Received: 2 October 2019 | Revised: 19 January 2020 | Accepted: 20 January 2020

DOI: 10.1002/ett.3890

SPECIAL ISSUE ARTICLE

WILEY

Multiscale deep network based multistep prediction of high-dimensional time series from power transmission systems

Hanlin Zhu[1,2] | Yongxin Zhu[1] | Hui Wang[1] | Shihui Wang[1,2] | Ziwei Liu[3] | Balusamy Balamurugan[4] | Pandi Vijayakumar[5] | Ming Xia[1,2]

[1]Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai, China

[2]University of Chinese Academy of Sciences, Beijing, China

[3]UMJI-Joint Institute, Shanghai Jiao Tong University, Shanghai, China

[4]School of Computer Science and Engineering, Galgotias University, Greater Noida, India

[5]Department of Computer Science and Engineering, University College of Engineering Tindivanam, Anna University Chennai, Melpakkam, India

Correspondence
Yongxin Zhu, Shanghai Advanced Research Institute, Chinese Academy of Sciences, 99 Haike Road, Pudong New District, Shanghai, China.
Email:zhuyongxin@sari.ac.cn

Funding information
NSFC Youth Fund, 61704179; Strategic Priority Research Program of Chinese Academy of Sciences, XDC02070700, XDC02070800

Abstract

Internet of energy makes the future power and energy network a more complicated and intelligent system. With the development of energy industry, the sample data of such system is high dimensional, dynamic, correlative, and complex. In order to meet people's needs and reduce the power redundancy, predicting the future energy demand and production is an essential approach. It is necessary for us to predict the later hours' or days' data, which means multistep prediction. However, the common one-step prediction model cannot forecast the power demand or production to make adequate preparation and the data have thousands of dimensions, which makes the problem challenging. In addition, the changeable pattern makes the common prediction algorithm do not perform good enough. In this article, we propose a sequence to sequence model to make multistep prediction with a baseline mean squared error (MSE) of 1.49×10^{-5}. In addition, we improve the model to be a multiscale deep network and decrease the MSE to 1.23×10^{-5} through adding extra information to match different patterns. Furthermore, the multitask learning trick makes the MSE decrease to 1.18×10^{-5}.

KEYWORDS

high-dimensional time series, LSTM, power system, seq2seq

1 | INTRODUCTION

Internet of energy's (IOEs) emergence provides a considerable number of promising research directions in the field of power system applications, including power monitoring, energy demand management, distributed storage coordination, and integration of traditional generators and renewable energy generators. With the rapidly growing demands for electricity, the state of power consumption correspondingly becomes more complex, the dynamics and complexity in power distribution system increase simultaneously. At the same time, new energy generator such as wind, nuclear, and solar energy also hove into the public's view and have accounted for an important part of the overall power system, which unavoidably adds to the complexity of power system. Under this circumstance, network congestion, low transmission efficiency, insufficient power transmission, and redundant transmission become striking problems that should not be underestimated, thus increasing the burden on power system. Therefore, lightening the unwanted burden, power system monitoring, demand managements, fault diagnosis, and automated management become a necessity. Meanwhile, the development of IOE and the popularity of artificial-intelligence-related technologies proffer a reliable source data and an effective data analysis method for smart grids. By processing and analyzing collected data, experts can improve and perfect the grid operation to achieve better performance, to reduce failures, to meet the needs of more people, and to reduce the power generation redundancy.

With the gradual completeness of the electricity and power system network, huge amounts of historical data can be collected and preserved, which, after sufficient analysis, can provide an instructing way on how to better operate the whole system. During the process of analyzing and synthesizing the historical and experiential information, new ideas of further improvements can be put forward to ensure efficient operation of complex power system.

In terms of the whole system's operation, the correctness of importing and exporting data is the foundation for further prediction. Insufficient power supply and redundant power supply can also be efficiently prevented by predicting power load. Relative people can correspondingly make full preparation for power provision according to multiple-step prediction on future energy consumption. By analyzing and monitoring the transmission data, demands for future electricity supply can be predictably met, power generation can be dynamically adjusted, power resources can be rationally allocated, foreseeable faults can be widely reduced, and resource imbalance can be beforehand avoided.

The data analysis is difficult and inefficient in intelligent energy networks.[1] In the changeable nature, it is difficult to perform complex computation to their changeable features.[2] Usually in complex power systems, data are mostly transmitted in the form of time series, which features for their high dynamics, high latitude, and its difficulty to express patterns. Taking the data of European energy system[3] as an example, the transmission system contains 1494 buses and 2146 lines, which means that in transmission part, the sampled grid data reaches thousands of dimensions. At the same time, the data of transmission lines presents small time, daytime, weekly correlation, and some other complex patterns that are difficult to apply to conventional mathematical models.

Syranidis et al[4] confirmed the necessity of prediction of European power system. However, there is no other work doing the multistep prediction job in this dataset. The data we study is sampled in an hourly round, where we use one day's 24-hour data to predict the 24-hour data of the next day. If only high-dimensional data are predicted at a future time step, such a complex system often cannot adjust power generation and transmission in time and cannot solve the problem in time. If we apply long short-term memory (LSTM) model in transmission system and only the single-step data prediction[5] by LSTM is used as a new data for cyclic prediction, the error is relatively large because of superposition effects, which makes system state incorrectly inferred.

On this condition, in order to lower the prediction error, we design a new multistep input and multistep output network model, which shows incomparable advantage in handling high-dimensional data and remarkable suitability to solve our problems. In addition, since sequence-to-sequence network have increasingly important application in machine translation, speech recognition, and text understanding, our design also improves the conventional sequence-to-sequence model and combine it with proposed multistep network model by using encoder network to sequentially read the sequence information and by using the decoder network to sequentially output the desired targets.

Contributions of this article can be summarized as follows:

- Improve the common sequence to sequence model to do the multistep prediction work for high-dimensional electric time series.
- Add statistic information to the above model to make a multiscale network and increase the accuracy.
- Use multitarget learning method to improve the result of complicated time series prediction problem.

The rest of this article is organized as follows. Section 2 is the related work that consists of the related internet of things work and related data analysis work. Section 3 is the background of transmission data. Section 4 is the definition of our problem. Section 5 introduces the common LSTM prediction network. Section 6 examines the improved sequence to sequence model, which is suitable for high-dimensional time series multistep prediction. Section 7 shows the multiscale and multitarget network. Section 8 describes the experimental result. Section 9 draws the brief discussion.

2 | RELATED WORK

Smart grids or power system plays an important role in the area of internet of energy research. Tsoukalas and Gao[6] described a virtual storage of energy system and a Consortium for Intelligent Management of Electric-power Grid (CIMEG) system. Their architecture modeled the constitution of smart grid, such as communication capability, forecasting capability, multiresolution agents, and short-term price elasticity. Baker et al[7] proposed an algorithm to evaluate the city map-based vehicular ad hoc network scenarios. They designed a routing protocol for vehicle communications based on GreeDi to reduce the power consumption and used a realistic urban scenarios to evaluate their framework. Their work made a contribution to the energy consumption in the connection of vehicles to data centers. Bui et al[8] described a web-enabled smart grid technology concept, which enabled wireless networks of tiny embedded objects. They expounded the prominent scenarios for Smart Grid, proposed a network solution for the Internet-based Smart Grid based on IPv6 over Low-Power Wireless Personal Area Networks, and presented an efficient XML Interchange framework. Song et al[9] briefly reviewed the representative LPWAN technologies of narrowband Internet of Things and remote (LoRa) technologies. Based on the main technical characteristics of LPWAN, a wireless-to-cloud architecture is proposed for IoET to achieve renewable greater integration of energy systems.

Through the progress of above work, the power system components are linked together and collect much useful data. Ali et al[10] proposed a fine-grained partial offloading based offloading scheme. In the background of mobile edge computing, they provided an effective mechanism for computational offloading and a mathematical model for local cost of different energy conditions and offloading cost in varying network conditions. Based on the above result, authors proposed a deep learning algorithm to train the data generated by their mathematical model and used the result to do the decision making. Their work inspired the idea that the deep leaning method can be used in the analysis of the internet of energy. In order to better analyze the historical time series data and accordingly make corresponding improvements to the system, many prediction method are applied, trying to achieve this goal. GARCH model is the extension model of ARCH model on the basis of statistical ARIMA, which helps solve the problems caused by the assumption of constant variance. It is commonly used to predict multiple-variable time series. The ARIMA methodology has been used in predicted next-day electricity prices,[11] short-term cloud coverage,[12] and electric energy consumption.[13] Some search also used GARCH to predict financial series[14,15] and cryptocurrency.[16] However, when faced with such complex problems at high-latitude output, it becomes a tough work for such statistic model. It is extremely complex to analyze thousands of dimensions in static way and it is infeasible to make univariate prediction for each sequence. de Arquer Rilo et al[17] used the neural network method to do the one-step prediction of the SP 500 stock market and the electrical power data; they used different multilayer perceptions, recurrent neural networks (RNN), and LSTM networks to predict such data and had a conclusion that LSTM had the best result. LSTM[18] is a special memory-maintained RNN, which has fabulous effects and useful application in predicting time series.[19] As the LSTM model, most prediction methods at the present stage is based on single-step data prediction,[19,20] which may unavoidably result in some errors when we apply them to multiple-step and cyclic prediction. Multiple-step prediction requires the combination of single-step predictions, which correspondingly accumulates errors in each single-step prediction. For better adjusting proper power supply and correcting potential faults, more future data after different certain time needs to be predicted in power transmission system.

3 | DATA DESCRIPTION

The renewable energy Europe dataset[21] models the continental European electricity system, including transmission system data, wind and solar observations, economic characteristics and related information. This dataset is collected in the stable frequency without missing point and all dimensions are clock synchronization. On the one hand, the stable frequency and clock synchronization make it possible to change the data frame into the supervised form without deviation in smoothing process. On the other hand, the intact and high-quality data can judge the model precisely without handling the outliers. For the above reasons, this dataset is the most suitable one to do the multistep prediction problem.

The transmission system data play an important part in this dataset. Such kind of data is collected from the transmission network of 1494 transmission buses. This system models the hourly averaged renewable infeed of 3 years, which means the system collect 1494 dimensions of data per hour and the dataset has 1096 time-step. In the application, researchers should use the previous data to predict the future data of each dimension. The prediction especially the hours or days of prediction of high dimensions (1494 dimension) transformation data can simulate the demand of such system and help us to satisfy the indeed more efficiently. The IOE makes this system face many challenges and opportunities.

4 | PROBLEM FORMATION

First, we model a network as a multistep input and output function $Y = f(X)$ according to our predicting objective.[22] Since we are intended to predict the next day's 24-hour data with the previous day's 24-hour data, we define X as observed time series data of all dimensions in one day and Y as predicted time series data of all dimensions in subsequent next day. Therefore, in function $Y = f(X)$, X and Y are both 1487*24 matrix, where $X = x_1, x_2.., x_t, .., x_n \in R^{m*n}$ denotes the set of input 24-hour data of dimensions and $Y = y_1, y_2 \dots y_t, .., y_n \in R^{m*n}$ denotes the set of multiple prediction result of 24-hour data of dimensions calculated according to the relation between each hour's data in two consecutive days. Considering all valid dimensions contained in each collected transmission line data, each vector x_t or y_t represents 1487 dimensions collected by 1494 sensors excluding seven abnormal dimensions. With the setting as above, we can formally present our observed transmission line data over 24 hours quantitatively as

$$X = \begin{bmatrix} x^{1,1} & x^{1,2} & \dots & x^{1,24} \\ x^{2,1} & x^{2,2} & \dots & x^{2,24} \\ \vdots & \vdots & \ddots & \vdots \\ x^{1487,1} & x^{1487,2} & \dots & x^{1487,24} \end{bmatrix}, \tag{1}$$

where x^{ij} measures transmission line data of ith dimension at jth hour. Similarly, we can also donate the predicted output data quantitatively as

$$Y = \begin{bmatrix} y^{1,1} & y^{1,2} & \dots & y^{1,24} \\ y^{2,1} & y^{2,2} & \dots & y^{2,24} \\ \vdots & \vdots & \ddots & \vdots \\ y^{1487,1} & y^{1487,2} & \dots & y^{1487,24} \end{bmatrix}, \tag{2}$$

where y^{ij} donates same form of transmission line data but predicted ones in the next day.

In addition, in order to raise the accuracy of prediction and find a more general changing principle, we pay attention to periodic change of weekly detected data and combine them with multistep prediction results to gain a weighted output as improvement.

We consider each day's data presents norm distribution and calculate the mean and variance of each day. As we can see from Table 1, the distribution is different among different days. If we let each day's data divide each day's mean, we can change the different day's distribution to a similar one.

Instead of costing excessive unnecessary time and efforts to do 1487-dimension intermediate calculation, we use 7*1 matrix as input for whole week's historical data of mean value of dimensions.

In our multiscale and multitarget network, we first use Seq2Seq model to predict all the day's data as the common day to do the learning of first stage, then we add history statistic mean value to multiply the output of Seq2Seq and use another fully connected layer to remap this combination output to generate the different distribution, which is the learning of second stage. We use the sum of two-stage learning loss be the total training loss. The maximum and minimum bound the data range. w^1 and w^2 are the weight of loss in each stage. In our work, we set w^1 be 0.8 and w^2 be 1.0. Therefore, we can

TABLE 1 Mean and variance of different week days

	Mean	Variance
Sunday	0.30809906	0.02166415
Monday	0.46426434	0.04417349
Tuesday	0.49088063	0.03993091
Wednesday	0.49343006	0.03982598
Thursday	0.48971482	0.03974915
Friday	0.47846683	0.03712029
Saturday	0.37232333	0.02264035

TABLE 2 Notations

Notations	Meaning
MSE	mean squared error
X	input data matrix
Y	output data matrix
LSTM	long short-term memory
IoE	internet of energy
RE-Europe	renewable energy Europe dataset
x_i	the ith line's data input vector
$x^{i,j}$	the ith line's jth time-step input data
y_i	the ith line's data output vector
$y^{i,j}$	the ith line's jth time-step output data
w^*	the parameter of one dimension data
W_*	the parameter of vector data
h_t	the hidden state of LSTM cell
b_*	the bias of each calculation in the network cell
O_*	the output of cell or layer
$\sigma(*)$	the sigmoid activation function
tanh	the tanh activation function
Seq2Seq	sequence to sequence

formally improve the multistep network $Y = f(X)$ by adjusting it to a new function $Y = w^1 * \text{static\_info} * f(X) + w^2 * f(X)$ with the foundation of above prediction based on historical statistic of per week's data, where X represents observed transmission line data of 1487 dimensions over 24 hours, $f(X)$ represents calculated transmission data through sequence-to-sequence multiple prediction, static_info represents the corresponding next day's mean of the seven sequential days, and w^1, w^2 represent the weight of directed learning and the normal learning. In our model, we use the Seq2Seq structure to generate the common distribution of the data and use the multiscale information to lead each day's predict to a specific direction and generate the corresponding distribution. We use MSE to evaluate our framework (Table 2).

5 | LONG SHORT-TERM MEMORY NETWORK

The traditional feed-forward neural network is mapping from one input matrix to one output matrix, but the RNN[23,24] is mapping an input sequence $X = (x_1, x_2 \ldots, x_T)$ to another output sequence $Y = (y_1, x_2 \ldots, y_T)$, which contains all the information learned from the whole input sequence by iteration:

$$h_t = \sigma(W_x x_t + W_h h_{t-1}), \tag{3}$$

$$y_t = W_y h_t, \tag{4}$$

where $\sigma(\bullet)$is the sigmoid function.

This architecture made RNN can easily be used in mapping sequence to sequence. However, as the length of input sequence increase, the gradient of RNN is vanishing, and the error is blowing up, the standard RNN cannot store information for long periods of time or access the long range of context.

Sepp et al and Jurgen et al[18]proposed a novel, efficient, gradient-based method called “long short-term memory” (LSTM). LSTM solved the vanishing gradient and error blowing up problems by using additional gates. These gates also

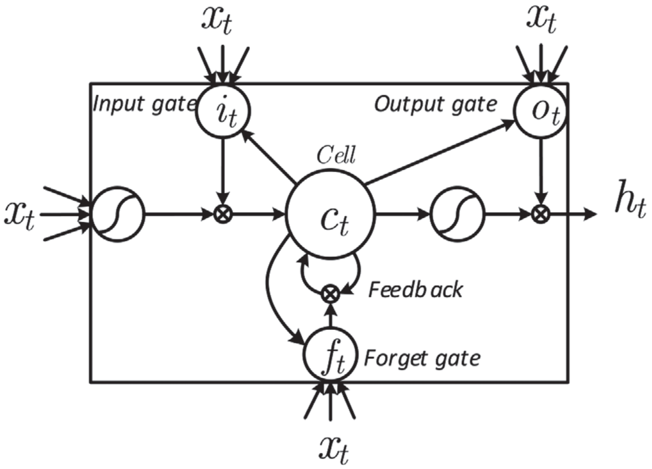

FIGURE 1 Illustration of long short-term memory block

determine when the input is significant enough to remember, when it should continue to remember or forget the value, and when it should output the value. The fact proves that the LSTM has the capability of storing and accessing information over very long timespans.

Figure 1 illustrates a typical LSTM memory block. It contains one self-connected memory cell c and three gates: the input gate i, the forget gate f, and the output gate o, which can store and access the long-range contextual information of a temporal sequence. The activations of the memory cell and three gates are given as follows:[25]

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i), \tag{5}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{6}$$

$$c_t = F_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{7}$$

$$O_T = \theta W_{xo}x_t + Whoh_{t-1} + W_{co}c_t + b_o, \tag{8}$$

$$h_t = O_t \tanh(C_t) \tag{9}$$

Similar to RNN, $\sigma(\bullet)$ is the sigmoid function and all the matrices W are the connection weights between two units. By this formulation, LSTM has the ability to learn information with long temporal dependencies.

As a kind of vital network, LSTM plays an important role in feature extraction. In our research, in each time step of time series processing, the LSTM cells load the preprocessed data, update the weight of parameters such as $W_{xi}x_t$, $W_{hi}h_{t-1}$, $W_{ci}c_{t-1}$, W_{xc}, and so on, and also update the cell gate state such as f_t, O_T, h_t, and so on.

The network based on the LSTM cells plays an important part in sequence analysis. In the common LSTM prediction network, researchers use LSTM cells to do the feature extraction for all time steps and use the fully connected cells to generate the output after the final time step. In our sequence to sequence model, we make the LSTM cells be the encoder layer to extract the information of input data and make the LSTM cells be the decoder to generate the data per future time step.

6 | SEQUENCE TO SEQUENCE NETWORK

In machine translation field, although LSTM is powerful and flexible, it can only be used to solve the problems whose inputs and targets are encoded into vectors of fixed length. To break this limitation, Sutskever et al[26] presented a novel model called sequence to sequence model, also was known as RNN encoder-decoder[27] which had made a remarkable improvement in machine translation field. As shown in Figure 2, the main idea of sequence to sequence network is that use a RNN called “encoder” to process the input sequence through all time steps and obtain a large vector representing the whole input sequence, then use another RNN called “decoder” to extra information from this vector and generate output sequence by time steps.

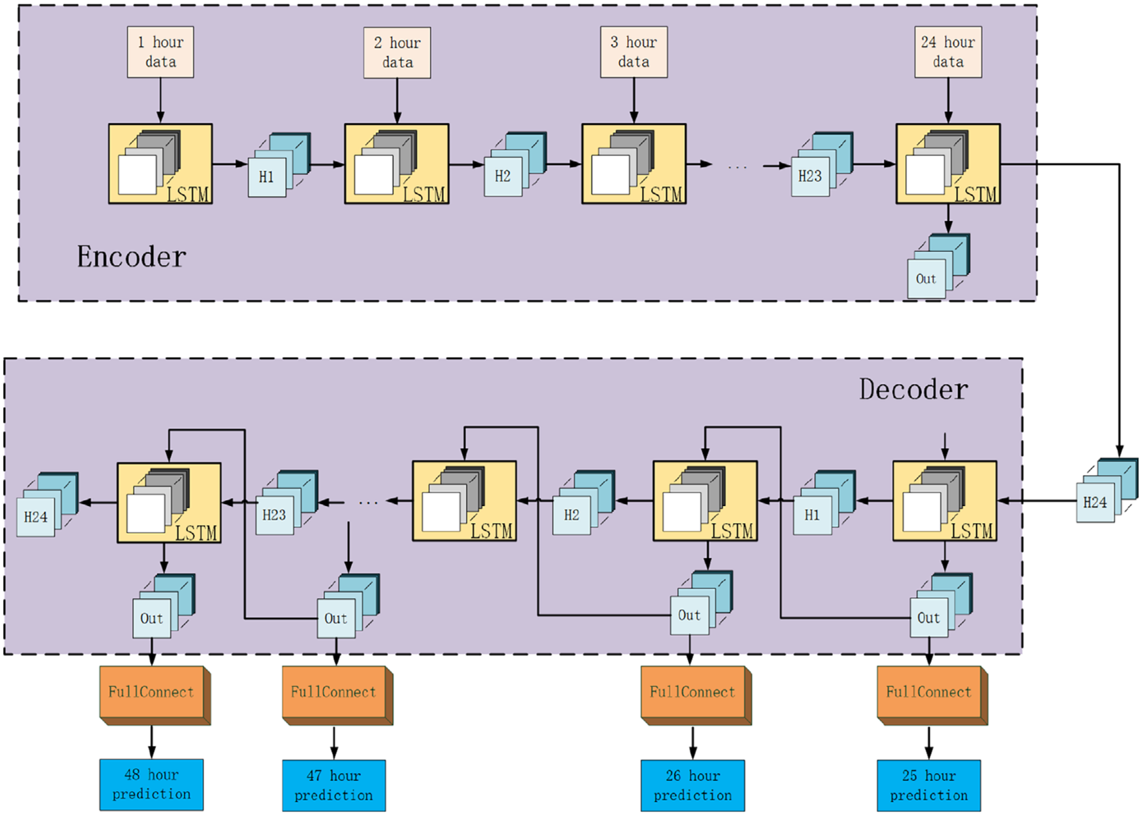

FIGURE 2 sequence to sequence model in machine translation

In this model, LSTM is used to replace the above RNN in both encoder and decoder. The role of sequence to sequence learning can be seen as an estimator of the conditional probability $p(y_1, \ldots, y_{T'}|x_1, \ldots x_T)$ in which $(x_1, \ldots, x_T)$ is the input sequence and $(y_1, \ldots y_{T'})$ is the target sequence which may not be equal length to the input.

In this model, first, the encoder LSTM computes this probability through obtaining the fixed large dimension representation v, which is the last hidden state of encoder LSTM related to the input sequence $(x_1, \ldots, x_T)$. Then, the probability of $y_1, \ldots y_{T'}$ can be computed by the decoder LSTM with an initial hidden state of v. The formulation is:

$$p(y_1, \ldots, y_{T'}|x_1, \ldots x_T) = \prod_{t=1}^{T'} p(y_t|v, y_1, \ldots, y_{t-1}). \tag{10}$$

In the above equation, $p(y_t|v, y_1, \ldots y_{t-1})$ is calculated by a softmax over all training dataset.

In our work, we change the output be the expected value, not the probability. We use the fully connected layer to get the information and return the output.

$$f(y_1, \ldots, y_{T'}|x_1, \ldots x_T) = \prod_{t=1}^{T'} f(y_t|v, y_1, \ldots, y_{t-1}). \tag{11}$$

The encoder-decoder model can not only map input sequence to output sequence with different length but also increase the learning ability by the increasement of the number of model parameters.

7 | MULTISCALE AND MULTITARGET NETWORK

7.1 | Multiscale network

Multiscale deep neural networks are a deep learning method widely used in complex problems. This method is widely used in problems such as target detection. In the literature,[28] the author describes the main strategies of multiscale deep

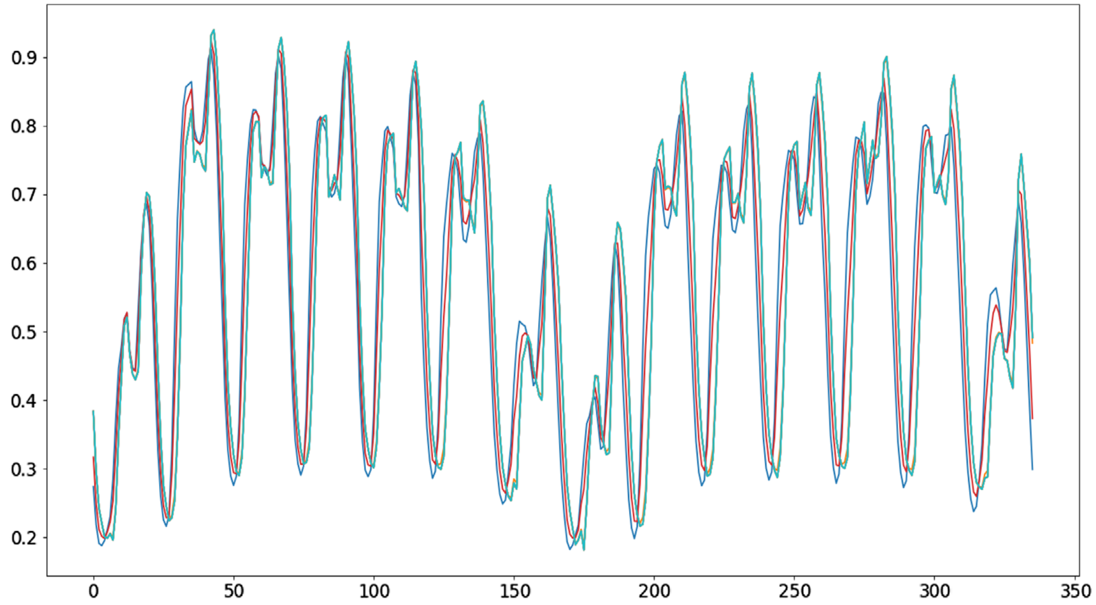

FIGURE 3 The example of transmission data that are collected by hour

neural networks in target detection: (i) Using a single classifier to rescale the image multiple times and perform feature calculations on multiple scales so that the classifier can match all possible object sizes. (ii) Applying multiple classifiers to a single output image. In the application of target detection, detection is performed at each intermediate network layer and its sensing domain is matched with various objects, so that all problems can be detected by feeding forward a single image through the network.

In the transmission data we study, we find that the transmission data have a time domain relationship. Figure 3 shows the example of 10 transmission data series that are collected by hour and the data of every 24 time steps means the sample of one nature day. This figure contain 336 time steps of data, which means the data re obtained for 2 weeks. As we can find from the figure, the data of each day have the relationship with the data of other days. At the same time, the regularity of data is unstable. Such unstable information makes the common model difficult to have an accurate prediction data of the next day. If we only consider the time domain relationship of data among hours to make the prediction, the volatility of the data will seriously affect the prediction performance of the model.

In another scale, we can observe that there exists a regularity among weeks. The transmission data in workday have a similar range and the data in the weekend have the similar range. In this case, we calculate the daily mean, minimum, medium, and maximum values. Figure 4 shows an example of the minimum transmission value of 10 transmission lines of each day. Each point means the minimum value of that day. There are 49 days of data in the figure. The data series have a period of seven, such domain of data can describe the transmission system in another aspect. In addition, the mean, medium, and maximum values have the similar tendency as well. We find that the additional indicators of these statistics have regular fluctuations in the weekly cycle. The fluctuation of these statistics can describe the overall data range of the day to a certain extent.

Therefore, based on the idea of multiscale network, we propose a deep neural network with additional scale information to improve the prediction accuracy of the model.

Figure 5 shows the multiscale network used to predict the high-dimensional transmission data series. The encoder layers are same as the structure in above sequence-to-sequence model. We use fully connected layers to get the information from LSTM layers and return the output O_{FC1}. At the same time, we add the extra information i_{extra} such as the mean or minimum value among days described in the above paragraph to the model, such kind of data is multiplied by O_{FC1}. Then we add a concat layer to concatenate these information. Finally, we use a new fully connected layer to return the output.

$$\text{Ouput} = W_1^t(O_{\text{FC1}}) + W_2^t(O_{\text{FC1}}i_{\text{extra}}) + w_3^t(i_{\text{extra}}) + w_4. \tag{12}$$

Adding this additional information to the network can guide the training of the model and improve the accuracy of the network.

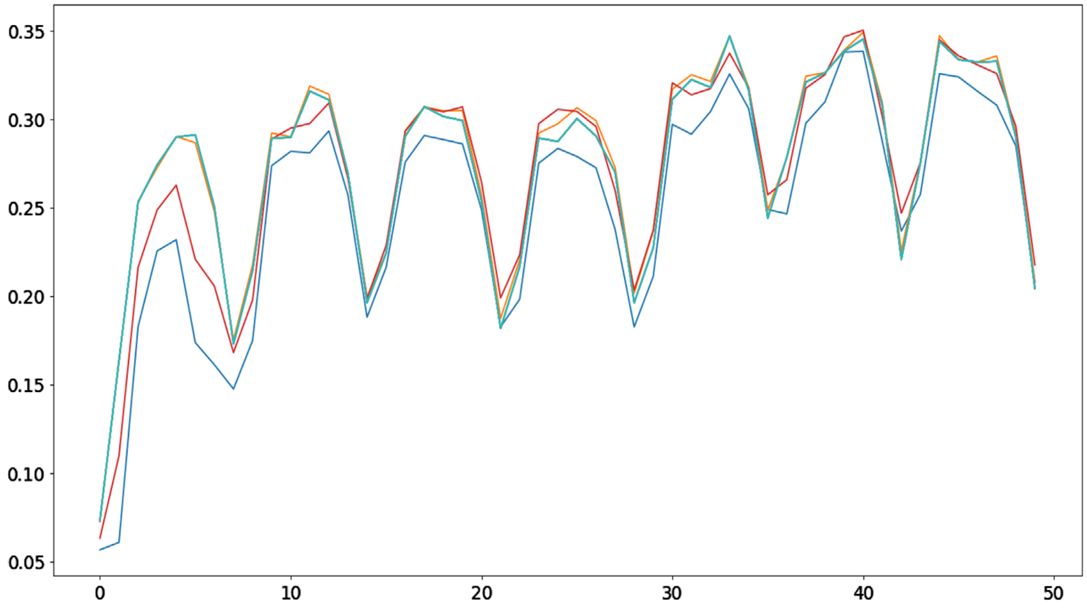

FIGURE 4 The example of each day's minimum normalized value of five transmission lines

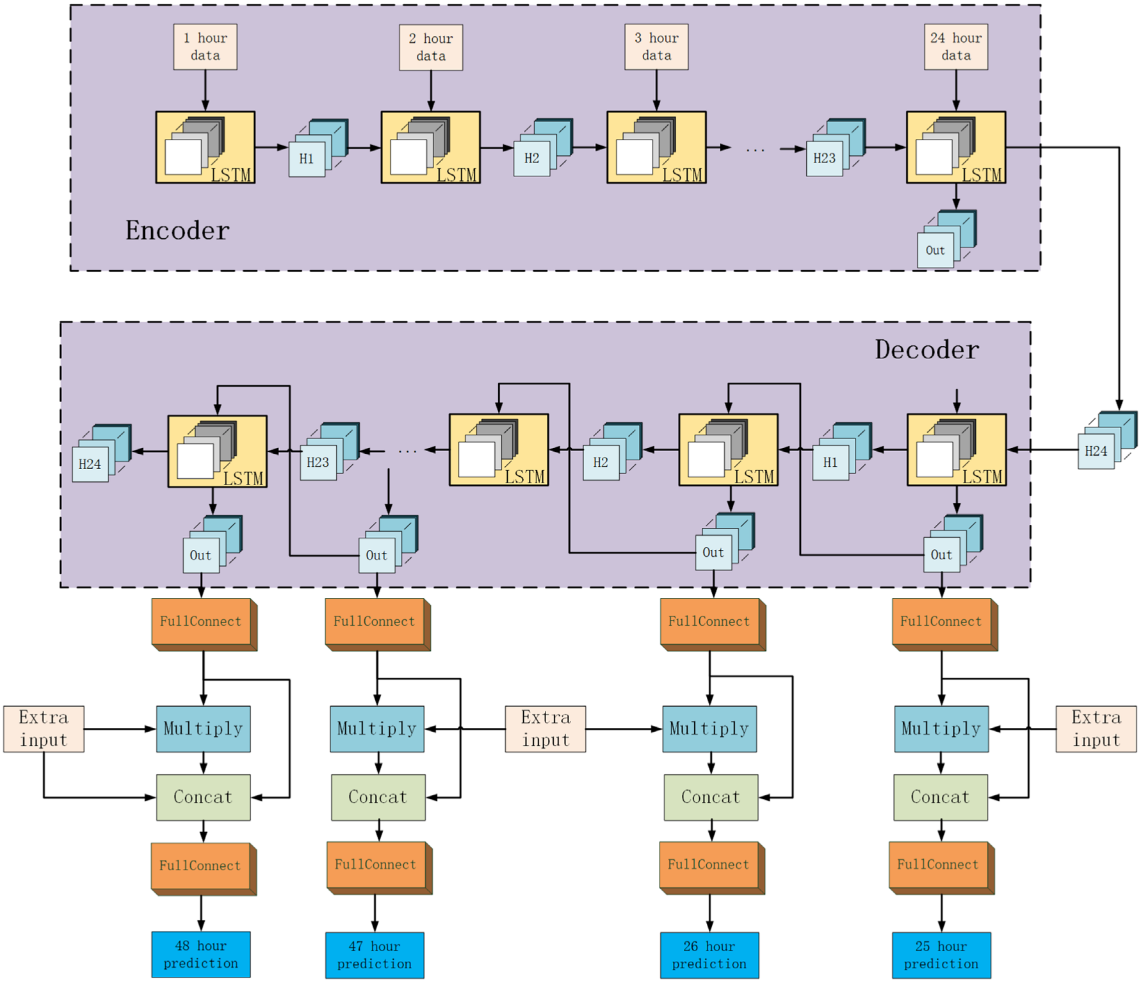

FIGURE 5 The Seq2Seq based multiscale network

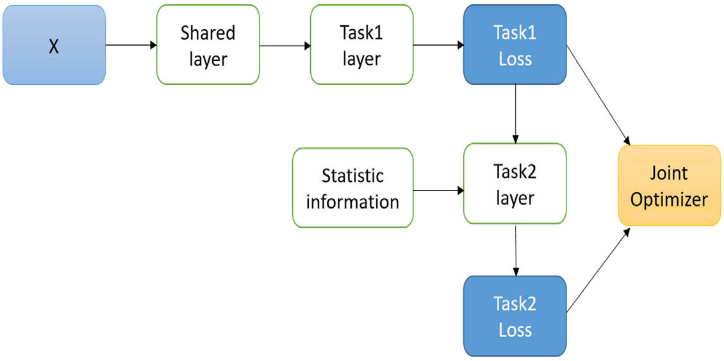

FIGURE 6 The multitarget learning framework

7.2 | Multitarget network

Multitasking has made great progress in the field of machine learning and has a wide range of applications in natural language processing, computer vision, and bioinformatics.

In the task of machine learning, we care about the optimization of specific indicators. In the previous research of complex problem, we optimized the subproblem, the single model or the model set. This method splits a complex problem into multiple subproblems, which are often independent. Such models tend to ignore the potential relevance of parameters between models. Multitask learning often puts multiple indicators of complex problems in the same model for optimization. Each indicator shares many model parameters, which can better abstract and solve complex problems.

Multitask learning is motivated by two different aspects. Biologically, we can learn from human activity that people commonly apply the knowledge acquired from old tasks to help do an interrelated task. From a machine learning point of view, multitask learning can be regarded as a type of inductive bias retrieved from auxiliary tasks which prefer a hypothesis that leverages the performance of all tasks.

For deep learning methods, multitask learning is typically implemented by hard or soft parameter sharing of hidden layers of neural network.[29]

For hard-parameter sharing, multiple neural networks share the several hidden layers and own their task-specific parameter on top layers for the purpose of capturing a common representation for all models. This method is a intuitionistic implementation because it is a proven commodity that lower layers of a neural network learns those basic features. At the same times, training models together alleviate the impact of overfitting.

For soft-parameter sharing, each model conceives separated parameter including hidden layers. Constraints technique like l_1 distance or l_2 distance is added to hidden layers of different deep neural networks to force them to learn similar parameters. This idea is inspired by regularization in traditional machine learning method (Figure 6).

Based on the idea of multitarget network, we make the network be a two-stage learning model. Such model combines the Seq2Seq output and the multiscale output and makes the weight loss of them be the total loss. This trick guides the learning direction of model and improves the model performance.

8 | EXPERIMENT AND RESULTS

8.1 | Data normalization

The high-dimensional transformation data are collected from different area of Europe mainland; the regional differences lead to differences in the data. The range of each dimension's data is different. After statistics, we find that the smallest range of data is from 0 to 1 and the largest range is from 2868.5439 to 9930.2408. In order to reduce the impact of data size on our network model, we use the min-max normalization method to preprocess the data. The normalize formulation is:

$$x^{ij*} = \frac{x^{ij} - \min(x^i)}{\max(x^i) - \min(x^i)}. \tag{13}$$

In the data normalization, we find that there are eight dimensions of data, which is invariable or not numeric type. Hence, we finally analyze the common 1487 dimensions.

8.2 | Data reshape

The original data are a 1495 × 26304 matrix. We change the data feature and data slice into the supervised learning samples. First, we select every 24 steps of data be the part of input X, which means the known data of previous day and we choose the following 24 steps of data be the output Y, which means the data to be predicted of next day. Second, we calculate the mean value of each day in a week and store seven values. We make the mean value of the corresponding day of a week be part of input X. Finally, our input is a matrix with a 1487 × 24 matrix and a statistic parameter, our output is a 1487 × 24 matrix.

8.3 | Experiment and discussion

Sections 4–6 expound the main research framework and the improvement. In order to demonstrate the usefulness of our improvement, we have several experiment to make comparisons. The details of experiments are shown as follows:

- Sequence to sequence network: We set 2048 LSTM cell be the encoder structure of Seq2Seq model and 2048 LSTM cell be the decoder structure. The input of this model is 1487 × 24 matrix of the first day and the output is 1487 × 24 matrix of second day. We use 1487 dimensions of fully connected layer to load the input vector. The connected layer and the LSTM encoder layer are linked together so that the input information can be load to the architecture properly. In each encoder time step, we input each hour's data to calculate and update the parameter. In each decoder time step, we let the last time step LSTM cell's output be the following LSTM cell's input and then calculate and update the parameter. The output of decoder is loaded by a 1487-dimension fully connected layer and such layer returns the final predicted value. The loss function is:

$$Loss(X, Y) = \frac{1}{n}\sum_{i=1}^{n}(f_{\mathrm{seq2seq}}(X) - Y)^2. \tag{14}$$

- Multiscale network: The basic of this network is similar to the Seq2Seq model. When the whole Seq2Seq network calculate the output, we use the extra information multiply each dimension of this output and use a 1487 × 2 dimension concat layer to concatenate the multiply result with the previous output. Finally, a 1487-dimension fully connected layer loads the concat layer's output and returns the final result.
- Multiscale and multitarget network: The basic of this network is similar to the multiscale model. The difference is that this model is a two-stage learning method. We consider the output of both Seq2Seq model and the multiscale model be the final loss. We calculate the weighted loss of two stage and do the backpropagation. The loss function is:

$$\mathrm{Loss}(X, Y) = \frac{1}{n}\sum_{i=1}^{n}[w_1 \times (f_{\mathrm{seq2seq}}(X) - Y)^2 + w_2 \times (f_{\mathrm{multiscale}}(X) - Y)^2]. \tag{15}$$

In our experiment, we set w_1 be 0.8 and w_2 be 1.0.

In the contrast experiment, we realized the above networks. We train each network for 500 epochs, the first 100 epochs' learning rate is 0.02 and the later 400 epochs' learning rate is 0.0005. The loss function is MSE, the optimizer we use is Adam and the batch size is 256.

Table 3 show the MSE of each models' train and test loss. Table 4 shows the MSE of each models' train and test loss of one-step presiction.

TABLE 3 Comparison of the MSE of different methods of multistep prediction

Model	Train loss	Test loss
Seq2Seq	4.33×10^{-6}	1.49×10^{-5}
Multiscale	1.78×10^{-6}	1.23×10^{-5}
Multiscale + multitarget	1.56×10^{-6}	1.18×10^{-5}

TABLE 4 Comparison of the MSE of different methods of one-step prediction

Model	Train loss	Test loss
Basic LSTM[5]	3.78×10^{-6}	4.62×10^{-6}
Seq2Seq (average)	1.80×10^{-7}	6.21×10^{-7}
Multiscale (average)	7.42×10^{-8}	5.13×10^{-7}
Multiscale + multitarget (average)	6.5×10^{-8}	4.92×10^{-7}

As shown in Table 3, the Seq2Seq model has a wonderful performance in our problem. The common Seq2Seq is a widely used language model, which change the word into embedded vector and treat the sentence as a vector sequence. In our work, we change the original embedding layer into fully connected layer that we can map the input data into a proper space and transfer such information to the encoder layer. The original Seq2Seq can be considered as a kind of classification problem, which selects the proper word step by step. In our model, by changing the output layer and the loss function, we change this classification problem into a prediction problem. According to the experiment, the result confirms that the Seq2Seq multistep prediction model has a good performance.

In the research of time series prediction, there exists a common condition that the series in different states have different features. In the area of time series classification, such different features can be used to classify the different pattern. However, in the area of prediction, the unstable feature may affect the robustness of the model. Take our problem for example, use the Wednesday's data to predict the Thursday's data, use the Friday's data to predict the Saturday's data, and use Sunday's data to predict the Monday's data are different. The input data distribution and the output data distribution are different. The common prediction loss function is $\text{Loss}(X, Y) = \frac{1}{n} \sum_{i=1}^{n} (f_{\text{seq2seq}}(X) - Y)^2$. In imbalance dataset, if we do not do the extra processing, the common Seq2Seq will generate all the conditions as the highest proportion distribution. However, the statistics information and the multiscale idea guide the learning direction in different conditions and improve the precision of Seq2Seq model obviously.

When we use the multitarget trick, we have a faster rate of convergence and the performance become better as well. Such multitarget network can help us understand the learning section of the architecture as well.

However, the common one-step prediction model cannot forecast the power demand or production in a few hours or days to make adequate preparation. If we take each time step's result to make a comparison from Table 4, we can find that our model is more precise than the basic LSTM prediction model.[5]

9 | CONCLUSION AND FUTURE WORK

The analysis and prediction of transformation data plays an important part in the power system. Thousands of transmission lines, renewable power sources, and traditional stations make up the intricate power system, which need to be evaluated in automatic detection method. In this back ground, it is necessary for us to build a high-precision multistep prediction model to forecast the status of system in the future. In this article, we propose a multistep prediction network framework for electricity transmission system data. We transfer the widely used sequence to sequence algorithm in natural language processing to the area of multistep prediction of high-dimension power transmission data creatively. We change the original word selected problem into a sequence prediction problem and achieve a success. The MSE result is 1.49×10^{-5}, which shows that our model has a satisfactory performance. The data range is different between weekday and weekend that we cannot use the sample Seq2Seq network to predict all the conditions accurately enough. After the statistical analysis, the extra statistic information is added to the above network, which makes our model a multiscale network and such information guides the range of output in a reasonable way. Through multiscale information, we decrease the MSE to 1.23×10^{-5}. In addition, we use a multitask learning trick to make our network be a model with two-stage loss. This trick guides the learning direction of specific layer and makes the model more accurately with a loss of 1.18×10^{-5}. The experiments certify the validity of our model. With the development of the explanations of network, we can do more research in analyzing the reason and the statistic law between the network in the future.

ACKNOWLEDGMENT

This work is partially supported by Strategic Priority Research Program of Chinese Academy of Sciences, Grant No.XDC02070800, No.XDC02070700 and NSFC Youth Fund 61704179.

ORCID

Hanlin Zhu https://orcid.org/0000-0002-8954-3479

REFERENCES

1. Yin Y, Sun Y, Yu H, Bi Z, Xu B, Cai H. PCA based energy network temporal and spatial data analysis and prediction. Paper presented at: Proceedings of the International Conference on e-Business Engineering; 2020:590-605; Springer, Cham.
2. Wang K, Li H, Feng Y, Tian G. Big data analytics for system stability evaluation strategy in the energy internet. *IEEE Trans Ind Inform*. 2017;13(4):1969-1978. https://doi.org/10.1109/TII.2017.2692775.
3. Pinson P, Jensen T. RE-Europe, a large-scale dataset for modeling a highly renewable European electricity system. *Scientific Data*. 2017;4:170175. https://doi.org/10.1038/sdata.2017.175.
4. Syranidis K, Markowitz P, Linssen J, Robinius M, Stoltcn D. Flexible demand for higher integration of renewables into the european power system. Paper presented at: Proceedings of the 2018 15th International Conference on the European Energy Market (EEM); 2018:1-6. https://doi.org/10.1109/EEM.2018.8469962.
5. Cao Z, Zhu Y, Sun Z, et al. Improving prediction accuracy in lstm network model for aircraft testing flight data. Paper presented at: Proceedings of the 2018 IEEE International Conference on Smart Cloud (SmartCloud); 2018:7-12; IEEE. https://doi.org/10.1109/SmartCloud.2018.00010.
6. Tsoukalas LH, Gao R. From smart grids to an energy internet: assumptions, architectures and requirements. Paper presented at: Proceedings of the 2008 3rd International Conference on Electric Utility Deregulation and Restructuring and Power Technologies; 2008:94-98; IEEE. https://doi.org/10.1109/DRPT.2008.4523385.
7. Baker T, García-Campos JM, Reina DG, et al. GreeAODV: an energy efficient routing protocol for vehicular ad hoc networks. Paper presented at: International Conference on Intelligent Computing; 2018:670-681; Springer, Cham.
8. Bui N, Castellani AP, Casari P, Zorzi M. The internet of energy: a web-enabled smart grid system. *IEEE Netw*. 2012;26(4):39-45. https://doi.org/10.1109/MNET.2012.6246751.
9. Song Y, Lin J, Tang M, Dong S. An internet of energy things based on wireless LPWAN. *Engineering*. 2017;3(4):460-466. https://doi.org/10.1016/J.ENG.2017.04.011.
10. Ali Z, Jiao L, Baker T, Abbas G, Abbas ZH, Khaf S. A deep learning approach for energy efficient computational offloading in mobile edge computing. *IEEE Access*. 2019;7:149623-149633. https://doi.org/10.1109/ACCESS.2019.2947053.
11. Contreras J, Rosario E, Francisco N, Antonio C. ARIMA models to predict next-day electricity prices. *IEEE Power Eng Rev*. 2002;22:57-57. https://doi.org/10.1109/MPER.2002.4312577.
12. Wang Y, Wang C, Shi C, Xiao B. Short-term cloud coverage prediction using the ARIMA time series model. *Remote Sens Lett*. 2018;9:275-284. https://doi.org/10.1080/2150704X.2017.1418992.
13. Erick O. Oliveira Fernando Luiz. forecasting mid-long term electric energy consumption through bagging ARIMA and exponential smoothing methods. *Energy*. 2018;144:776-788. https://doi.org/10.1016/j.energy.2017.12.049.
14. Bauwens L, Laurent S, Rombouts J. Multivariate GARCH models: a survey. *J Appl Econ*. 2006;21:79-109. https://doi.org/10.2139/ssrn.411062.
15. Jingjia C, Reg K, Hao Y. Modelling the common risk among equities: a multivariate time series model with an additive GARCH structure. 2016:205-218. https://doi.org/10.1007/978-981-10-2594-5_12.
16. Conrad C, Custovic A, Ghysels E. Long- and short-term cryptocurrency volatility components: a GARCH-MIDAS analysis. *J Risk Financ Mgmt*. 2018;11(2):1-12.
17. Arquer RJ, Hussain A, Al-Taei M, Baker T, Al-Jumeily D. Dynamic neural network for business and market analysis. Paper presented at: Proceedings of the International Conference on Intelligent Computing; 2019:77-87.
18. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput*. 1997;9:1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735.
19. Graves Alex. Generating sequences with recurrent neural networks; 2013. arXiv preprint arXiv:1308.0850.
20. Zhu H, Zhu Y, Wu D, et al. Correlation coefficient based cluster data preprocessing and LSTM prediction model for time series data in large aircraft test flights. Paper presented at: Proceedings of the 3rd International Conference, SmartCom 2018; December 10-12, 2018:376-385; Tokyo, Japan.
21. Jensen TV, Sevin H, Greiner M, Pinson P. The RE-Europe data set. 2015;. https://doi.org/10.5281/zenodo.35177.
22. Zhao J, Qu H, Zhao J, Jiang D. Spatiotemporal traffic matrix prediction: a deep learning approach with wavelet multiscale analysis. *Trans Emerg Telecommun Technol*. 2019;30:e3640. https://doi.org/10.1002/ett.3640.
23. Rumelhart DE, Hinton GE, Williams RJ. *Neurocomputing: Foundations of Research*. Cambridge, MA: MIT Press; 1988:696-699.
24. Werbos PJ. Backpropagation through time: what it does and how to do it. *Proc IEEE*. 1990;78(10):1550-1560.
25. Gers F, Schraudolph N, Schmidhuber J. Learning precise timing with LSTM recurrent networks. *J Mach Learn Res*. 2002;3:115-143. https://doi.org/10.1162/153244303768966139.
26. Sutskever I, Vinyals O, Le Quoc V. Sequence to sequence learning with neural networks. *Proceedings of the 27th International Conference on Neural Information Processing Systems*. 2014;2:3104-3112.
27. Cho K, Merriënboer B, Gulcehre C, Bougares F, Schwenk H, Bengio Y. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation; 2014. arXiv preprint arXiv:1406.1078. https://doi.org/10.3115/v1/D14-1179.

28. Min C, Qian X, Jianming L, Wenyin L, Qing L, Jianping W. MS-LSTM: a multi-scale LSTM model for BGP anomaly detection. 2016;:1-6. https://doi.org/10.1109/ICNP.2016.7785326.
29. Ruder S. An overview of multi-task learning in deep neural networks; 2017. arXiv preprint arXiv:1706.05098.

How to cite this article: Zhu H, Zhu Y, Wang H, et al. Multiscale deep network based multistep prediction of high-dimensional time series from power transmission systems. *Trans Emerging Tel Tech*. 2020;e3890. https://doi.org/10.1002/ett.3890