Short-term Wind Power Prediction Model Based on Encoder-Decoder LSTM

Kuan LU¹,a, Wen Xue SUN², Xin Wang¹, Xiang Rong MENG¹, Yong Zhai³, Hong Hai Li³, Rong Gui ZHANG³

¹ State Grid Shandong Electric Power Research Institute, Jinan, Shandong China.
² State Grid Zhangqiu Power Supply Company, Jinan, Shandong, China.
³ Shandong Luneng Software Technology Co., LTD, Jinan, Shandong, China.
a Email: lk83@163.com

Abstract. We propose a long short-term memory (LSTM) network based encoder-decoder (E-D) model for wind power prediction (WPP). The LSTM-based E-D model is constructed as an auto-encoder for mapping the wind power (WP) time-series into a fixed-length representation, state of the trained E-D LSTM. Then, the representation concatenated with weather forecasting information is used as a new input to another multiple LSTM network to make WPP. Real data collected from a wind farm with capacity of 50 MW of Shan Xi province were used to verify the conclusions. Results illustrate that the proposed method improves the model generalization ability and lowers misspecification risk by utilizing the WP time relationship through auto-encoding (AE) process. Combining extracted representation with weather forecasting information further improves the prediction accuracy.

1. Introduction

It is widely accepted that the intermittent and random nature of wind energy leads to the uncertainty and weak controllability of WP [1]. This brings challenges to the safe and stable operation of the power grid. Accurate WPP can relieve pressure from grid frequency and peak regulation, which is of great significance for large-scale wind power grid integration and operation management.

At present, divided by forecast horizons, WPP method includes: long-term forecast over one year; medium-term forecast over one month or week; short-term forecast over one hour; and ultra-short-term forecast over minutes [2]. Meanwhile, WPP can also be divided into physical approaches and statistical approaches. The physical approaches mainly utilize weather factors such as wind speed, wind direction, atmospheric pressure, and temperature provided by numerical weather prediction (NWP). In combination with the topography and topographic information around the wind farm, local wind speed is predicted and WPP made [3]. Statistical methods mainly include time-series extrapolation methods such as auto-regression moving average (ARMA), exponential smoothing (EM), Kalman filter, as well as machine learning and neural networks such as support vector machine (SVM) and artificial neural network (ANN) [3-5]. However, the extrapolation method has strict assumptions on the data distribution [6]. The kernel function for SVM is selected arbitrarily, and the increase of data dimension will lead to computational complexity. Meanwhile, Shallow ANN has the disadvantages of over fitting and poor generalization [7].

With the success of dealing with temporal and sequential forecasting problem [8-9], more and more deep learning models are put into use in the field of energy forecast. Daniel M., Kasun A. and Milos
M. [11] compare the short-term electric load forecasting accuracy of standard LSTM using LSTM-based sequence to sequence (Seq2seq) architecture. Huiting Zheng, Jiabin Yuan, and Long Chen [12] show a generic framework that combines extreme gradient boosting, empirical mode decomposition and LSTM-based Seq2seq model to forecast short-term electric load. In the field of WPP, Qiaomu Zhu proposed a simple LSTM model for ultra-short-term WP [13].

In this work, we propose a long short-term memory (LSTM) based encoder-decoder (E-D) model for wind power prediction (WPP). Firstly, the LSTM-based E-D model is constructed as an auto-encoder for mapping the wind power (WP) time-series into a fixed-length representation, state of the trained E-D LSTM. Secondly, the representation concatenated with weather forecasting information is used as a new input to another multiple LSTM network to make WPP. Real data collected from a wind farm with capacity of 50 MW of Shan Xi province were used to verify the conclusions. Results illustrate that the proposed method improves the model generalization ability and lowers misspecification risk [17] by utilizing the WP time relationship during auto-encoding (AE) process. Combining extracted representation with weather forecasting information further improves the prediction accuracy.

2. LSTM-based Network

2.1. LSTM Units

By introducing input gate, forget gate and output gate, LSTM overcomes the problems of vanishing gradient, as shown in Fig1 [14]. The input gate defines how much of the newly computed state for the current input to let through. The forget gate defines how much of the previous state to let through. And the output gate defines how much of the internal state to expose to the external network.

Here, the equations (1) through (3) express a single LSTM cell’s three gates.

\[ i_t = \sigma_{sig}(W_i c_{t-1} + U_i x_t + b_i) \]  
\[ f_t = \sigma_{sig}(W_f c_{t-1} + U_f x_t + b_f) \]  
\[ o_t = \sigma_{sig}(W_o c_{t-1} + U_o x_t + b_o) \]

\[ c_t \] and \[ c_{t-1} \] determine the intermediate state \( c_t \) for LSTM at time step \( t \).
\[ \tilde{c}_t = \phi_{\text{tanh}}(W_c c_{t-1} + U_c x_t + b_c) \]  
(4)

\[ c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \]  
(5)

\[ h_t = o_t \odot \phi_{\text{tanh}}(c_t) \]  
(6)

where: \( \odot \) is element-wise product; \( W_i, W_f, W_o \) and \( W_c \) are the weight matrix for input gate, forget gate, output gate and candidate state; \( U_i, U_f, U_o \) and \( U_c \) are weights linked to input data; \( b_i, b_f, b_o \) and \( b_c \) are constants; \( \sigma_{\text{sig}} \) is sigmoid activation function and \( \phi_{\text{tanh}} \) is tanh activation function [14].

2.2. LSTM-based E-D Structure
In the LSTM-based E-D structure, shown in Fig2 [14], the encoder reads the input sequence and maps it into a fixed-length vector representation i.e. the candidate state \( c_t \). After that, the decoder will use \( c_t \) and the value predicted by the previous time step to forecast the next time step.

\[ X = \{x^1, x^2, ..., x^n\} \]

\( \tilde{c}_t \) is the intermediate state of the encoder at step \( t \), where \( c_t \in R^m \) and \( m \) is the number of neurons in the encoder [16]. The decoder decodes \( c_t \) into the target sequence \( Y = \{y^1, y^2, ..., y^n\} \).

3. Wind Power Prediction Model

3.1. Error Analysis
Let \( f^W(X) \) be the WPP model to build, \( W \) is the parameter set to be estimated. Given input \( x^* \) and output \( y^* \), we have:

\[ y^* = f^W(x^*) + \varepsilon \]  
(7)

The variance of the prediction can be further decomposed using law of total variance [17]:

\[ Var(y^*|x^*) = Var(f^W) + Var(\varepsilon) \]  
(8)
where: (1) $Var(f^W)$ refers to the model uncertainty; (2) $Var(\epsilon)$ is the noise level during data generating process. Considering into the effect due to covariance shift [18], $Var(\epsilon)$ can be further treated as the result of model misspecification and inherent noise [17].

3.2. LSTM-based E-D WPP Model

As is shown in Fig 3, the WPP model here uses the encoder-decoder architecture to perform auto-encoding (AE) thereby reducing the model’s misspecification error [17]. During AE [19] process, the fixed-length intermediate state $c_t$ is extracted as an abstract representation of the WP time relationship. In order to improve the training efficiency, target value equals input value with reverse order. Therefore, given the input value $\{x^1, x^2, ..., x^n\}$, the target value is $\{x^n, ..., x^2, x^1\}$. Here, auto-encoder uses one layer LSTM architecture with 32 neurons.

After the AE process, an embedding layer representing the features of WP time-series is obtained. Then combine this embedding layer with future weather data to form a new input and feed it into another LSTM network. The intuition is that change in weather patterns is the main reason for the volatility of WP. If weather time-series pattern remains unchanged, WP historical information has been already learned during the AE process. If there are new changes in weather patterns, the forecast model should consider both of this and features learned from AE process.

Here, we use another 3-layer LSTM network as a prediction model, where: input data has a dimension of 69 (including 64 dimensions of the embedded layer as well as wind speed, air temperature, air pressure, air density and wind direction) and neurons are set to 128, 64, 32.
3.3. Data Scaling

Min-max scaling method is used to scale the features of wind speed, air temperature, air pressure and air density down to a given range, as follows:

\[ x_{scaled} = 0.8 \times \frac{x - (x_{max} + x_{min})/2}{(x_{max} - x_{min})/2} \]  

(9)

this is because the proposed model uses hyperbolic tangent (tanh) function as the activation function for LSTM. The neurons of a layer will saturate if the value is too close to 1 or -1. So, we choose the scaled range of \([-0.8, 0.8]\).

For the wind direction, Sine function [13] is used to scale the data, as follows:

\[ x_{wd}_{scaled} = \sin(x_{wd}) \]  

(10)

4. Dataset and experimental results

The Model is applied to a wind farm with an installed capacity 50MW in Shanxi Province for training and testing data. The data ranges from January 1, 2016 to May 31st with interval 15min, a total of 14496 observations. Input data include WP, wind speed, air temperature, air pressure, air density at the height of hub and wind direction.

The data samples are constructed using a one-step sliding window, in which each sliding window contains the first 5 days of input data as historical data. The prediction period is 3, 6, 9, 12 hours in the future. Divide the dataset into train set, validation set and test set with a percentage of 50%, 30% and 20%. In the training process, the stochastic gradient descent with momentum (SGDM) is used to optimize the parameters of the model. Error function uses the root mean square error (RMSE) as a measure of the prediction level, as follows:

\[ R_{MSRE} = \frac{1}{Cap} \times \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left( \frac{\hat{y}_i - y_i}{y_i} \right)^2} \]  

(11)

where: Cap indicates the installed capacity of the wind farm; \( \hat{y}_i \) is the predicted interval value, and \( y_i \) is the predicted interval target value.

At the beginning of the training, the higher learning rate was maintained to improve the training efficiency. After the 500 rounds of training, learning rate is reduced at an exponential rate, and the overall training was 1500 rounds. In addition, an early-stop mechanism is set up in the training process. When the validation set error increased after 10 iterations, the training terminates.

In order to compare with the LSTM-based E-D model, a LSTM model without AE is used. The LSTM model without AE is set to have same number of parameters and layers with the proposed model.

Table 1. RMSE for 3-12 hour-ahead prediction %

| Prediction interval | Test data                |
|---------------------|--------------------------|
|                     | LSTM-based E-D | LSTM without AE |
| 3-hour              | 2.6              | 3.1              |
| 6-hour              | 5.2              | 7.1              |
| 9-hour              | 8.5              | 13.0             |
| 12-hour             | 11.8             | 18.8             |

Table 1 shows that:

1) In all prediction intervals, the prediction error of the LSTM model based on E-D was 2.6%, 5.2%, 8.5% and 11.8%, lower than that of the LSTM without AE. This is because the LSTM-based on E-D extracted the historical time relationship of WP through AE process, and took into account the new changes of meteorological laws in the prediction period, such as sudden changes of wind speed
and temperature, etc. Therefore, the prediction made by the LSTM model based on E-D is more sensitive to the WP mutation, as shown in Fig4, while the prediction curve of the LSTM model without AE is relatively lagged. In terms of the error distribution, the LSTM model without AE has a more dispersed error distribution, while the distribution of the proposed model is more concentrated, as shown in Fig5.

![Fig.4. WPP Performance of 2 Models (9 hour ahead)](image1)

![Fig.5. Error Distribution (9 hour ahead)](image2)

(2) With increase in prediction horizon, the prediction error of both models increased to some extent. From 3-hour to 12-hour period, the prediction error of LSTM-based on E-D model increased by 9.2%, less than the LSTM model without AE, 15.7%. The error changes were also calculated between 6 and 9 hours, 12 hours. For the LSTM model without AE, the prediction errors were 5.9% and 5.8%, while the LSTM-based E-D model only had 3.3% and 2.7%, which benefits from intermediate state extraction through AE process.
5. Conclusions
This work presents a LSTM-based E-D neural network for WP forecast. It shows that: (1) the LSTM-based E-D architecture as an auto-encoder can reduce the model’s misspecification risk by extracting the feature of the WP time relationship; (2) it will make the model more sensitive to mutations by including new external change factors. Experiment result demonstrates a better performance of the proposed model compared with the LSTM model with no AE pre-processing.

References
[1] Xiuyuan Y, Yang X, Shuyong C 2005 Proceeding of the CESS 25 1-5
[2] Jian W, Gangui Y, Wei S and Gang M 2011 Journal of Northeast Dian li University 31 20-24
[3] Giebel G 2003 Journal of Virology, 82 9513-24
[4] Pan Z, Yiping D, Junrong X and Yingxin S 2011 Journal of Xi’an Jiao Tong University 45 47-51
[5] Yusheng X, Chen Y, Junhua Z, Kang L, Xueqin L, Qiuwei W, Guangya Y 2015 Automation of Electric Power Systems 39 141-51
[6] Bin M 2016 Study on Combination Forecasting for Short-term Wind Power based on Artificial Intelligence (Chengdu: Southwest Jiaotong University)
[7] Jie Y 2016 Uncertainty of Wind Power Forecasting and Power System Economic Dispatch (Beijing: North China Electric Power University)
[8] Graves A, Jaitly N and Mohamed, A R 2013 Automatic Speech Recognition and Understanding (ASRU) 2013 IEEE Workshop on. IEEE 273-278
[9] Fan X, Hua C, Yiquan F 2017 Journal of East China University of Science and Technology (Natural Science Edition) 43 553-558
[10] Hochreiter S and Schmidhuber J 1997 Neural Computation 9 1735-80
[11] Daniel L M, Kasun A and Milos M 2016 Int. Conf. Industrial Electronics Society (IECON) 7046-51
[12] HuitingZ, Jiabin Y and Long C 2017 Energies 10 1168-86
[13] Qiaomu Z, Hongyi L, Ziqi W, Jinfu C and Bo W 2017 Power System Technology 41 3797-3802
[14] Goodfellow I, Bengio Y, Courville A 2016 Deep learning (Cambridge, Massachusetts: MIT Press)
[15] Sutskever I, Vinyals O and Le Q V 2014 Advances in neural information processing systems 4 3104-12
[16] Malhotra P, Ramakrishnan A, Anand G, Vig L, Agarwal P and Shroff G 2016 arXiv preprint arXiv: 1607.00148
[17] Lingxue Z, Laptev N 2017 Int. Conf. on Data Mining Workshops (ICDMW) 103-110
[18] Shimodair H 2000 Journal of Statistical Planning and Inference 90 227-244
[19] Srivastava N, Mansimov E, Salakhutdinov R 2015 Int. Conf. on machine learning 843-852