Short-term wind power forecasting model based on multi-feature extraction and CNN-LSTM

Honghai Kuang¹*, Qian Guo¹, Shengqing Li¹ and Hao Zhong²

¹ School of Electrical and Information Engineering, Hunan University of Technology, Zhuzhou, Hunan, 412007, China
²Hubei Key Laboratory of Cascaded Hydropower Stations Operation & Control, China Three Gorges University, Yichang, Hubei, 443002, China

*Corresponding author’s e-mail: kuanghonghai_1114@hut.edu.cn

Abstract. Improving the accuracy of short-term wind power forecasting is critical to wind power consumption. This paper establishes a short-term wind power prediction model based on the multi-feature extraction and deep learning network CNN-LSTM. Muti-features are extracted from original data to improve the accuracy of training. In addition, clustering algorithm is used to classify training data and train the models corresponding to those classes. CNN-LSTM prediction models are established for each cluster and compared with ARIMA, RNN, CNN and LSTM models.

1. Introduction

Affected by natural factors and random factors, the output power of wind power has a strong degree of non-linearity, which makes it more difficult for the grid to dispatch wind energy efficiently. To ensure the quality of power supply, the dispatching method of wind farms is conservative wind curtailment, which results in a high rate of wind curtailment. The accuracy of short-term wind power forecasting directly affects the utilization rate of wind energy.

Short-term wind power forecasting methods include physical methods, statistical methods and deep learning methods. The statistical method realizes the wind power prediction through regression strategy like autoregressive integrated moving average model (ARIMA). The deep learning method has a deeper network, which enables it to build more complex nonlinear relationship models. Deep learning methods include recurrent neural network (RNN), convolutional neural network (CNN), and long short-term memory (LSTM) [1]. Literature [2] used the LSTM model to predict wind power, and compared it with some machine learning models. LSTM showed the advantages of its unique structure for time series.

2. Data and feature pre-extract

2.1. Wind farm data

The wind farm data is from Xinjiang Province in China in 2017. The installed capacity of the wind farm is 49.5MW, and the historical data has a resolution of 15 minutes, including the historical wind speed, wind direction, and actual power generation of the wind turbine. The NWP data includes historical wind speed and direction at heights of 10m, 30m, 50m, and 70m, as well as temperature, humidity, and pressure data. There are 11 variables in total, and each variable has 96 data per day.
All variables need to be normalized: the cosine and sine of the wind direction angle are calculated as the normalized form of the wind direction, and other variables are calculated by the maximum-minimum normalization method. Therefore, there are 15 NWP variables for each day now. And the test set accounts for 10% of the original data, which is used to detect prediction results, and the training set accounts for 90%, which is used to train the model.

2.2. Features extraction

Because wind power is affected by many factors, it is difficult to ensure the predictive ability with only one training model. Therefore, extract the basic and statistical features of the data before training, and cluster the multi-features into some classes for which corresponding prediction models are established. When predicting the wind power of a day, NWP data of the day can determine the class that day belongs to, and predict with the corresponding CNN-LSTM model.

According to 15 NWP variables, 19 basic features and 150 statistical features are extracted. The features of one day form a 169-dimensional vector. Basic features are shown in Table 1.

| Variables | Description |
|-----------|-------------|
| $v_{i,b,1}$ | Minimum wind speed at height $b$ meters of day $i$ ($b = 10, 30, 50, 70$) |
| $v_{i,b,2}$ | Maximum wind speed at height $b$ meters of day $i$ ($b = 10, 30, 50, 70$) |
| $d_{i,b,1}$ | The mean value of the sine of the wind direction angle at height $b$ meters of day $i$ ($b = 10, 30, 50, 70$) |
| $d_{i,b,2}$ | The average value of the cosine of the wind direction angle at height $b$ meters of day $i$ ($b = 10, 30, 50, 70$) |
| $t^i$ | The average temperature of day $i$ |
| $s^i$ | The average humidity of day $i$ |
| $h^i$ | The average pressure of day $i$ |

The other 150 features are obtained from 10 statistical formulas of 15 NWP variables. Statistical features include trend factor ($K^j$), sequence correlation ($C^j$), sequence nonlinearity ($G^j$), Skewness ($U^j$), Kurtosis ($F^j$), long-term dependence ($O^j$) and Chaos ($Z^j$).

Generally, the time series $L^j_i$ of the NWP variable $j$ on the $i$-th day can be standardized by Box-Cox transformation is denoted as $L^j_i$. According to the Loess method, $L^j_i$ is decomposed into a trending component $Q^j_i$ and a non-trending component $E^j_i$. The trend factor $K^j_i$ of the NWP variable is calculated by equation (2):

$$K^j_i = 1 - \frac{Var(E^j_i)}{Var(L^j_i)}$$

where $Var(\cdot)$ represents the variance of a set of data.

Sequence correlation is a key feature of non-trend data. The Box-Pierce method is used to quantify the autocorrelation of non-trend data. The calculation formula for the serial correlation of the non-trending component $E^j_i$ is as follows:

$$C^j = N \sum_{m=1}^{M} r_m^2$$

where, $N = 96$ is the serial length of the NWP variable $j$ on the $i$-th day; $r_m$ is the autocorrelation function of the $m$-th timestamp in a day; $M = 20$ is the largest considered timestamp.

Most of the series are nonlinear. This paper uses the two-test regression method of Terasvirta’s neural network test to detect the nonlinearity of $Q^j$ and $E^j$. Suppose $Q^j$ of a certain day is expressed as $A^j = [A_{j,1}, ..., A_{j,t}, ..., A_{j,N}]$. Calculate the regression of $A_{j,t}$ at $1, A_{j,t-1}, ..., A_{j,t-N}$ to get the residual $\tilde{v}_t^j$, and the sum of squares of the parameter $SSR_t = \Sigma \tilde{v}_t^j$. The number of auxiliary quantities used during the
period is $\alpha = 2$, namely $A_{j-t}^2$ and $A_{j-t}^1$. Then the nonlinearity of $Q^j_i$ and $E^j_i$ is expressed by formulas (5) and (6), where $SSR'$ is calculated from the $Q^j_i$, and $SSR''$ is calculated from the $E^j_i$.

$$G_{j-t}^{1} = \frac{SSR_{i}^{j} - SSR_{i}^{1}}{SSR_{i}^{1}} \times \frac{N - I - 1 - \alpha}{\alpha}$$  \hspace{1cm} (4)$$

$$G_{j-t}^{2} = \frac{SSR_{i}^{j} - SSR_{i}^{2}}{SSR_{i}^{2}} \times \frac{N - I - 1 - \alpha}{\alpha}$$  \hspace{1cm} (5)$$

Skewness characterizes the degree of data distortion or asymmetry. When the Skewness is greater than 0, there are more extreme values at the right end of the data.

$$U_{i}^{j} = \sum_{i=1}^{N}\left(q_{i}^{j} - \bar{Q}_{i}^{j}\right)^{3}/N(\sigma_{j}^{i})^{3}$$  \hspace{1cm} (6)$$

$$U_{i}^{j} = \sum_{i=1}^{N}\left(e_{i}^{j} - \bar{E}_{i}^{j}\right)^{3}/N(\sigma_{j}^{i})^{3}$$  \hspace{1cm} (7)$$

where $N = 96$; $\bar{Q}_{i}^{j}$ is the average of 96 values in $Q^j_i$; $q_{i}^{j}$ is the value of the $t$-th time stamp in $Q^j_i$; $\sigma_{j}^{i}$ is the standard deviation of $Q^j_i$.

Kurtosis represents the features’ number with high and low peaks in the probability density distribution at the average position. The formula is as follows:

$$F_{i}^{j} = \sum_{i=1}^{N}\left(q_{i}^{j} - \bar{Q}_{i}^{j}\right)^{4}/N(\sigma_{j}^{i})^{4}$$  \hspace{1cm} (8)$$

$$F_{i}^{j} = \sum_{i=1}^{N}\left(e_{i}^{j} - \bar{E}_{i}^{j}\right)^{4}/N(\sigma_{j}^{i})^{4}$$  \hspace{1cm} (9)$$

The change process of the time series reflects the dependence of the data in the time dimension. In this article R/S analysis is used for estimating this long-term dependence [3]. First, split $L_{i}$ into $n_0$ sub-serials $\varphi_{j,n}(n = 1, 2, \ldots, n_0)$ with the same length $h_0$. Then, standardize each sub-sequence to get $\varphi_{j,n}^*$, and build a cumulative time series $\psi_{j,n} = \sum_{i=1}^{n_0} \varphi_{j,n}^*(h = 1, 2, \ldots, h_0)$. Finally, calculate the range $R_{j,n} = \max(\psi_{j,n}) - \min(\psi_{j,n})$, and the ratio to the standard deviation is $R_{j,n}/S_{j,n}$. The average of $R_{j,n}/S_{j,n}$ of the sub-serial is used to characterize the long-term dependence, calculated as follows:

$$Q_{j} = \sum_{n=1}^{n_0} \frac{R_{j,n}}{S_{j,n}}$$  \hspace{1cm} (10)$$

In the initial stage, the small errors that affect the wind power forecast may accumulate to a non-negligible degree over time. The Lyapunov exponent $\lambda$ can be used to measure whether this complex system will eventually be classified as Chaos. The chaotic calculation method refers to the robust method proposed by Rosenstein et al. [4].

$$Z_{j} = \lambda$$  \hspace{1cm} (11)$$

The value range of each feature is different, and the logistic transformation is required to ensure that the value range of each feature variable is $[0, 1]$.

All 169 features can be assembled into a vector expression, where $j = 1, \ldots, 15$ in equation (12).

$$\Phi_{j} = (K_{j}^{1}, C_{j}^{1}, G_{j}^{1}, G_{j}^{2}, U_{j}^{1}, U_{j}^{2}, F_{j}^{1}, F_{j}^{2}, Q_{j}, Z_{j}, v_{j}^{b,1}, v_{j}^{b,2}, d_{j}^{i,1}, d_{j}^{i,2}, T^{i}, T^{j}, \bar{H})$$  \hspace{1cm} (12)$$

It is difficult to fully learn the high-dimensional features by training only one prediction model, so the clustering algorithm is used in this paper to distinguish features. It is considered to classify data through clustering, and to train predictive models for each clustering category. To ensure the accuracy of prediction on the basis of retaining multi-dimensional features. For the convenience of clustering, dimensional reduction is required in advance.
2.3. Feature reduction processing
T-distributed stochastic neighbor embedding (t-SNE) can extract most of the local structure of high-dimensional data. The application of t-SNE has shown good results with many real data, so it is used for dimensional reduction in this paper.

$t$-SNE uses conditional probability to express the similarity between data. If the similarity between the feature quantities $\Phi_1$ and $\Phi_2$ is expressed by the probability $p_{2|1}$, and the probability is proportional to the similarity. The conditional probability $q_{2|1}$ between the two points is equal to $p_{2|1}$, if $\Phi_1'$ and $\Phi_2'$ are low-dimensional mappings of feature quantities, and $\Phi_1'$ and $\Phi_2'$ can express high-dimensional $\Phi_1$ and $\Phi_2$. Therefore, by minimizing the cost function obtained by Kullback-Leibler divergence, a low-dimensional mapping of the feature quantity can be obtained, which is implemented by the gradient descent method. The cost function is as follows:

$$C = KL(P \parallel Q) = \sum_i \sum_j p_{ij} \log(p_{ij} / q_{ij})$$ (13)

In addition, the perplexity is a tunable hyper-parameter of $t$-SNE, and the recommended range in the literature [5] is [5, 50], here it is 42.

2.4. K-means clustering

K-means clustering can divide the data set into several groups according to the distance between the data when there is no prior knowledge. The K-means algorithm is used to cluster the daily feature vectors. The number of clusters $k$ is determined by the elbow method. Figure 1 shows that $k = 3$ is the most appropriate. The clustering result is shown in Figure 2.

3. Wind power forecasting model

3.1. CNN-LSTM wind power prediction model

In this paper, CNN is used as a feature extractor to prevent gradient dispersion when LSTM predicts multi-dimensional sequences. LSTM is used as a wind power predictor, which is conducive to the retention of long-term sequence information. CNN is used for feature extraction of wind power feature vectors. The results are inputs to LSTM, and the output is the future 24 hours wind power, the structure is shown in Figure 3.

In this paper, batch normalization (BN) [5] is used to keep the input distribution in the multi-layer network basically unchanged to prevent convergence difficulties. The CNN part contains 3 one-
dimensional convolutional layers, 2 BN layers and 3 pooling layers. In the activation function, SELU has better convergence than ReLU, which can avoid the problem of gradient disappearance between training [6]. The LSTM part is composed of a fully connected layer and a sigmoid activation function to realize the final prediction and output of wind power.

The process of the CNN-LSTM wind power short-term prediction model in this article is shown in Figure 4.

![Figure 4: The framework of prediction system](image)

### 3.2. Assessment criteria

The indicators for evaluating wind power forecasting models are mean absolute error (MAE) and root mean square error (RMSE), which are as follows:

\[
\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |\hat{x}_i - x_i| / N
\]

\[
\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\hat{x}_i - x_i)^2 / N} \tag{15}
\]

where \(x_i\) is the actual value of wind power; \(\hat{x}_i\) is the predicted value of wind power.

### 3.3. Results & Discussion

Table 2 shows the average error of the wind power models based on ARIMA, RNN, CNN, LSTM and CNN-LSTM in the case of clustering and non-clustering.

| Prediction models | MAE/MW | RMSE/MW |
|-------------------|--------|---------|
|                   | No clustering | Clustering | No clustering | Clustering |
| ARIMA             | 4.301 2 | 4.156 0 | 5.582 3 | 5.434 1 |
| RNN               | 4.134 5 | 3.907 4 | 5.150 7 | 4.931 4 |
| CNN               | 4.263 5 | 4.039 6 | 5.372 4 | 5.099 8 |
| LSTM              | 4.103 6 | 3.806 3 | 5.126 2 | 4.907 8 |
| CNN-LSTM          | 3.898 4 | 3.700 0 | 5.003 1 | 4.788 3 |

The MAE and RMSE of the five models after clustering are reduced by 4.2% to 7.2% compared with no clustering. In conclusion, the effect of ARIMA and RNN is not pleasing. Compared with CNN and LSTM, the clustered CNN-LSTM has lower prediction errors, MAE is reduced by 8.40% and 2.28%, and RMSE is reduced by 3.76% and 2.43%, respectively. It can be seen that multi-feature extraction and clustering are beneficial to training, and the CNN-LSTM wind power short-term prediction model can better capture non-linear features with higher accuracy.

### 4. Conclusions

The new short-term wind power power prediction model proposed by combining deep learning network and multi-dimensional feature extraction can improve the prediction accuracy of deep learning network for wind power sequence. It can be seen that multi-dimensional feature extraction and clustering distinguish features more effectively, and the CNN-LSTM model has better performance than other models.
Acknowledgments
The present study was co-funded by national natural science foundation of China (51977072), natural science foundation of Hunan Province (2018JJ4076) and key laboratory open foundation of Hubei Province (2019KJX06).

References
[1] Sun, Y., Wang, P., Zhai, S., et al. (2020) Ultra short-term probability prediction of wind power based on LSTM network and condition normal distribution. Wind Energy, 23: 63-76.
[2] Qu, X.Y., Kang, X.N., Zhang, C., et al. (2016) Short-term prediction of wind power based on deep long short-term memory. In: IEEE PES Asia-Pacific Power Energy Engineering Conference. Xi’an. pp. 1148-1152.
[3] Saleh, A.E., Moustafa, M.S., Abo-Al-Ez, K.M., et al. (2016) A hybrid neuro-fuzzy power prediction system for wind energy generation. International Journal of Electrical Power & Energy Systems, 74: 384-395.
[4] Rosenstein, M.T., Collins, J.J., De Luca, C.J. (1993) A practical method for calculating largest Lyapunov exponents from small data sets. Physica D: Nonlinear Phenomena, 65: 117-134.
[5] Maaten L., Hinton G. (2008) Visualizing data using t-SNE. Journal of Machine Learning Research, 9: 2579-2605.
[6] Klambauer, G., Unterthiner, T., Mayr, A., et al. (2017) Self-normalizing neural networks. Advances in Neural Information Processing Systems, 30: 971-980.