, USA, Tech. Rep. 395-400, 2018.
- [6] K.-B. Song, Y.-S. Baek, D. H. Hong, and G. Jang, "Short-term load forecasting for the holidays using fuzzy linear regression method," *IEEE Trans. Power Syst.*, vol. 20, no. 1, pp. 96–101, Feb. 2005.
- [7] S. M. Moghaddas-Tafreshi, "A linear regression-based study for temperature sensitivity analysis of Iran electrical load," in *Proc. IEEE Int. Conf. Ind. Technol.*, Apr. 2008, pp. 1–7.
- [8] T. Hong, M. Gui, M. E. Baran, and H. L. Willis, "Modeling and forecasting hourly electric load by multiple linear regression with interactions," in *Proc. IEEE PES General Meeting*, Jul. 2010, pp. 1–8.
- [9] G. Juberias, R. Yunta, J. G. Moreno, and C. Mendivil, "A new ARIMA model for hourly load forecasting," in *Proc. IEEE Transmiss. Distrib. Conf.*, vol. 1, Apr. 1999, pp. 314–319.
- [10] L. Wei and Z. Zhen-Gang, "Based on time sequence of ARIMA model in the application of short-term electricity load forecasting," in *Proc. Int. Conf. Res. Challenges Comput. Sci.*, Dec. 2009, pp. 11–14. [Online]. Available: <http://ieeexplore.ieee.org/document/5401272/>
- [11] W. R. Christiaan, "Short-term load forecasting using general exponential smoothing," *IEEE Trans. Power App. Syst.*, vol. PAS-90, no. 2, pp. 900–911, Mar. 1971.
- [12] A. Tascikaraoglu and B. M. Sanandaji, "Short-term residential electric load forecasting: A compressive spatio-temporal approach," *Energy Buildings*, vol. 111, pp. 380–392, Jan. 2016.
- [13] J. Che and J. Wang, "Short-term load forecasting using a kernel-based support vector regression combination model," *Appl. Energy*, vol. 132, pp. 602–609, Nov. 2014. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0306261914007478>
- [14] W.-C. Hong, "Electric load forecasting by seasonal recurrent SVR (support vector regression) with chaotic artificial bee colony algorithm," *Energy*, vol. 36, no. 9, pp. 5568–5578, Sep. 2011. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0360544211004634>
- [15] W. Li, X. Yang, H. Li, and L. Su, "Hybrid forecasting approach based on GRNN neural network and SVR machine for electricity demand forecasting," *Energies*, vol. 10, no. 1, p. 44, Jan. 2017. [Online]. Available: <http://www.mdpi.com/1996-1073/10/1/44>
- [16] H. Mori and N. Kosemura, "Optimal regression tree based rule discovery for short-term load forecasting," in *Proc. IEEE Power Eng. Soc. Winter Meeting. Conf. Proc.*, vol. 2, Feb. 2001, pp. 421–426. [Online]. Available: <http://ieeexplore.ieee.org/document/916878/>
- [17] S. Li, P. Wang, and L. Goel, "Short-term load forecasting by wavelet transform and evolutionary extreme learning machine," *Electr. Power Syst. Res.*, vol. 36, no. 9, pp. 96–103, May 2015. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0378779615000036>
- [18] S. B. Taieb and R. J. Hyndman, "A gradient boosting approach to the Kaggle load forecasting competition," *Int. J. Forecasting*, vol. 30, no. 2, pp. 382–394, Jun. 2014.
- [19] H. M. Al-Hamadi and S. A. Soliman, "Short-term electric load forecasting based on Kalman filtering algorithm with moving window weather and load model," *Electr. Power Syst. Res.*, vol. 68, no. 1, pp. 47–59, Jan. 2004.
- [20] H. M. Al-Hamadi and S. A. Soliman, "Fuzzy short-term electric load forecasting using Kalman filter," *IEE Proc. Gener. Transmiss. Distrib.*, vol. 153, no. 2, pp. 217–227, Mar. 2006. [Online]. Available: [https://digital-library.theiet.org/content/journals/10.1049/ip-gtd\\_20050088](https://digital-library.theiet.org/content/journals/10.1049/ip-gtd_20050088)
- [21] Y. Lu and X. Lin, "Short-term load forecasting based on grey model and correction system," in *Proc. IEEE/PES Transmiss. Distrib. Conf. Expo. Asia Pacific*, Aug. 2005, pp. 1–5.
- [22] X. Zhang, J. Wang, and K. Zhang, "Short-term electric load forecasting based on singular spectrum analysis and support vector machine optimized by Cuckoo search algorithm," *Electr. Power Syst. Res.*, vol. 146, pp. 270–285, May 2017.
- [23] Z. Xiao, S.-J. Ye, B. Zhong, and C.-X. Sun, "BP neural network with rough set for short term load forecasting," *Expert Syst. Appl.*, vol. 36, no. 1, pp. 273–279, Jan. 2009. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S095741740700423X>
- [24] 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, Nov. 2016.
- [25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [26] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," 2015, *arXiv:1506.00019*. [Online]. Available: <https://arxiv.org/abs/1506.00019>
- [27] 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. 2017.
- [28] 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.
- [29] A. V. D. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," 2016, *arXiv:1609.03499*. [Online]. Available: <https://arxiv.org/abs/1609.03499>
- [30] N. Kalchbrenner, L. Espeholt, K. Simonyan, A. V. D. Oord, A. Graves, and K. Kavukcuoglu, "Neural machine translation in linear time," 2016, *arXiv:1610.10099*. [Online]. Available: <https://arxiv.org/abs/1610.10099>
- [31] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. 34th Int. Conf. Mach. Learn.-Vol.*, 2017, pp. 933–941.
- [32] A. Almalaq and G. Edwards, "A review of deep learning methods applied on load forecasting," in *Proc. 16th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2017, pp. 511–516.
- [33] R. Bonetto and M. Rossi, "Parallel multi-step ahead power demand forecasting through NAR neural networks," in *Proc. IEEE Int. Conf. Smart Grid Commun.*, Nov. 2016, pp. 314–319. [Online]. Available: <http://ieeexplore.ieee.org/document/7778780/>
- [34] K. Yan, X. Wang, Y. Du, N. Jin, H. Huang, and H. Zhou, "Multi-step short-term power consumption forecasting with a hybrid deep learning strategy," *Energies*, vol. 11, no. 11, p. 3089, Nov. 2018.
- [35] M. Cai, M. Pipattanasomporn, and S. Rahman, "Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques," *Appl. Energy*, vol. 236, pp. 1078–1088, Feb. 2019. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0306261918318609>
- [36] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 156–165.
- [37] Z. Yu, Z. Niu, W. Tang, and Q. Wu, "Deep learning for daily peak load forecasting—A novel gated recurrent neural network combining dynamic time warping," *IEEE Access*, vol. 7, pp. 17184–17194, 2019. [Online]. Available: <https://ieeexplore.ieee.org/document/8629072/>
- [38] *All Ireland Load Dataset' EirGridGroup*, Accessed: Oct. 4, 2018. [Online]. Available: <http://www.eirgridgroup.com/>
- [39] M. Abadi, P. Barham, J. Chen, S. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, and M. Isard, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Operating Syst. Design Implement. (OSDI)*, 2016, pp. 265–283.
- [40] T. J. Brailsford and R. W. Faff, "An evaluation of volatility forecasting techniques," *J. Banking Finance*, vol. 20, no. 3, pp. 419–438, Apr. 1996. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0378426695000151>
- [41] Y. Wang, D. Gan, M. Sun, N. Zhang, Z. Lu, and C. Kang, "Probabilistic individual load forecasting using pinball loss guided LSTM," *Appl. Energy*, vol. 235, pp. 10–20, Feb. 2019.



**ZHUOFU DENG** received the B.E. and M.E. degrees in software engineering from Northeastern University, China, in 2007 and 2009, respectively, where he is currently a Lecturer with the College of Software. His main research interests include the areas of computer vision, artificial intelligence, and medical image analysis.



**BINBIN WANG** received the B.E. degree from the College of Software, Northeastern University, China. He is currently an Algorithm Expert with Deep NEU Technology Company Ltd., and an Off-Campus Teacher with Northeastern University. His research interests include electric load forecasting, predictive maintenance, and artificial intelligence.



**CHENXU LIU** is currently pursuing the degree in software engineering with the College of Software, Northeastern University, China. His main research interests include data mining, computer vision, and complexity software systems.



**YANLU XU** received the B.E. degree in electronic information science and technology from Liaoning University, Shenyang, China, in 2008, and the M.E. degree from Northeastern University, Shenyang, in 2011. He is currently with the Economic Research Institute, State Grid Liaoning Electric Power Supply Company Ltd., Shenyang. His research interests include electricity energy marketing, data mining, and artificial intelligence in the Internet of Things.



**ZHILIANG ZHU** received the M.S. degree in computer applications and the Ph.D. degree in computer science from Northeastern University, China. He is a Fellow of the China Institute of Communications, and a member of the Software Engineering Education Instruction Committee of Education Ministry Expert of SOA Standards Working Group of Ministry of Industry and Information Technology. His main research interests include information integrate, complexity software systems, network coding and communication security, chaos-based digital communications, applications of complex-network theories, and cryptography. He has authored and coauthored over 130 international journal papers and 100 conference papers. He has also published five books, including *Introduction to Communication and Program Designing of Visual Basic .NET*. He is a Senior Member of the Chinese Institute of Electronics and the Teaching Guiding Committee for Software Engineering under the Ministry of Education. He was a recipient of nine academic awards at the national, ministerial, and provincial levels. He has served in different capacities for many international journals and conferences. He serves as the Co-Chair for the 1st–8th International Workshop on Chaos-Fractals Theories and Applications.



**TENGTENG XU** received the B.S. degree from the City Institute, Dalian University of Technology, in 2017. He is currently pursuing the master's degree with Northeastern University. His research interest includes artificial intelligence in power load forecasting.

...