Spatial wind power forecasting using a GRU-based model: WindTeam CSU123

Zhi Liu
Min Li
Baichuan Yang
liuzhi407@163.com
cnlimin@csu.edu.cn
1393803667@qq.com
Central South University
Changsha, Hunan, China

He Wei
whywei@tencent.com
Machine learning platform department, Tencent
Beijing, China

ABSTRACT

This paper summarizes technical details of the machine-learning forecasts of wind power of 134 turbines using a unique Spatial Dynamic Wind Power Forecasting dataset. The method of k-nearest-neighbor was applied to interpolate the missing values. The baseline gate recurrent unit model was selected as the predictor because of its moderate complexity and acceptable performance. To reduce computational time, we selected six typical turbines from the 134 turbines for hyperparameter optimization by the method of grid search. The suboptimum hyperparameters were applied to estimate wind power supply of all turbines. The codes of this work can be found at https://github.com/LiuZhihhxx/KDD2022.

CCS CONCEPTS

• **Applied computing** → **Engineering**; **Forecasting**; • **Hardware** → **Renewable energy**.

KEYWORDS

Wind power forecast, Deep learning, Wind resource, Time series, PaddlePaddle

ACM Reference Format:
Zhi Liu, Min Li, Baichuan Yang, and He Wei. 2022. Spatial wind power forecasting using a GRU-based model: WindTeam CSU123. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Wind Power Forecasting (WPF) is at the heart of minimizing the uncertainty of the integration of wind power plants into the grid [1], including reserves allocation, the scheduling of conventional power plants, and the optimization of the electricity value in the market. Although various methods are avaliable such as conventional regression-based methods[2, 3, 5], machine learning[8, 10, 11], deep learning methods[6, 7, 9], and hybrids of these methods[4, 12], WPF with high accuracy an precision is still a great challenge.

To meet this challenge, Baidu KDD CUP 2022 launched the task of Spatial Dynamic Wind Power Forercasting (SDWPF)[13], a task that required all teams to accurately estimate the wind power supply of a wind farm. The organizer provided a unique SDWPF dataset, which has two features differing from previous WPF competition settings:

1) involving the spatial distribution of all wind turbines.

2) providing important weather data and turbine internal contexts.

Specifically, this task required that each team must estimate the wind power supplied by each turbine and the total power of the 134-turbine wind farm. The forecast should be 10-min time resolution with a 2-day forecast horizon.

2 SOLUTION OVERVIEW

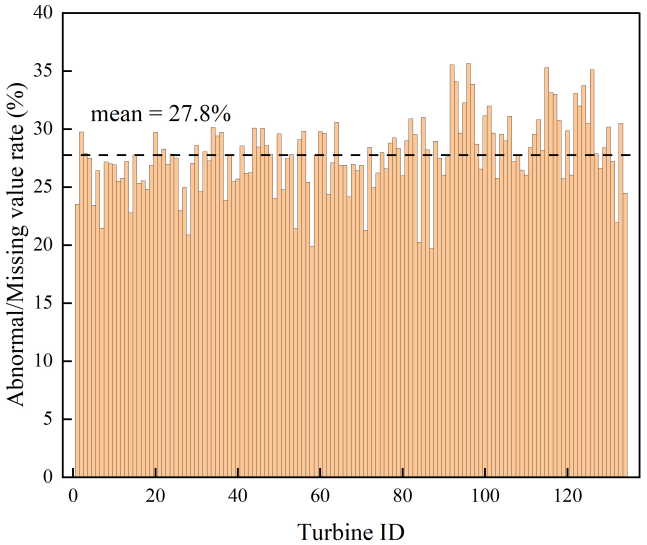

Figure 1: Abnormal/missing value proportion of each turbine.

We used a 3-step solution approach to accomplishing the competition task: data cleaning, hyperparameter optimization, and modeling.

Unpublished working draft. Not for distribution.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference'17, July 2017, Washington, DC, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

The original dataset is incomplete in missing data and contains some abnormal values having no physical meaning (Fig. 1). The missing data were estimated by an interpolation algorithm, and the abnormal values were set to zeros as required by the competition document.

Considering the problem scale and the computational efficiency, we decided to use the baseline Gate Recurrent Unit (GRU) model for predicting the wind power supply. The recorded data of six typical turbines were used as the validation set for optimizing the hyperparameters by a grid search strategy.

3 DETAILED METHOD

3.1 Data cleaning

As shown in Fig. 1, the missing and abnormal data account for a large proportion of the dataset. To facilitate the forecasting, we used the method of k-nearest-neighbor (kNN) to interpolate the missing values[3]. The kNN interpolation identifies neighboring points by distance measurements and can use the full value of neighboring observations to estimate missing values. The number of neighbors k is an essential parameter in the kNN algorithm. A large k tends to fill the missing value with the mean value of the entire dataset, and a small k may introduce noise neighbors. In our computation, ten nearest neighbor values were used in the kNN interpolation.

3.2 Model

The baseline GRU model [3] was used because it is moderately complex and performs quite well in solving time-series problems. GRUs use a single gating unit to simultaneously control the forgetting factor and the decision to update the state unit. For the competition task, we used the cleaned data for constructing 134 GRU models with the same structure.

PaddlePaddle provides many APIs for organizing and training any kind of deep learning model. For example, the GRU model is included in *paddle.nn*, an API including different network models. In the *PaddlePaddle* framework, data are almost treated as tensors, and the *paddle.tensor* API offers functions related to tensor computations. The data flow in the network is as follows: the data is first transformed into a tensor that fits the dimensions of the model, and then flows through a GRU layer(s), a dropout layer, and a linear layer, resulting in a one-dimensional vector output of length 288.

Learning rate is so adjusted that the learning rate declines as the epoch increases, allowing the loss function to decrease fast at the early stage of the training while avoiding divergence at the later stage. Additionally, an early-stopping strategy was applied to avoid overfitting. The Adam optimizer included in the *paddle.optimizer* was used for training the GRU model.

3.3 Hyperparameter optimization

Six typical turbines were selected for optimizing the model hyperparameters. Fig. 2 indicates the locations of the selected turbines (labeled in red). Though the choice is somewhat random, the selected turbines are evenly distributed in the wind farm so that these turbines could be representative of the overall situation. The evaluation script was modified to evaluate the partially selected models based on the officially offered test-1 data. The prep_env() method in prepare.py was also rewritten to allow tuning the hyperparameters by the grid search method.

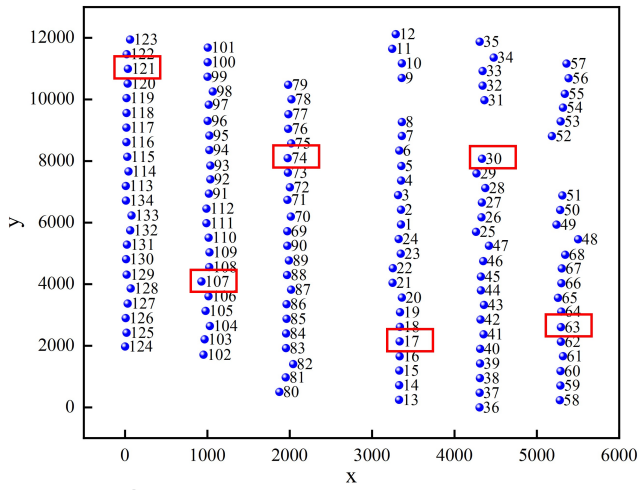

Figure 2: Locations of the selected turbines for hyperparameter optimization.

4 EXPERIMENT PARTS

In practice, the kNNImputer method included in the Python scikit-learn library was used to realize kNN interpolation. The number of neighbors was set as 10, and the Euclidean distance was calculated as the standard distance between objects. The processed dataset was saved as a CSV table named wtdata_245days_knned and used to train the 134 models.

We selected six turbines (17, 30, 63, 74, 107, and 121) to build the model and generate the grid search. These turbines are evenly distributed in location throughout the wind farm and represent the overall pattern to some extent.

The optimized hyperparameters include batch size, learning rate, number of GRU layers, and input length. In the grid search, the candidate values of the four hyperparameters are: input length=[72, 144, 288], number of layers=[1, 2, 3], batch size=[16, 32, 64], and learning rate=[1e-4, 5e-4]. The evaluation results based on the validation data is shown in Table 1. The best combination is [144, 2, 5e-4, 64], which results in a score of -0.9411.

Note that there are several hyperparameter combinations resulting in a lower score than that of the selected combination [144, 2, 5e-4, 64]. This is because these combinations lead to non-converging models and hence yield NaN prediction values, which should be excluded. In the predict script, we replaced the NaN value with zero so that warning messages could be seen in the evaluation output. In detail, small batch size and large learning rate are more likely to lead to non-convergence models. Although the best hyperparameters worked well on the released test data, there are 196 pairs of data_x and data_y on the server, leading to different overall results.

REFERENCES

[1] Hao Chen, Yngve Birkelund, and Qixia Zhang. 2021. Data-augmented sequential deep learning for wind power forecasting. *Energy Conversion and Management* 248 (2021). https://doi.org/10.1016/j.enconman.2021.114790

Unpublished working draft. Not for distribution.

[2] Gregor Giebel and George Kariniotakis. 2017. Wind power forecasting—a review of the state of the art. In *Renewable Energy Forecasting*, George Kariniotakis (Ed.). Woodhead Publishing, 59–109. https://doi.org/10.1016/B978-0-08-100504-0.00003-2

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[4] Md Alamgir Hossain, Ripon K Chakrabortty, Sondoss Elsawah, and Michael J Ryan. 2021. Very short-term forecasting of wind power generation using hybrid deep learning model. *Journal of Cleaner Production* 296 (2021), 126564.

[5] Xin Jin, Yao Dong, Jie Wu, and Wang Jujie. 2010. An improved combined forecasting method for electric power load based on autoregressive integrated moving average model. , 476-480 pages. https://doi.org/10.1109/ISME.2010.124

[6] Chaoshun Li, Geng Tang, Xiaoming Xue, Adnan Saeed, and Xin Hu. 2019. Short-term wind speed interval prediction based on ensemble GRU model. *IEEE transactions on sustainable energy* 11, 3 (2019), 1370–1380.

[7] G. Memarzadeh and F. Keynia. 2020. A new short-term wind speed forecasting method based on fine-tuned LSTM neural network and optimal input sets. *Energy Conversion and Management* 213 (2020). https://doi.org/10.1016/j.enconman.2020.112824

[8] Mohamed Ahmed Mohandes, Talal Omar Halawani, S. Rehman, and Ahmed A. Hussain. 2004. Support vector machines for wind speed prediction. *Renewable Energy* 29, 6 (2004), 939–947. https://doi.org/10.1016/j.renene.2003.11.009

[9] Qin Qiong, Lai Xu, and Zou Jin. 2019. Direct multistep wind speed forecasting using LSTM neural network combining EEMD and fuzzy entropy. *Applied Sciences-Basel* 9, 1 (2019). https://doi.org/10.3390/app9010126

[10] Sancho Salcedo-Sanz, Emilio G.Ortiz-Garcı´a, Ángel M.Pérez-Bellido, AntonioPortilla-Figueras, and LuisPrieto. 2011. Short term wind speed prediction based on evolutionary support vector regression algorithms. *Expert Systems with Applications* 38, 4 (2011), 4052–4057. https://doi.org/10.1016/j.eswa.2010.09.067

[11] Zhao Shuangyi, Zhao Jing, Zhao Ge, Zhang Wenyu, and Guo Zhenhai. 2010. Effective wind power density prediction based on neural networks. , 4 pages. https://doi.org/10.1109/icmult.2010.5631154

[12] Shuai Zhang, Yong Chen, Jiuhong Xiao, Wenyu Zhang, and Ruijun Feng. 2021. Hybrid wind speed forecasting model based on multivariate data secondary decomposition approach and deep learning algorithm with attention mechanism. *Renewable Energy* 174 (2021), 688–704. https://doi.org/0.1016/j.renene.2021.04.091

[13] Jingbo Zhou, Xinjiang Lu, Yixiong Xiao, Jiantao Su, Junfu Lyu, Yanjun Ma, and Dejing Dou. 2022. SDWPF: A Dataset for Spatial Dynamic Wind Power Forecasting Challenge at KDD Cup 2022. *Techincal Report* (2022).

Unpublished working draft. Not for distribution.