trymore: Solution to Spatial Dynamic Wind Power Forecasting for KDD Cup 2022

Hanhan Liu
Sieyuan Electric Co., Ltd
Shanghai, China
hanhanliu0214@gmail.com

Haowei Yang
Sieyuan Electric Co., Ltd
Shanghai, China
yeungharvey@gmail.com

ABSTRACT

Over the past few decades, the research into wind power prediction has exploded. As an important clean energy, wind power also has the characteristics of instability and volatility. Therefore, to ensure the stable operation of the power grid, it is necessary to effectively predict the future power generation of wind farms, and then stabilize the influence of wind power fluctuations on the power grid by using other energy sources. There are still some challenges in predicting the future power output of wind turbines smoothly and accurately, such as the large variation in the quality of historical data of each wind turbine and the increasing error of multi-step prediction over time. In this paper, we proposed a solution to the Spatial Dynamic Wind Power Forecasting Challenge at KDD Cup 2022. The task of this competition is to effectively predict the power generation of 134 wind turbines at the wind farm over the next two days. Through the analysis of SDWPF data set, we designed an effective data processing flow, and analyzed the impact of wind turbine data on the overall wind farm output, and designed a multi-model hybrid prediction integration scheme. The model consists of three modules: LightGBM, GRU and Local-Ensemble. We use different feature combinations and individualized multi-step prediction schemes for each model to achieve efficient learning. Finally, the multi-model multi-step prediction results are mutually corrected to output the final result. Experiments show that the method proposed in this paper has obvious advantages in solving the spatial dynamic wind power prediction problem.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

Wind Power Forecasting, Machine Learning, Deep Learning, KDD CUP 2022

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Hanhan Liu and Haowei Yang. 2022. trymore: Solution to Spatial Dynamic Wind Power Forecasting for KDD Cup 2022. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Wind power plays a very important role in power grids around the world. Wind energy has become an important source of global energy due to its pollution-free and wide availability[9]. Wind power forecasting (WPF) aims to accurately estimate the wind supply of wind farms over different time scales. Accurate and reliable wind power output prediction technology can provide decision support for power dispatchers to adjust power generation plans and control the operation capacity of wind turbines in time. This helps the power dispatch department to optimize and organize the power generation scheme, so as to improve the reliability and security of the power grid[3]. Therefore, WPF is widely regarded as one of the most critical issues in wind power integration and operation. In the field of data mining and machine learning, there has been a lot of research on the prediction of wind power generation. However, getting WPF right remains a challenge.

- Complete and correct data recording is a powerful guarantee for wind power output forecasting technology. However, due to some reasons, a large number of outliers appear in the collected historical data, which brings trouble to correctly judge the turbine state and the relationship between characteristics.
- Available data provide very limited useful features. How to use limited features to extract the hidden feature correlation and contextual information provided by the turbine location matrix to extract the correlation between turbines is very important for the accuracy of turbine power output and the correlation of power prediction between turbines.
- The longer the time range of wind power multi-step single output forecasting, the more unknown factors, and the accuracy of prediction decreases with the increase of prediction step. This means that the accuracy of multi-step time series prediction of wind turbine power generation is challenging and important.

In order to solve the above problems, this paper deals with abnormal data by analyzing data rules and data constraint rules to reduce the interference of outliers on model stability. Using the contextual information provided by the Wind turbine location matrix, the similarity among turbines was analyzed and the wind turbines were grouped. In order to better characterize the interaction between various influencing factors, this paper adopts the method of feature engineering to expand the turbine features by capturing and utilizing the dynamic dependency between multiple variables. At present, a series of prediction models obtain the global optimal prediction performance by using the fusion of two or more single models[7]. In this paper, multiple feature combinations are used, and different models are used to learn different feature combinations, and differences are constructed through different models, data, and modeling methods.

The main contributions of this paper are as follows:

- We propose a multi-model integrated framework for wind power forecasting, which constructs differences by using different feature sets for different models, Develop a personalized multi-step forecasting plan for each model.
- By analyzing the historical data of different wind turbines, the wind turbines with the most average power output are selected as the data quality standard, and the multi-step correction results are output by its model.
- The multi-step results are corrected by using the differences between the models, and the long-step results are corrected with the short-step model results to improve the long-term prediction accuracy.
- Through the analysis of SDWPF data, we identified several major problems in the task, and grouped them through the wind turbine matrix from the perspective of data characteristics. These challenges were solved by data filling, outlier processing, feature enhancement and other methods.

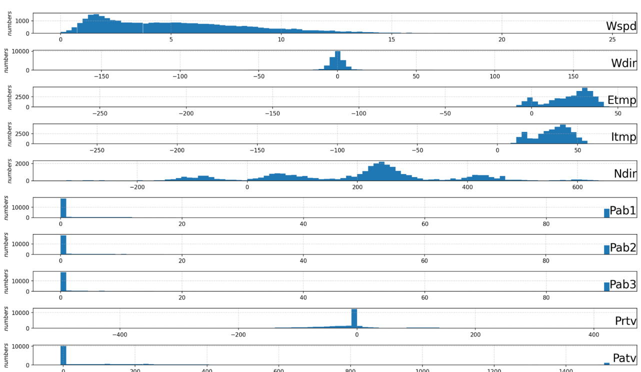

Figure 1: Illustration of data distribution of each column of one wind turbine

2 RELATED WORK

With the rapid development of technology, the demand for wind energy is huge and growing rapidly. Wind energy has great potential for energy conversion and will make a significant contribution to the world's electricity demand. Wind energy production is very sensitive to climate variables, such as geographical location, wind speed, pressure, temperature, wind direction, etc.The patterns of wind are quite unstable in nature[6]. Therefore, direct statistical models do not provide accurate predictions.

Some researchers use statistical methods to predict short-term wind power generation statistical models including the historical average (HA) method and the autoregressed-moving average (ARMA) method[2]. ARMA is the most known method based on time series for predicting the future value of wind power generation. The researchers tried several variations of ARMA (e.g., ARIMA) to achieve better prediction performance.

With the rapid development of machine learning technology, many researches on WPF have been carried out with different prediction methods and different levels. Machine learning is a subdivision of statistical methods. It can learn patterns from data and make predictions accordingly[1]. For example, ANN (Artificial Neural Networks), LightGBM, Decision Tree, Bagging, Random Forest are widely used to predict the value of wind power generation.

In addition to traditional machine learning models, extreme learning and deep learning are also gaining more and more attention in wind speed and power prediction. These advanced learning models have higher accuracy and can learn more complex nonlinear relationships. Yubotao et al.[8] created a model using deep belief networks. Deep belief networks (DBN) are more accurate at extracting latent rules of wind energy from historical wind farm data.

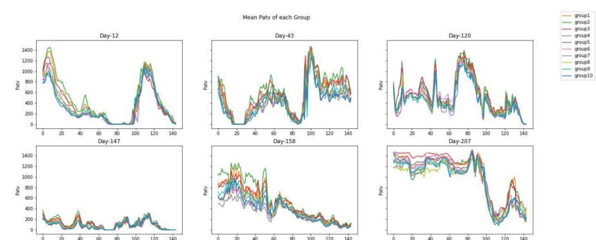

Figure 2: According to the location of wind turbines, the closer turbines are divided into a group, and each subdiagram shows the average power output of each group in a single day.

Considering the context information learning of time series, recursive neural network (RNN) model can extract the explicit time dependence of sequence learning. Yiwei Fu et al.[4], proposed a prediction model of RNN based on Gru to improve accuracy. First, the overall prediction framework of wind power generation and several optional hybrid models are presented. Using NWP data and wind speed correction program, a prediction model based on Gru is established. The results show that Gru model has obvious advantages.

However, due to the limited performance of the single model mentioned above, hybrid models combining different technologies are becoming more and more popular in wind power generation prediction. By combining the model [5] in the pre-processing or post-processing stage, the predictive performance of two hybrid methods is improved. Based on the excellent performance of the hybrid prediction method, this paper also adopts the hybrid prediction method, and adopts the weighting method in the post-processing stage, assigns a weight coefficient to the prediction of each model according to the effectiveness of the model, and improves the performance of the final prediction by combining different prediction methods.

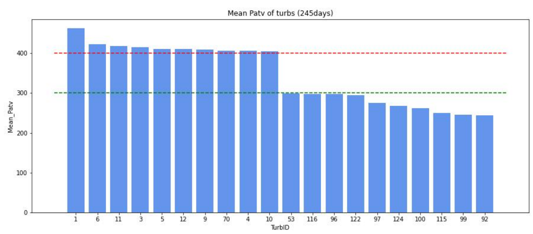

Figure 3: Mean_Patv vs. TurbID at 245 days wtbdata. Left ten bars are the ten turbines with most power output (mean_patv > 400), and right ten bars are the ten turbines with least power output (mean_patv < 300).

3 PROPOSED METHOD

3.1 Overview

Fig.4 shows the overall framework proposed in this paper. Based on the weighted hybrid prediction method, our method designs three main modules, each of which plays a different role in the final prediction. In order to obtain better integration results, we use different feature sets for different models to build differences. Next, we will introduce our data processing and the details of the three component models.

3.2 Data Preprocessing

As shown in the Fig.1, we have checked the historical data of 134 wind turbines, in which there are a large number of null values and outliers, so we need to process the data before starting work. The data processing part is divided into two parts: data pretreatment and feature engineering.

3.2.1 Data cleaning and filling.

(1) `Abnormal data handling`: Filter the abnormal data by setting the attributes of the abnormal data records other than TurbId and Day to Nan. The abnormal conditions include:
   - Patv<0.
   - Wspd<1 and Patv>10
   - Wspd<2 and Patv>100
   - Wspd<3 and Patv>200
   - Wspd>2.5 and Patv==0
   - Wspd==0 and Wdir==0 and Etmp==0
   - Etmp<-21
   - Itmp<-21
   - Etmp>60
   - ITmp>70
   - Wdir>180 or Wdir<-180
   - Ndir>720 or Ndir<-720
   - Pab1>89 or Pab2>89 or Pab3>89

(2) `Group fill`: As shown in the Fig.2 , according to the spatial position matrix of the wind turbine, the closer the distance is, the more similar the external environment and power output of the wind turbine will be. We divided 134 turbines into ten groups according to the principle of proximity. In each group, the attributes of Nan are populated with the average of other non-Nan values in that group on the Tmstamp dimension.

(3) The number of Nan Patv of 134 turbine in a day was counted as NanCount, and the records whose NanCount is less than or equal to 28, were filled with the records before or after them.

In the original data, the total number of Patv less than 0 or equals to Nan is 1312580, which is 582716 after the above three steps of data processing.

3.2.2 Feature engineering.

(1) Normalize the hours and minutes with sin and cos.
(2) Perform a multi-step lag operator on the data column and expand it into a new column, identified by lag_N.
(3) Perform a multi-step difference operation on the data column and expand it into a new column, identified by diff_N.
(4) Considering the sensitivity of temperature change to air flow and wind speed, we average the top n temperature maxima every day and add them to the newly expanded column etmp_Max.

3.3 Model

3.3.1 GRU Model. WPF is a typical scenario of sequence modeling, and RNN model is mainly used for sequence prediction, which urges us to integrate RNN model into our integration. Gru network is a variant of LSTM network, which has fewer parameters and faster training speed. Therefore, this paper selects Gru to obtain the explicit time dependence of time series. Considering the impact of data integrity and data quality on the RNN model, in this module, we train each wind turbine separately and output an independent model. It is worth noting that we expect the RNN to learn the hidden associations between its features on the original features, so in this module we train on the preprocessed data without feature engineering and feature expansion. As shown in the Fig.4, the GRU module Layer is set to 1, using multi-step input and multi-step output mode for multi-step prediction. Such as inputs (step:144,step:126,,step:108...) and outputs (step:288,step:252,step:216...), each step outputs a corresponding model, which is Each wind turbine acquires a number of models with different output steps.

3.3.2 LightGBM. As an efficient gradient enhanced decision tree, LightGBM has remarkable ability in processing noise data, making it a popular choice for wide application. For LightGBM, we first perform feature engineering on the preprocessed data (the benchmark dataset is the result of Section 3.2.1). In addition to normalizing the time, we perform multi-step lag operator operations on the data columns, and generate the corresponding extended column column_lag_N for each column($N \in [1, 6]$).At the same time, perform multi-step difference operations on the data columns, and generate corresponding extended column column_diff_N ($N \in [1, 6]$) for each column, and establish the relationship between historical time points and current data through these two operations. Different from the GRU module, the LightGBM module uses the full data of all wind turbines for training, and does not target a single turbine.As shown in the Fig.4, LightGBM adopts a multi-step output strategy, setting the step size to 1, and predicting the next 288 pieces of data in turn through the loop, thereby generating 288 LightGBM models.

3.3.3 Local Ensemble. In order to better characterize the interaction between all influencing factors, and to construct the difference by modeling different models using different feature groups, we

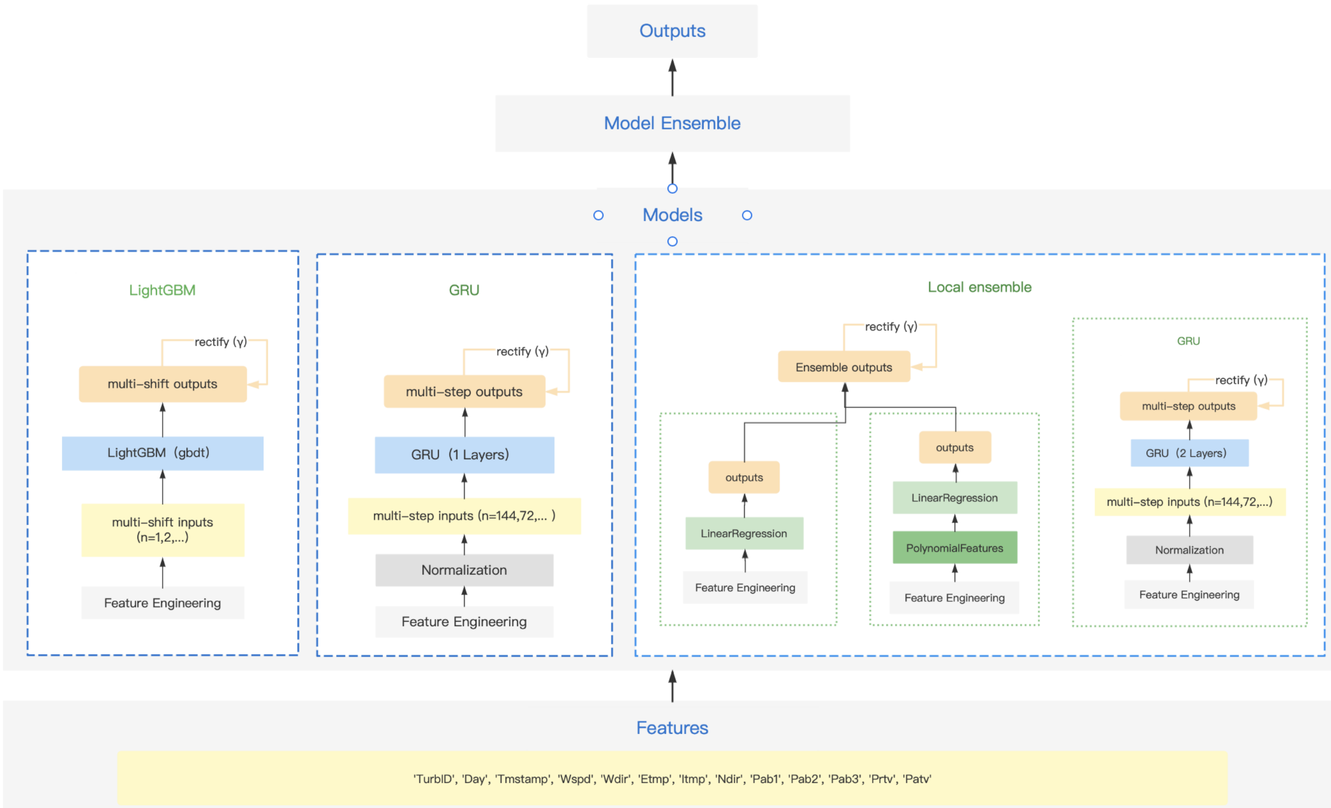

Figure 4: Schematic diagram of the proposed model, which includes LightGBM, GRU Model, and Local-Ensemble model.

designed a local-Ensemble module.As shown in the Fig.4,It contains LinearRegression, PolynomialFeatures + LinearRegression, and local_Gru units.

- `LinearRegression`: It has better generalization ability for unstable prediction in the future, especially for mean-distributed data.We first perform feature engineering on the preprocessed data (the benchmark dataset is the result of Section 3.2.1). In addition to normalizing the time, unlike the LightGBM module, here we consider the sensitivity of temperature changes to airflow and wind speed, average the first 6 maximum temperature values per day, and add them to the newly expanded column Etmp_Max; and use the maximum value in the 7*144 Etmp_Max data to fill the null values with the following data. We then perform multi-step lag operator operations on the data columns, and generate the corresponding extended column column_lag_N for each column($N \in [1, 12]$).At the same time, perform multi-step difference operations on the data columns, and generate corresponding extended column column_diff_N ($N \in [1, 12]$) for each column, and establish the relationship between historical time points and current data through these two operations. And fuse the 288 Patv data after the current row to the current row. For the LinearRegression model, we use the whole station data for full training, and finally output a full model.
- `PolynomialFeatures + LinearRegression`: It has a good fitting ability for ultra-short-term (within 3.5 hours) prediction. We use the same feature engineering as LinearRegression, and also use full-scale training with full-scale data, and finally output a full-scale model.
- `local_Gru`: local_Gru uses the preprocessed data, uses the same temperature feature expansion as LinearRegression, and performs one-hot encoding on the turbine ID. Local_Gru uses two layers of gru layers, and uses the full data to using multi-step input and multi-step output mode for multi-step prediction, such as inputs (step:144,step:72,step:36...) and outputs (step:288,step:144,step:36...), each step outputs a corresponding full model.

3.4 Ensemble Methods

The hybrid prediction method combines different prediction methods to improve the performance of the final prediction. The performance of a single model is limited in many cases. Hybrid forecasting methods combine the ability of multiple models to better accommodate changes in the sample by setting the results of each model in combination. For time series forecasting, the closer the time is to the starting point, the more accurate the forecast, and the error will become larger as the forecast step increases. Considering this feature, we propose a multi-step correction scheme.As introduced in the Section 3.3, we use the multi-step multi-model training method in model training to generate multiple models with different step sizes, and use multiple short-step model results to correct the long-step results to make the tracking results more accurate.

For example, in GRU MODEL, we use the step size to predict in order from small to large, and use the short-step prediction in turn to correct the prediction result of the previous step. For example, if the output with a step size of 36 is used to correct the output with a step size of 72, the output of the nearest 36 steps of the two predictions is averaged, and then the difference is made to obtain the difference between the long-term and short-term predictions.Then use the difference to correct proportionally for the last 36 steps with the step size of 72. The specific formula is as follows:

$$M_i = mean(P_i[0 : step_i]) \tag{1}$$

$$M_{i+1} = mean(P_{i+1}[0 : step_i]) \tag{2}$$

$$P[step_i : step_{i+1}] = P_i[step_i :] - \sigma(M_{i+1} - M_i) \tag{3}$$

σ is the correction factor.i represents the model, step represents the step size of the model output and $step_i \in (12, 36, 72, 144, 180, et.al)$

The LightGBM module performs multi-step result correction in the same way as the GRU Model.

The LightGBM module performs multi-step result correction in the same way as the GRU Model. In Local-Ensemble, we first make the first 22 predicted by PolynomialFeatures + LinearRegression to modify the first 22 predicted by LinearRegression. And pass the corrected LinearRegression result P_l as the transmission, like:

$$P_l[: 22] = P_p[: 22] \tag{4}$$

And then use the same multi-step result correction method as GRU Model to correct by increasing the step size by 4.In addition, for the prediction results of local_gru, we take the average of the first 144-step prediction and the last 144-step prediction, and make the difference, and use the difference to correct the last 144-step prediction through the proportional coefficient.

In the weighting method, we assign a weight coefficient to the prediction of each model according to the effectiveness of the model, and fine tune the weight through experiments to achieve better results. The weight distribution in Local-Ensemble is as follows:

$$P_{l_e} = \omega * P_l + (1 - \omega) * P_{l\_gru} \tag{5}$$

where P_{l_e} represents the prediction result of Local-Ensemble module, P_l represents the prediction result of LinearRegression, and P_{l_gru} represents the prediction result of local_gru module. We set $\omega = 0.59$.

The overall architecture weight distribution is as follows::

$$P_{final} = \omega_1 * (\omega_2 * P_{gru} + \omega_2 * P_{lgm}) + \omega_3 * P_{l_e} \tag{6}$$

where P_{gru} represents the prediction result of Gru module, P_{lgm} represents the prediction result of LightGBM module, and P_{l_e} represents the prediction result of local ensemble module. We set $\omega_1 = 0.75$,$\omega_2 = 0.5$ and $\omega_3 = 0.25$.

4 EXPERIMENTS

In this experiment, the 245 day data set is used as the benchmark to divide the training set and the verification set, and the multi model is used for training. Then, the reasoning results of each training model are fused and modified. Finally, the final output result is obtained by weighted combination of the modified data according to a specific proportion.

4.1 Datasets

The data set provided contains 134 wind turbines, and each wind turbine contains 245 days of data. The test and evaluation data set contains 14 days' data of 134 wind turbines, which is used to predict the output data of each wind turbine every 10 minutes in the next two days.

4.2 Metrics

Predict the output value of each wind turbine in the next 48 hours and 10 minutes, that is, predict a wind power time series with a future length of 288. The average value of RMSE (Root Mean Square Error) and MAE (Mean Absolute Error) is used as the evaluation index. The evaluation index formula of each wind turbine is:

$$s^i_{t_0} = \frac{1}{2}\left(\sqrt{\frac{\sum_{j=1}^{288}(Patv^i_{t_0+j} - \overline{Patv}^i_{t_0+j})^2}{288}} + \frac{\sum_{j=1}^{288}|Patv^i_{t_0+j} - \overline{Patv}^i_{t_0+j}|^2}{288}\right) \tag{7}$$

According to the evaluation index score of each wind turbine, calculate the total score index of 134 wind turbines:

$$S_{t_0} = \sum_{i=1}^{134} s^i_{t_0} \tag{8}$$

4.3 Model Training

4.3.1 linearregression. The data of the first 225 days were used as the training data set, and the last 20 days were used as the verification data set. The current data and the past 12 data after the fusion feature processing are used as input X, and the Patv data of the future 288 data are used as label Y. The LinearRegression is used for model training.

4.3.2 polynomialfeatures + linearregression. The data of the first 225 days were used as the training data set, and the last 20 days were used as the verification data set. Take the current features ('Wspd','Etmp','Patv') and the past 2 data after data feature processing as input X, and the Patv data of the future 288 data as label Y. The Pipeline([('poly', PolynomialFeatures(degree=4)), (' Linear ', LinearRegression())]) is used to perform quadratically polynomial fitting + linear regression for model training.

4.3.3 LightGBM. The data of the first 224 days were used as the training data set, and the last 21 days were used as the verification data set. The current data and the past 6 data after fusion feature processing were used as input X, and each Patv data of the future 288 data was used as label Y. The 288 models were successively trained by LightGBM.

4.3.4 GRU of PaddlePaddle. The data of the first 225 days after feature engineering were used as the training data set, and the last 20 days were used as the verification data set. The GRU network based on PaddlePaddle framework is used to train single turbine multi-step independent training model (GRU Model) and all turbine data to a large model (local_gru).

4.4 Performance Comparison

Model ensemble is a common method to improve model accuracy in algorithmic competitions. Sometimes dozens or even hundreds of models are put into service in the late stages of intense competition. We design a hybrid forecasting method by combining different forecasting methods to improve the performance of the final forecast. Through continuous in-depth research and experiments, as shown in Fig.3, we found that the power output data of each turbine in the wind turbines matrix is quite different. Using the 245-day data set to calculate the average Patv of each site. The higher average power outputs are 1, 6, 11, 3, and 5, and the average output value is greater than 400, while the sites with the lower output value have an average output value of less than 300. This big difference shows that the location and the operating conditions of wind turbines have a big effect on the power output.

Based on the assumption that, the higher the Patv is, the wind power output is closer to the theoretical work. This is subject to less mechanical adjustment and fewer outliers by the turbine itself. At the same time, it is difficult to predict the mechanical adjustment and abnormal conditions. Therefore, we select the wind turbines with the highest wind power output to train the models and use their predictions to represent the output of other turbines. We have tried top15, top10, top5 and top1, finally we found that top5 had the best fit. Maybe top4 or top6 would be better than top5, but we do not have enough time to find the optimal topN. We tested our assumption on both Phase2 and Phase3. Therefore, we only use the five best-performing turbine models to predict, and rectify the results using the multi-step correction method described in section 3.3.1. Finally, we averaged the five corrected prediction results to cover all other turbines.

At LightGBM model, we finally choose the model with step size in (2, 4, 6, 12, ...) for prediction, and perform multi-step result correction. In Local-Ensemble, we usef local_gru to predict turbine No.1, and use the prediction results as the prediction results of each turbine. The specific correction method has been introduced in Section 3.3. Through quantitative comparison, our solution is in the process of optimization, and the performance results are constantly improving. We have recorded the module combination submission process as fellows:

Table 1: Comparison of Schemes

Score	Models	Schemes
-45.15699	LightGBM + GRU	A
-45.12303	LightGBM + GRU	B
-45.10273	LightGBM + GRU + Local-Ensemble	C
-44.92340	LightGBM + GRU + Local-Ensemble	D

The details of each scheme are as fellows:

(A) LightGBM + GRU, GRU uses top 10 turbine models to predict, and the results of LightGBM and GRU rectified by one step
(B) LightGBM + GRU, GRU uses top 10 turbine models to predict, and rectifies the results by multi-step, at the same time, the results of LightGBM rectified by one step
(C) LightGBM + GRU + Local-Ensemble, based on the previous one, Local-Ensemble model is incorporated
(D) LightGBM + GRU + Local-Ensemble, the difference from previous ones, is that GRU use top 5 turbine models to predict, and the results of LightGBM and GRU both rectified by multi-step

5 CONCLUSIONS

In this paper, we introduce our work on Spatial Dynamic Wind Power Forecasting for KDD Cup 2022. We achieved excellent results through data processing, feature design and multi model mixed prediction. In the 2490 team competition, we won the second place. From Talbe.**??**, we can see that we are only 0.006 behind the first place. It is worth pointing out that the competition only provides data elements such as wind speed, temperature and turbine angle. However, different geographical locations and weather changes have a great impact on atmospheric circulation, pressure and real-time wind speed. These factors have a direct impact on the power output of future turbines. Therefore, wind field coordinate information and weather forecast information can be added, which will be of great help to the accuracy of long-term and short-term turbine power prediction. In the future, we will also continue to improve our solution, integrate multiple models into a single model, reduce the number of models, simplify the complexity of training and practical applications. And continue to explore the coefficient adjustment of multi-model mixed prediction, and explore more effective time series problem processing methods.

REFERENCES

[1] Hanieh Borhan Azad, Saad Mekhilef, and Vellapa Gounder Ganapathy. 2014. Long-term wind speed forecasting and general pattern recognition using neural networks. *IEEE Transactions on Sustainable Energy* 5, 2 (2014), 546–553.
[2] George EP Box and David A Pierce. 1970. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association* 65, 332 (1970), 1509–1526.
[3] Niya Chen, Zheng Qian, Ian T Nabney, and Xiaofeng Meng. 2013. Wind power forecasts using Gaussian processes and numerical weather prediction. *IEEE Transactions on Power Systems* 29, 2 (2013), 656–665.
[4] Yiwei Fu, Wei Hu, Maolin Tang, Rui Yu, and Baisi Liu. 2018. Multi-step ahead wind power forecasting based on recurrent neural networks. In *2018 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*. IEEE, 217–222.
[5] Inci Okumus and Ali Dinler. 2016. Current status of wind energy forecasting and a hybrid method for hourly predictions. *Energy Conversion and Management* 123 (2016), 362–371.
[6] S Preethi, H Prithika, M Pramila, and S Birundha. 2021. Predicting the Wind Turbine Power Generation based on Weather Conditions. In *2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. IEEE, 132–139.
[7] Saurabh S Soman, Hamidreza Zareipour, Om Malik, and Paras Mandal. 2010. A review of wind power and wind speed forecasting methods with different time horizons. In *North American power symposium 2010*. IEEE, 1–8.
[8] Yubo Tao, Hongkun Chen, and Chuang Qiu. 2014. Wind power prediction and pattern feature based on deep learning method. In *2014 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*. IEEE, 1–4.
[9] Ryan Wiser, Eric Lantz, Trieu Mai, Jose Zayas, Edgar DeMeo, Ed Eugeni, Jessica Lin-Powers, and Richard Tusing. 2015. Wind vision: A new era for wind power in the United States. *The Electricity Journal* 28, 9 (2015), 120–132.