A Method to Detect Remote Alarm Based on Edge Computing in Wind Energy

Li-fang GAO\(^1\), Jing WANG\(^1\), Di LIU\(^2\), Ze-san LIU\(^2\), Yang-yang LIAN\(^1\), Qi-meng Li\(^1\) and Ting-shun Li\(^3\)\(^*\)

\(^{1}\)State Grid Hebei Information & Telecommunication Branch, Shijiazhuang, China
\(^{2}\)State grid information & Telecommunication Group Co. Ltd., China
\(^{3}\)School of Control and Computer Engineering, North China Electric Power University, China

\(^*\)Corresponding author

Keywords: Blade icing detection, Machine learning, Edge computing, Time delay.

Abstract. Wind parks are usually located in places where few people live and is far from control center. Now, wind turbines often are embedded in so many sensors that operators can obtain their performance. But there are some important alarms difficultly to be monitored directly by sensors, such as blade icing which wind turbines in cold regions often encounter. These alarms involve many problems, such as energy losses, mechanical drawbacks and security. These complex alarms usually attribute to combination of many factors such as low temperature, wind speed, blade accelerator. Many scientists and researchers have proposed different methods or created new sensors to detect complex alarms. But these methods or facilities have different drawbacks that can cause low prediction accuracy, even high failure or error rate etc. This paper, taking blade icing for example, provides a practical method to detect remote alarm using monitoring data of wind turbine by machine learning algorithm without any new equipment. This method consists of four processes: preprocessing monitoring data, main features extraction, remote warning model construction, and deploying every sub-model as edge computing node. Firstly, getting massive historic monitoring data of wind turbine in wind park, and classifying; secondly, extracting main trait factors associating with blade icing; and then constructing detection model for each type wind turbine separately via machine learning according to massive historic monitoring data; finally, building edge computing nodes for every wind turbine to predict blade icing with model. This method has many advantages, such as low cost, high accuracy, clearly targeting, and early icing warning. Furtherly, with the help of edge computing, the generalization of machine learning model is improved. This study was completed with practical monitoring data of wind turbines in some park. Results show that the way to detect blade icing via machine learning is feasible and accuracy.

Introduction

Nowadays, wind power is one of most important renewable energy. Wind generation is an important part of China’s new energy industry, which was the third largest power by 6.1% in 2018[1]. But the places where the wind power turbines almost locate are abominable high latitudes regions and far from control center. In order that operators in control center know performance of wind turbines instantly, many sensors are embedded in wind turbines, just like wind turbines having many kinds of perception. Some running parameters of wind turbine can be obtained directly by sensors, others cannot, especially some complex alarms such as blade icing.

Ice accretes on the blades of wind power turbine in winter. Ice accretion on blades involves many problems, such as wind turbine energy losses, performance failures and security [2]. What’s more, most locations of wind park are far from control center, and it’s very difficult for operators to know whether blade is icing or not. So many scientists and researchers have been studying on how to perceive and deice remote wind turbine icing instantly for recent years, and propose different method to detect blade icing [3-5]. Those methods have different drawbacks, such as low accuracy, high cost, less feasibility.
Recently, with the development of Big Data technology and deep machine learning, some researchers try to predict remote blade icing only by data. Liu designs a model to detect blade icing using SCADA (supervisory control and data acquisition) data [6]. The model has many advantages, such as less cost, high accuracy, even having early icing warning ability. However, the model obtained by machine learning has poor generalization. If the sample data come from many different wind turbines, it is very difficult to get an ideal model. On the other hand, if the sample is gained from several similar wind turbines which are located the same area and have same type, it is easy to acquire a perfect model which is overfit and is difficultly deployed on others wind turbines. So, it seems contradictory for the model acquired by machine learning between accuracy and generalization. For improving the generalization of the model, this paper, taking blade icing for example, proposes a method to construct a model cluster which includes different sub-models built for different type wind turbines and these sub-models are gained separately from the massive data of different type wind turbines via deep machine learning, and then are deployed separately wind turbines to predict remote alarm as edge computing node. If so, to some extent, the contradictory between accuracy and generalization is relieved and balanced.

Preliminaries and Related Work

Remote Alarm - Wind Turbine Blade Icing

We know that most of wind parks are in odious places with rare men living and far from control center. Operators in control center can instantly acquire states of wind turbines at the help of sensors that are installed or embedded in wind turbines, and remote-control wind turbine runs. So, it is a big problem for operators how to know instantly whether wind turbine blade is icing, according for blade icing involving so many problems.

Now, more and more attentions of scientists and researchers are being attracted to detect and deice blade icing. Among these studies, the research of Liu Yao is the most fantastic, which predict blade icing by a data model gained from SCADA data set via sharp deep machine learning [6].

A typical SCADA monitoring different items of wind turbine by sensors are shown in Table 1.

| SN | Monitoring point | SN | Monitoring point | SN | Monitoring point | SN | Monitoring point |
|----|-----------------|----|-----------------|----|-----------------|----|-----------------|
| 1  | wind_speed      | 8  | pitch1_angle    | 15 | pitch2_moto_tmp | 22 | pitch2_ng5_tmp  |
| 2  | generator_speed | 9  | pitch2_angle    | 16 | pitch3_moto_tmp | 23 | pitch3_ng5_tmp  |
| 3  | Power           | 10 | pitch3_angle    | 17 | acc_x           | 24 | pitch1_ng5_DC   |
| 4  | wind_direction  | 11 | pitch1_speed    | 18 | acc_y           | 25 | pitch2_ng5_DC   |
| 5  | wind_direction_mean | 12 | pitch2_speed    | 19 | environment_tmp | 26 | pitch3_ng5_DC   |
| 6  | yaw_position    | 13 | pitch3_speed    | 20 | int_tmp         |     |                 |
| 7  | yaw_speed       | 14 | pitch1_moto_tmp | 21 | pitch1_ng5_tmp  |     |                 |

We wish that operators would gain the icing state of remote wind turbine instantly from real-time monitoring data by some model which is obtained by deep machine learning from historic monitoring data.

Machine Learning

Machine-learning research is to study and apply the computer modeling of learning processes in their multiple manifestations, which facilitate the development of intelligent system [7]. Machine learning is more and more fashionable and has fast growing effects in many areas of pattern recognition, computer vision, speech recognition and so on [8].

Whether blade icing or not is binary classification. Machine learning algorithms are known to effectively classify complex datasets, including binary and multi-class datasets. These usual algorithms are Classification and Regression Trees, k-Nearest Neighbor, Support Vector Machines
and Naive Bayes [9]. LR (Logical Regression) is a well-known binary classification algorithm [10]. We can use LR algorithm to classify iced data and non-ice data.

**Edge Computing**

With the proliferation of Internet of things (IoT) and the burgeoning of 4G/5G network, we have seen the dawning of the IoE(Internet of Everything) era, where there will be a huge volume of data generated by things that are immersed in our daily life, and hundreds of applications will be deployed at the edge to consume these data[11]. However, due to the limited computing resources of the sensors, the overload resource usage may occur. In order to satisfy the requirements for strong computing power, edge computing, which emerges as a novel paradigm, provides computing resources at the edge of networks [12].

As for edge computing application, some researchers identified that Mobile edge computing (MEC) technology integrates IT service environment and cloud computing technology at the edge of the network, improving the capacities of computing and storage of the edge of network, reducing network operation and service delivery delay and further enhancing the quality of experience (QoE) of users. Thus, MEC has been added in the 5G standard as a key technology [13].

In this paper, we hope that edge computing pattern can enhance blade icing prediction accuracy and instantaneity.

**Blade Icing Detection Model**

In this section, we describe the model of blade icing detection based on edge computing.

**Process of Gaining the Model**

The process of gaining the model is shown in Figure 1.

1) **Gaining Data Set**

Gain all historic monitoring data of wind turbines in winter from historic data base, and then identify ice records with 1 and non-ice ones with 0, which can form sample data.

2) **Classifying**

Get wind turbine classifications according to the position and the type of wind turbine, and then judging by these classifications of wind turbine, classify the sample data into subclassification sample data.
3) Training Subclasses
Acquire the blade icing detection model of every subclassification of wind turbine via machine learning algorithm on subclassification sample data. The machine learning algorithms of each subclassification may be different as long as the algorithm can be helpful to get better classification effect on the sample data.

The sample data need to be preprocessed to enhance accuracy and avoid overfit to sample data, such as denoising, adjusting occupation between the number of ice state and non-ice to 1:1, normalizing to reduce influence of numeric value between sensor data.

The blade icing detection model of subclassification can be called sub-model, different from the blade icing detection model of wind turbine.

4) Getting Model Set
The all sub-models and the machine learning algorithms corresponding with sub-model make up the blade icing detection model and which is model set.

5) Deploying Submodel
Each wind turbine is viewed as an edge, and the sub-models in blade icing detection model are assigned to every wind turbine according to its classification. Wind turbine will have edge computing ability after getting blade icing detection sub-model matching with it.

6) Predicting Icing
The icing state of wind turbine is predicted via sub-model corresponding with turbine, regarding real-time monitoring data as input.

Edge Computing Node

7) Edge Computing Node
In this blade icing detection model, every wind turbine in wind park is viewed as an edge and has its edge computing node which is sub-model matching with turbine. Through real-time monitoring data gained by all kinds of sensors, sub-model acquired by machine learning algorithm predicts icing state.

8) Relation between Wind turbine and edge computing node
The relations between wind turbine, wind turbine classification, sub-model and blade icing detection model are shown in Figure 2.

![Diagram](image)

Figure 2. Relations chart between wind turbine, wind turbine classification, sub-model and blade icing detection model.

9) Deployment ways
The edge computing deployment of wind turbine has two ways: one is that one computing unit which can load sub-model is installed on one wind turbine. The computing unit collects real-time monitoring data of the wind turbine and locally predicts icing state, finally upload the prediction result to server in control. Server reorganizes results to show operators. This way has advantages such as instantaneity, less communication, calculation quick, corresponding with disadvantages such as high cost and maintain.

The other is that all sub-models are saved on server and built relations to each wind turbine. After wind turbine monitoring data upload server, sub-model matching with turbine predicts icing state. That means all edge computing nodes are deployed on server so that server load is very heavy.
Delay Introduction

Icing state of wind turbine is predicted according to real-time monitoring data, and it means that wind turbine has an instant icing state at every time. Some ridiculous prediction results maybe occur like that state of wind turbine is ice, then non-ice, then ice again, then non-ice. It’s impossible in fact. The fact is that once wind turbine blade is icing, the state usually lasts for a while, cannot disappear instantly and then appear at once. The prediction result is different from the blade icing fact. Hence, we need to introduce delay time $t$ into the icing inertial system to revise edge computing node to enhance prediction accuracy.

After introduction of delay $t$, the determination process from non-ice state to ice is so: once a real-time monitoring data record of wind turbine is determined to be icing state by sub-model, edge computing node need to continue to judge next dozens of real-time data records in $t$ time, instead of reporting icing state to server right now. In dozens of prediction results, if ratio of icing is more than some threshold, all prediction results in delay $t$ will be set icing state. And then edge computing node communicate with server to send all these dozens of prediction results. If the ratio is less than the threshold, the state of wind turbine will not be changed.

The same will be done to judge whether from non-ice state to ice.

So, state of wind turbine is determined by all real-time data records in delay $t$, instead of one.

Experimental Evaluation

Experimental Data

This paper took the historic data of 1# and 2# wind turbine in some park as sample data set. Each wind turbine has 26 sensors which can real-time get state data. The 26 data items are shown Table I.

There are over 37 hundred thousand original records in sample data set. The numeric construction of the two wind turbines is shown Table 2.

The sample data need to be preprocessed, such as denoising, normalizing, regularizing. And finally, a sample data set with 200,000 records is gained. At the same time, we can get two training sample data set: one came from 1# wind turbine with 38000 records; the other formed from 2# wind turbine with 20000 records.

Table 2. Numeric construction of 1# and 2# wind turbines.

| Wind turbine | Amount | Occupation | Total proportion |
|--------------|--------|------------|------------------|
| 1# Normal    | 171043 | 87.74%     | 52.05%           |
| 1# Icing     | 23892  | 12.26%     |                  |
| 1# Total     | 194935 | 100%       |                  |
| 2# Normal    | 168930 | 94.08%     | 47.95%           |
| 2# Icing     | 10638  | 5.92%      |                  |
| 2# Total     | 179568 | 100%       |                  |
| Total Normal | 339973 | 90.78%     |                  |
| Total Icing  | 34530  | 9.22%      |                  |
| Total        | 374503 | 100%       |                  |

The distribution of ice records and non-ice is shown as Figure 3.

![Figure 3. Distribution chart of ice and non-ice records in sample data set.](image)

Experimental Results and Discussion

Using simple Logical Regression algorithm to train on sample data, a sub-model will be got. Initial prediction sub-model
Without delay time, the prediction results is shown in Figure 4. The accuracy of prediction icing state is 84.25%, while the accuracy of prediction normal state is 81.23%, and the average accuracy is 81.62%.

11) Prediction sub-model with delay
With delay time, the prediction results is shown in Figure 5. The accuracy of prediction icing state is 88.26%, while the accuracy of prediction normal state is 90.43%, and the average accuracy is 90.12%.

The introduction of delay can improve accuracy.

12) Prediction model
If we use 1# plus 2# wind turbine sample data as one big total training sample data, a single model will be gained. And using the single model to test, the prediction accuracy is only nearby 86%. While if we use 1# and 2# wind turbine sample data separately to get two sub-models, and then forming edge computing node to test data via sub-models corresponding with wind turbine, the accuracy will enhance to 91%.

Conclusion
In this paper, taking blade icing of wind turbine for example, we propose a method to detect remote alarm based on edge computing in wind energy. This method is model cluster including many sub-models which are obtained via machine learning training on monitoring data separately, and it is possible for every sub-model to have different algorithm. Regarding as edge computing node, each sub-model will be deployed to wind turbine corresponding with it and directly use wind turbine real-time monitoring data to predict alarm to enhance total prediction accuracy. Experiments show this method can have ability of being compatible with traits of wind turbine, as well as improving generalization and total prediction accuracy of remote alarm in wind energy.

References
[1] China Industrial Information. In 2018, the statistical analysis of power consumption by industry, thermal power, hydropower, nuclear power and wind power generation accounted for respectively[online]. Available: http://www.chyxx.com/industry/201807/663652.html, November 16, 2019.
[2] Blasco, P.; Palacios, J.; Schmitz, S. Effect of icing roughness on wind turbine power production[J]. Wind Energy, vol. 20, n. 4, pp. 601-17, April 2017.
[3] Xu Bin; Guo Xueyi; Chen Hongbing. Active Icing Monitoring for Wind Turbine Blade Models With PZT Technology[J]. Piezoelectrics & Acoustooptics, vol. 39, 2017, 02, pp. 72-76.
[4] Sabatier, J; Lanusse, P; Feytout, B; Gracia, S. Ice Accretion Detection and Anti-icing/Deicing Systems for Wind Turbine Blades[J]. Informatics in Control, Automation And Robotics, Icinco, July 2017, vol. 495, pp. 641-663.
[5] Kabardin, I.K.; Meledin, V.G.; Dvoinishnikov, S.V.; et, al. Remote monitoring of ice loading on wind turbine blades based on total internal reflection[J]. Journal of Engineering Thermophysics, vol. 25, n 4, pp. 504-8, Oct. 2016.

[6] Liu, Y; Cheng, H; Kong, X; etc. Intelligent wind turbine blade icing detection using supervisory control and data acquisition data and ensemble deep learning[J]. Energy Science & Engineering. Oct. 2019.

[7] Al-Omary, A.Y.; Jamil, M.S. A new approach of clustering based machine-learning algorithm[J]. Knowledge-Based Systems, Aug. 2006, vol 19, n. 4, pp. 248-58.

[8] Chaudhari, Poonam, Agarwal, Himanshu. Progressive review towards deep learning techniques[J]. Advances in Intelligent Systems and Computing, 2017, vol. 468, pp. 151-158.

[9] Veena, K.M.; Manjula Shenoy, K.; Ajitha Shenoy, K.B. Performance comparison of machine learning classification algorithms[J]. Communications in Computer and Information Science, 2018, vol, 906, pp. 489-497.

[10] Ngadisih; Bhandary, N.P.; Yatabe, R.; et, al. Logistic regression and artificial neural network models for mapping of regional-scale landslide susceptibility in volcanic mountains of West Java (Indonesia)[J]. AIP Conference Proceedings, v 1730, 060001 (11 pp.), 6 May 2016.

[11] Shi Weisong, Sun Hui, Cao Jie, etc. Edge Computing—An Emerging Computing Model for the Internet of Everything Era[J]. Journal of Computer Research and Development, 2017, vol. 54(05), pp. 907-924.

[12] Zhanyang Xu, Xihua Liu, Gaoxing Jiang, Bowei Tang. A time-efficient data offloading method with privacy preservation for intelligent sensors in edge computing[J]. EURASIP Journal on Wireless Communications and Networking, 2019, Vol. 2019 (1), pp. 1-12.

[13] Li Zishu, Xie Renchao, Sun Li, etc. A survey of mobile edge computing[J]. Telecommunications Science, 2018, vol. 34(01), pp. 87-101.