An accurate detection method for turbine icing issues using LSTM network

Minglu Li¹ and Xin He²*
¹ China-EU Institute for Clean and Renewable Energy, Huazhong University of Science and Technology, Wuhan, Hubei, 430074, China
² School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, Hubei, 430074, China
* xinh@hust.edu.cn

Abstract. As one of the most widely applied clean energy resources, wind power generation is helpful for energy structure adjustment and sustainable development, whereas blade-icing failure affects the stability and benefits of wind farm. In consideration of efficiency improvement and safety hazard control, this paper proposed a failure detection model based on long short-term memory network. With the combination of turbine operation mechanism and data-driven method, the early warning of blade icing was implemented, which made it possible to take timely measures such as turning on the de-icing system at early moments of icing issues. The operational data collected by SCADA system were used as input for model, and progressive trend as well as hidden messages were learned from historical correlation at both temporal and characteristic scales. Consequently, the model achieved the fault detection and performed well on test dataset.

1. Introduction
With the exhaustion of fossil fuels and the increasing demand of environmentally friendly energy resources, renewable energy gradually becomes a nonnegligible part of energy structure. Wind energy is widely concerned and applied due to its cleanness and enrichment. There have been lots of wind turbine generators installed in the region of mountains and coastal areas with humid and cold climate, leading to the icing events on wind turbine blade. Blade icing has been a worldwide problem in wind electric field. As a critical component capturing the wind energy and converting it into kinetic energy, blade should avoid being frozen to maintain the load performance and balance performance. If no measures are taken in time, load of wind turbine cannot satisfy the requirements of grid connection. Meanwhile, turbine imbalance influences the generating capacity, severe imbalance even damages the weak parts of turbine. Although many new types of wind turbines are equipped with automatic de-icing system, challenges occur in practical application when nascent moment of icing process cannot be predicted precisely to trigger the de-icing system as soon as possible. Hence, studying the wind turbine blade icing detection technology deserves much effort due to its significance to turbine safety.

Most wind turbines are equipped with SCADA, the supervisory control and data acquisition system, which measures and stores considerable data reflecting the real-time running status. SCADA system collects many variables, but existing methods for blade icing detection mainly focus on monitoring the bias between the actual power and theoretical power. Once the bias reaches a certain value the alarm will be triggered to shut down the turbine. However, the alarm is often triggered after large area icing has happened, and running on such condition will increase the risk of blade damaging. If severe failure
can be diagnosed by mining the big data collected from SCADA system, then an active maintenance way can replace the previous passive one to implement an effective improvement on usage and reduction on maintenance cost of wind power equipment.

There are some research about wind turbine icing, for instance, Feng et al [1] collected the parameters of blade side, and combined the methods of grey theory model and LEWICE to predict the shape of blade icing. Dierer et al [2] used WRF model to simulate various parameters, then applied an accretion model to predict the turbine icing, this research also pointed a 6.5% loss difference in power production when turbine worked with and without blade heating. Yang et al [3] proposed a condition monitoring method based on the study of correlations among relevant SCADA data, which was beneficial for detecting incipient turbine blade and drive train faults.

2. Dataset and methods

2.1. Experimental dataset description

The dataset we experimented on in this paper pertains to the SCADA real-time data. SCADA is a critical system to manage, monitor and control the wind turbine. SCADA system produces huge amount of data each day, and the sampling interval varies from 3 seconds to 11 seconds when turbines are running normally, meanwhile large-scale data missing occurs when turbine stops working in case of failure or human factor. Three different turbines are involved in this dataset, and for each turbine, the data collection quantity varies from 100,000 to 400,000, corresponding to the time duration from one month to two months [4]. As for the feature dimension of dataset, there are hundreds of features in SCADA system generally, but presented dataset only retains 26 continuous numerical features after screening. The reserved features concerning the working condition parameters, environmental parameters and status parameters are shown below.

| Parameter type     | Specific descriptions                                         |
|--------------------|--------------------------------------------------------------|
| Environment        | Wind speed, wind direction, ambient temperature              |
| Operating condition| Generator speed, power, yaw position, yaw speed, DC current  |
| Status             | Pitch angles, pitch speed, cabin temperature, variable-pitch motor temperature, acceleration in x and y directions |

2.2. Data pre-processing

2.2.1. Subsets division. The applied model on icing fault detection is mainly based on LSTM neural network. To obtain enough information of relevance between sequential samples, the data continuity within a certain range of time should be ensured. Thus, in data pre-processing, we detected the time break and divided the samples into multiple continuous subsets, avoiding feature trend with time break being learned mistakenly by LSTM layers. Firstly, we accessed some prior knowledge about time that training data owned uniquely and implemented the subsets segmentation. The next task was to construct an indicator related to time interval in test set. Considering that temperature usually changes steadily over time, we chose temperature of turbine cabin and checked that its difference value had close relationship with real time interval.

2.2.2. Sliding window and voting system. Besides time indicator, sliding window and accumulating vote were also proposed to adapt to the timestep of LSTM and realize the reification of data correlation. Sliding window can attain multistage window data with fixed length from a continuous subset and feed these isometric samples into LSTM layer to learn the variation tendency both at characteristic and temporal scales. As a result of sliding window segmentation, most samples belong to different LSTM inputs simultaneously, which means that a sample can be predicted in different sliding window data and gets multiple predicted values. In this context, it is desirable to integrate overlapping labels by using
accumulating vote, which produces more reliability of sample prediction. Timestep equals to the length of sliding window and reflects the length of smooth variability and similar working status of samples over a period. To some extent, the length of sliding window represents the length of series data with same label, so the length should be moderate because too long window may include different labels and too short window would make model learn the noise in data. Otherwise, the sliding step also spreads the risk of samples with different status in a window, giving the model more flexibility in prediction. Here we chose the length of window as 50 and the step value as 10, so the prediction of five votes represents a credible result of icing issues. In case that continuous subset has a length not divisible by 10, duplicate sliding window sampling will occur at the end of subset, resulting in votes above five. These votes are also treated as five votes. In a traditional way, the output from a LSTM layer should be converted to label by a threshold classifier. Through the sigmoid function, the final output gets a probability value ranging from zero to one, so output larger than the threshold 0.5 is converted to one, whereas zero. Then these outputs from different windows were accumulated as final decision. However, later experiments proved that ignoring the threshold classification and accumulating the probability values directly improved the model performance.

2.2.3. Sample weight and feature selection. Considering the real application of accurate prediction for early icing moments as well as the classification imbalance problem caused by normal samples that are far more than fault samples, we selected the method of setting sample weight and applying the weight into cost function. Basically, the model trained by overfull normal samples could lead to the fault omission, so the weight of icing samples were set higher than the weight of normal samples. Furthermore, to be close to the practical problem of accurate warning, the model ought to be sensitive to the start and end moment of icing case. Thus, a crest curve of weight value was prepared for the interval near icing start and end points, and the start and end points have higher weight than the points in middle of ice period. After preliminary experiments, it seems that variance between train set and test set predictions is not negligible. According to the statistical parameter values of all features and data distribution of each dimension for all turbines, we select ten features by distributional similarity as input features. In the same way, similarity also determines which turbine has higher weight in trainset, then the turbine weight overlays on the sample weight setting for cost function.

2.3. LSTM model
Theoretically, recurrent neural network is capable to learn the long-term dependencies, but in reality recurrent neural network cannot deal with long historical information, causing the problem of gradient explosion and gradient vanishing. Thus LSTM, the long short-term memory neural network rises in response to the demand. LSTM utilizes the significant information captured from historical learning process to assist with current prediction task [5]. Compared to RNN, LSTM is able to delete or save the information of neuron status, making it possible to make a certain choice from a continuous stream of messages passing through the neuron structure. The selection mechanism is achieved by the combination of several gate structures. One LSTM unit has three such gates, input gate, forget gate and output gate, to implement the functions of message selection and control. After the matrix multiplication and addition with bias, the data flow goes out from sigmoid layer to be an activation value between zero and one. Activation value decides how much knowledge will be absorbed by neuron, for instance, output one means keep all information, conversely output zero means forgetting all the transmitted messages.

Input gate acts on the input \(x_t\) and previous output \(h_{t-1}\) to memorize selectively. After generating the candidate state \(C_t\), input gate decides the activation value. Thus, the product of activation value and candidate state value composes part of the new state of neuron.

\[
i_t = \sigma(W_i * x_t + U_i * h_{t-1} + b_i) \quad (1)
\]

\[
C_t = \tanh(W_c * x_t + U_c * h_{t-1} + b_c) \quad (2)
\]

Forget gate, on the other hand, decides on the part of deletion. In the process of continuous learning, only important information is necessary to retain and deliver. Forget gate selects the part for deletion from cell state at previous moment.
\[ f_t = \sigma(W_f * x_t + U_f * h_{t-1} + b_f) \]  \hspace{1cm} (3)

Input gate and forget gate cooperate to determine the update content of memory cell state, one for new information learned from candidate state, another for old knowledge kept from last cell state.

\[ C_t = i_t * \tilde{C}_t + f_t * C_{t-1} \]  \hspace{1cm} (4)

Output gate provides an activation value to multiple with the new cell state normalized by tanh function. Product of these two forms the output of memory cell \[ o_t = \sigma(W_o * x_t + U_o * h_{t-1} + b_o) \]  \hspace{1cm} (5)

\[ h_t = o_t * \tanh(C_t) \]  \hspace{1cm} (6)

3. Results and conclusion

3.1. Model evaluation

Since there exists class imbalance in experimental dataset, mainly reflected in the large amounts of normal samples and small amounts of failure samples, if classification accuracy is chosen to be the only criteria of model evaluation, the accuracy could be good even though all samples are predicted to be in normal working condition. Thereby, the respective precision and recall of normal samples and failure samples are included in parameters for model evaluation. Precision value can reflect the false alarm rate to a certain extent, and a higher precision value means less false alarms. Similarly another recall value reflects the missing alarm rate, higher recall value indicates less missing alarms.

3.2. Experimental results

The prediction result on test dataset is shown below in table:

| Prediction result | Precise | Recall |
|-------------------|---------|--------|
| Normal samples    | 0.9832  | 0.9894 |
| Failure samples   | 0.8121  | 0.7305 |
| Classification accuracy | 0.9742 |

According to certain rules, the prediction results were converted to multiple entire intervals representing the blade icing issues. There are two main rules to modify the results, one is that only continuous interval with five votes is determined to be a real icing period, and the start and end point should be the band edge with one vote. Another rule is that if interval between two icing predictions does not exceed 100 points of sample, then the two icing predictions should be concluded as one.
As is shown in figure, the prediction of end point is much better than start point. Besides, there exists alarm delay of the start point for some icing predictions. The reason for this phenomenon is that the annotation for start point in train dataset is not so clear. Since the icing periods always begin with an unlabeled period, the real start point of icing period cannot be learned by model. The uncertainty of information also reflects in the results of test dataset.

3.3. Conclusion
Wind power generation has great prospects for development. Therefore, to solve the common problem of blade icing fault happening in cold and humid areas, a fault diagnose model was proposed. Multiple pre-processing methods including sliding window and voting system were used to convert raw data to a better data form fitting LSTM network for sequential relationship learning, likewise, the slight specificity of different types of turbine was considered. Serving the requirements of sensitive alarm at icing issues rather than general omission, we made improvement based on traditional icing prediction methods and verified the model validity on test dataset of turbine.

Acknowledgements
This research was supported by Innovation Platform of Industrial Big Data Industry and the first author would like to thank Prof. Ye Yuan for supervision.

References
[1] Feng, C., & Papachristou, C. (2013). Grey-model based ice prediction sensor system on wind turbine system.
[2] Dierer, S., Oechslin, R., & Cattin, R. (2011). Wind turbines in icing conditions: performance and prediction. Advances in Science and Research, 6(1), 245–250.
[3] Yang, Wenxian, Richard, Jiang, & Jiesheng. (2013). Wind turbine condition monitoring by the approach of scada data analysis. Renewable Energy, 53(9), 365-376.
[4] Innovation Platform of Industrial Big Data Industry. (2017). http://www.industrial-bigdata.com/?from=singlemessage&isappinstalled=0
[5] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
[6] Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. Neural computation, 12(10), 2451-2471.
[7] Graves, A. (2008). Supervised sequence labelling with recurrent neural networks. Studies in Computational Intelligence, 385.
[8] Bastien, Frédéric, Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I., & Bergeron, A., et al. (2012). Theano: new features and speed improvements. Computer Science.