dataZhi: A multi-scale fusion method for wind power forecasting with spatiotemporal attention networks

Hongzhi Luan
luanhongzhi@hikvision.com
Hikvision Research Institute
Hangzhou, Zhejiang, China

Junxiong Hou
houjunxiong@hikvision.com
Hikvision Research Institute
Hangzhou, Zhejiang, China

ABSTRACT

Accurate wind power forecasting plays an important role in the economic power system operating, and has attracted more and more attentions in recent years. In this paper, we demonstrate a multi-scale fusion method to make more accurate power predictions, with a spatiotemporal network to extract more relevant correlations. By fusion multi models trained with different datasets or with different network configurations, our method benefits a high diversity and thus makes more accurate wind power prediction. We rank 9th in stage 3 of KDD Cup 2022, and the implementation and final submitted models are available on https://github.com/luanhzh/wpf_fusion.

CCS CONCEPTS

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

KEYWORDS

wind power forecasting, seq2seq, temporal aggregation, spatiotemporal attention

ACM Reference Format:
Hongzhi Luan and Junxiong Hou. 2022. dataZhi: A multi-scale fusion method for wind power forecasting with spatiotemporal attention networks. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (KDD Cup 2022).* ACM, Washington DC, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Wind power, as a kind of clean and renewable energy, plays an important role in the daily power supply. Therefore, wind power forecasting has become one of the important technical issues, and attracted extensive research and attention in recent years [1]. On the one hand, precise wind power forecasting is essential for making energy planning and reserve in advance; on the other hand, since wind power is essentially originated from natural wind energy, it is highly volatile and stochastic, and thus there is a big challenge in getting precise predictions.

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.
KDD Cup 2022, August, 2022, Washington DC, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

In essence, wind power forecasting is a time-series forecasting problem, and all the existing forecasting methods [2] can be used for wind power forecasting, such as ARIMA and ETS, etc. Meanwhile, with the fast development of artificial intelligence technology, the use of machine learning and deep learning models for wind power forecasting is a potential way to get high-precision predictions.

[3] proposed a deep learning approach based on encoder-decoder structure, which forecasts wind power generated by a wind turbine using its spatial location relative to other turbines and historical wind speed data, and is proved to be scalable and efficient with experiments on two real word datasets. Similarly, [4] propose a novel sequence-to-sequence model using the Attention-based Gated Recurrent Unit (AGRU) that improves accuracy of forecasting processes, and demonstrated competitive capabilities in wind power forecasting. On the other hand, [5] proposes a novel framework with spatiotemporal attention networks (STAN) for wind power forecasting, by capturing spatial correlations among wind farms and temporal dependencies of wind power time series, STAN achieved influential performance. In addition, a novel two-stage forecasting model based on the error factor, a nonlinear ensemble method and the multi-objective grey wolf optimizer algorithm is proposed for wind power forecasting, and improved both forecasting accuracy and stability[6].

From the above reviewd papers, we found that seq2seq is a common selected network architecture when settling with wind power forecasting , and capture spatiotemporal correlations is quite important to enhancing the forecasting performance, like employing attention mechanism, which is first proposed in [7].

Wind power forecasting has not only attracted great interest in academic research, some competitions are also organized by industry, as a link between academic research and industrial applications. A competition was organized in Kaggle to forecast wind power[8], of which the main task was to forecast how much wind power that could be generated from the windmill for the next 15 days. And now, the KDD Cup 2022 wind power forecasting competition [9], is another more challenging international event, and attracted more than 2400 teams to participate. As the report describes, and also be analyzed with dataset by us, there are the following challenges in this competition:

a. It is hard to capture the inter-temporal patterns since the correlation between time-series feature sequence is weak. The correlation coefficient is nearly close to zero when the delay is greater than 24 hours.

b. The cumulative error of prediction results over time would have a great impact in the multi-step time series prediction task.

c. Relatively high proportion of missing values, unknown values and outliers exist in the raw data. According to the given caveats

about the data [9], there are about 20% rows needed to be filtered out.

d. Spatio correlation and interference need to be considered, which may influence the performance of model greatly.

e. The model may encountered the risk of low generalization, since there may be a big difference in the distribution of the historical data from different time periods.

To solve these challenges, inspired by some sota methods[3][5], we propose a multi-scale fusion method for wind power forecasting with spatiotemporal attention networks, and the main contributions are as follows:

- A Seq2Seq model with a spatiotemporal attention mechanism was proposed for short-term wind power forecasting, in which the Encoder module applied TCN/GRU to capture longer historical time-series dependent information;
- Multi-scale fusion mechanism is employed to provide a high diversity, in which resampled time series with larger scale predictions is used to make a rough but more accurate trend prediction, and small scale fine-grained time series prediction is used to capture more refined seasonal information;
- Some tricks are utilized in the model training, including a large-stride dataset construction strategy to enrich the diversity among datasets, and different fusion configurations.

2 PROBLEM FORMULATION

From a machine learning perspective, the wind power forecasting is to predict the wind power in a future period based on several historical characteristics of the wind turbine. The historical information here mainly includes 10 features, which can be classified into 3 categories: external environmental information (including wind speed, wind direction and ambient temperature), wind turbine status (nacelle direction, nacelle internal temperature, blade 1 angle, blade 2 angle and blade 3 angle) and power information (active power and reactive power).

As for statistical method (statistical method and physical method are two common methods for wind power forecasting), this is a typical time series forecasting task. More specifically, it's a multi-time series, multi-variate and multi-step prediction problem. The multi-time series means that there are 134 wind turbines in the wind farm, and all the wind powers of each turbine need to be predicted; multivariate means that there are 10 characteristics of time series for each turbine; and multi-step forecasting means that it is necessary to forecast the wind turbine power for 288 steps in the next two days at a certain time.

Mathematically, the above process can be abstractly described as:

$$(x_{t-N+1}, x_{t-N+2}, ..., x_t) => (y_{t+1}, y_{t+2}, ..., y_{t+M})$$

where x_i denotes the features at time slot i, and y_j demotes the predicted wind power at future time j. As discussed before, seq2seq architecture is a naturally applicable solution, in which an Encoder is introduced to capture the patterns between historical data and characterize them as a context vector, followed by a Decoder for one-by-one output prediction of results based on this context vector and some other auxiliary features. The process described above can be further refined as

$$context = f(x_{t-N+1}, x_{t-N+2}, ..., x_t)$$

$$(y_{t+1}, y_{t+2}, ..., y_{t+M}) = g(z_{t+1}, z_{t+2}, ..., z_{t+M}|context)$$

where z_k denotes the available features for decoder input at time k. Specifically in this competition, with a 10 minutes time interval, a 288 steps forecast horizon is needed i.e. $M = 288$, which is a quite long forecast horizon, and may cause a large cumulative error as the step length progresses.

In fact, making 288 steps predictions directly is indeed a very challenging task. On the one hand, longer forecasting steps will cause more error accumulation, resulting in less accurate prediction results for the later horizon; on the other hand, wind turbine characteristics (including wind power) are inherently more random and fluctuating, and thus are not conducive to the model capturing time series dependencies, as seen from actual exploration results during the competition. As the number of prediction steps progresses, the prediction curve is nearly close to a smooth straight line, with very small fluctuations, implying that the model's prediction output may be less informative.

As demonstrated in [2], temporal aggregation makes transformation from a time series with high frequency to another of lower frequency, and is an appealing schema to tackle the problem in this competition. For example, by changing the original temporal intervals of the historical data from 10 minutes to 1 hour(i.e., calculate the average value of every 6 samples to produce a synthetic one), the resulting time series has a shorter length with 1/6 times, and the forecasting horizon changes to 48 steps. By making this temporal aggregation, it may make it possible for better modelling of trend patterns, the main idea of [10] also encourage us to seek a more accurate trend prediction.

Moreover, instead of focusing on a single aggregation level, the use of multiple levels of aggregation, usually abbreviated as MTA (multiple temporal aggregation) can not only tackles the need to select a single aggregation level, but also partly addresses the issue of model uncertainty, and typically lead to more robust predictions.

As for the fusion of forecast results from different temporal scales, two strategies are given:

1) Fusion with naive average, i.e., directly averaging the time-series predictions from different temporal scales, and using the corresponding mean value as the final forecast output.

2) Calibrate with mean trend. As introducing temporal aggregation for forecasting makes it easier to obtain a more accurate trend information for large scale temporal forecasting, and thus an calibration with trend can be applied to the temporal forecasting results from several different temporal scales. For example, there are three temporal scales: a daily forecasting version, a hourly forecasting version and an original 10 minutes forecasting version, we will calibrate the mean trend of the hourly predictions by subtracted the gap when compared with the daily forecasting version, and then calibrate the 10 minutes predictions with the hourly results, which will be used as the final prediction. The above processing can be described as follows:

$$\Delta_{trend} = y_s.mean() - y_l$$

$$\hat{y}_s = y_s - \Delta_{trend}.repeat()$$

where y_s and y_l denote the predictions of small-scale and large-scale respectively, and $y_s.mean()$ denotes a temporal aggregation process, while $\Delta_{trend}.repeat()$ denotes the opposite process.

3 METHOD

In order to deal with the multi-turbine power prediction problem, it is essential to capture the temporal dependence among the time series and the spatial correlation among different turbines. Therefore, we propose the network architecture shown in Figure 1, which mainly consists of three parts, namely 1) mining the spatial correlation among multiple wind turbines, 2) Encoder module, which processes historical data to extract context information for supporting the subsequent Decoder module to do output prediction, and 3) Decoder module, which outputs wind turbine power prediction results based on context information and other auxiliary features.

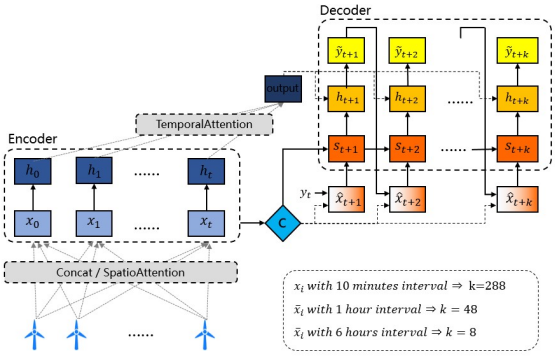

Figure 1: Network architecture

3.1 Spatial information fusion

Since there exist some spatial correlations between the historical data and future prediction results of each turbine and the surrounding turbines, the use of historical information from other turbines is beneficial to improve the power prediction results of the target turbines. This involves two questions: which turbine features are fused? And how to fuse these features? For the first question, inspired by [3], we employ a K nearest neighbor algorithm and use the information of the geographic coordinates data to find the nearest turbines, and incorporate the neighbor turbines' environmental features as an augmentation of the target turbine. Specifically, we mainly set K=8, with a naïve idea to benefit from the eight turbines surround directions. For the latter question, two methods are considered here, feature concatenation and spatio attention. Concatenation means that augment target turbines feature by augmenting neighbor features as additional features; and in the spatial attention mechanism, we employ an attention module to fusion multi turbines feature, aiming at extracting more relevant information instead of simply concatenation. By the way, the attention mechanism here can be either single-headed or multi-headed version, as proposed in [7].

3.2 Encoder module

The purpose of the Encoder module is to use the continuous historical features of the wind turbines by exploiting their intrinsic temporal dependencies and producing a context vector, which will serve as an important input for the subsequent decoder module. It is a good choice by utilizing a recurrent neural network unit, such as LSTM[11] or GRU[12], which is a widely adopted design and has been shown to be effective in many related studies, and we try it as well. However, specific to the scenario in this competition, a great challenge stems from the long forecast horizon - 288 steps; which always means that a nearly the same length or longer historical data is needed as the input, to extract an informative enough context vector for the latter Decoder module. In this case, LSTM and GRU, although better than RNN, may still decline obviously. TCN [13], which employed a dilated convolution, indicates that convolutional architectures can outperform recurrent networks on tasks, and provides a new choice for sequence modeling. Therefore, we also try to use the TCN module as an optional module of Encoder.

In mathematical form, when using LSTM or GRU as the main backbone of Encoder, the hidden vector at the last moment can be used as the context vector, which will act as an input of decoder module; when choosing TCN module, which has the same output length as its input, we simply treat the last-moment output as the context information, i.e.,

$$\hat{y_1}, \hat{y_1}, ..., \hat{y_M} = TCN(x_1, x_2, ..., x_M)$$

$$context = \hat{y_M}$$

It is worth to introduce an MLP network between Encoder and Decoder in order to coordinate the dimensionalities between them, and further enhance the nonlinear capacity of the model as well.

3.3 Decoder module

The Decoder module receives a context vector as the initial hidden and some other features as input for each moment, and introduce new hidden vectors to help make the final prediction. The GRU module would be introduced to accomplish this task.

To enhance the final prediction accuracy, we make some other designs described as follows:

1) To make fully use of the output vectors of the Encoder module, specifically, a temporal attention mechanism is configured to exploit correlations between Encoder input and the decoder hidden vector. Mathematically, it can be described as follows:

$$\hat{h}_j = Attention(h_j, outputs, outputs)$$

Where h_i is hidden of decoder at time t, and $outputs$ are from the encoder of all time slots.

2) What's more, since the context vector keeps the temporal dependence of all the historical data, highly use should be considered to improve the final prediction accuracy. And we treat it as an augmentation input of Decoder and the final forecasting module respectively, i.e.,

3.4 Model training

In this section, some details during the model training are described.

Dataset constructing. As to construct a 3-dimensional tensor dataset from a sequence of consecutive time series that can be used as input to the network architecture, like $[batch, seqlen, n\_feature]$, it is a common practice to intercept a segment of the sequence as a sample by means of a sliding window, and then sliding one step to intercept another sample. It often works well, but not always the best. In order to obtain a larger diversity, we generate samples from the original time series with an bigger stride as shown in Figure

2. Following this approach, say $stride = k$, k completely different datasets can be constructed, and resulting in k models with different performance. By fusing the prediction results of the k models, more accurate final prediction results can be gotten.

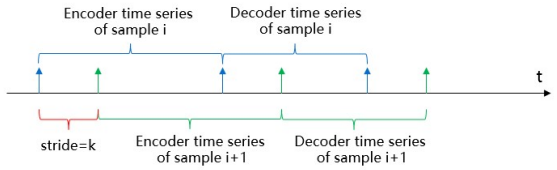

Figure 2: Constructing dataset with stride=k

Model learning. In the model training, we adopt a masked loss function to ignore the impact of irregular data, with an optional lasso regularization of the networks, and it can be described as follows:

$$loss = \frac{1}{134}\sum_{i=1}^{134}\sum_{j=1}^{288}||Patv_{ij} - \widehat{Patv}_{ij}||_r + \lambda||w||^2$$

where $||\cdot||_r$ means that we only consider the differences between predicted wind power referenced with regular real data, and $||w||^2$ is the regularization of the final output network to prevent overfitting, with λ actting as the regularisation strength.

Model ensemble. Except the above multi-model training process, we resort to some other fusion strategies. Specifically:

- Ensemble with different network configurations, e.g., spatial information fusion with concat or spatio attention; Encoder module with TCN or GRU, Decoder module with or without context vector involved in predicting the output registry. Different network configurations may be able to learn different patterns among the data and thus capture different spatio-temporal information.
- Ensemble with different machine learning methods, for example, we have tried to use random forest and lightgbm[14] to make an hourly prediction, and achieved slightly improvements in some cases.

4 EXPERIMENTAL RESULTS

In this section, we will illustrate some experimental results based on the competition dataset and draw some meaningful analytical conclusions.

4.1 Performance among different modules

In order to analyze the effectiveness of the network architecture employed in our method, model performance with different configurations are first compared. As described in the section 3.1-3.3, the configuration of this network architecture depends mainly on the choice of three parts: how spatial information is fused (concat or with spatio attention, abbreviation as CAT and SA respectively), the choice of Encoder module (LSTM/GRU/TCN, since GRU is a variation of LSTM network, which has less parameters and nearly the same principles, we only take GRU for comparison), the Decoder module configuration (whether to introduce the temporal attention mechanism, and whether to augment the context vector as a part of input for GRU module in the Decoder or the final estimating). Table 1 lists some scores of different model versions over the tested time series of 15 days (Days between 170 and 184) and metrics with toy test data, $CATn$ means concat the neighboring n turbines features, SA and TA are short for Spatio Attention and Temporal Attention, respectively.

As the results showed in Table 1, the wind power forecasting accuracy can be improved by adding a certain number of nearest neighbor turbines' features (model 1 vs model 2 and model 3), and the introduction of spatial attention mechanism behaves slightly better than the simple concat strategy. And the choice of TCN module for Encoder not always introduce benefit, but may have an average lower error. Finally, for the Decoder module, the temporal attention mechanism does not always improve the model performance, but may provide a diversity when fusion multi models' prediction results, which will be showed in the section 4.3.

4.2 Predictions with different temporal scales

In this part, we compare the prediction results under different temporal scales to analyze whether the proposed multi-scale fusion method can help improve the prediction performance. To benefit a potential diversity, we choose lightgbm and random foreset to make an hourly prediction, which is proved effective in both offline test and online submission. Taking the prediction results of TurbID=66 for days between 221 and 222 as an example, the corresponding prediction results are shown in Figure 2.

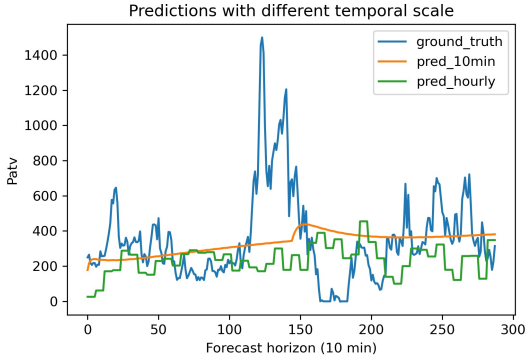

Figure 3: Predictions with different temporal scale

It is demonstrated that, as a shorter time-series prediction horizon is required when processed with temporal aggregation, the predictions behave coarse but maintain more accurate trend information. It is worth to be declared that, as showed in Figure 3, the accuracy of predictions declines dramatically as the forecast horizon goes, and it is wise to only reserve a front part of the predictions to participate in the fusion, i.e., discard the coarse latter prediction results, which may worsen the overall performance.

4.3 Model fusion results

Following the above experiments, the final prediction results by different model combinations are illustrated. Table 2 gives some of

Table 1: Scores with different model configurations

Model ID	Spatio fusion	Encoder	Decoder			Score on tested 15 days	Score on toy test data
			TA	Aug input	Aug output		
1	None	GRU	False	False	False	42.950	47.496
2	CAT 4	GRU	False	False	False	42.082	46.724
3	CAT 8	GRU	False	False	False	42.011	47.762
4	SA 8	GRU	False	False	False	43.913	42.020
5	SA 8	TCN	False	False	False	44.224	42.288
6	SA 8	TCN	True	False	False	43.455	43.735
7	SA 8	TCN	False	False	True	43.223	43.013
8	SA 8	TCN	False	True	True	43.567	46.220

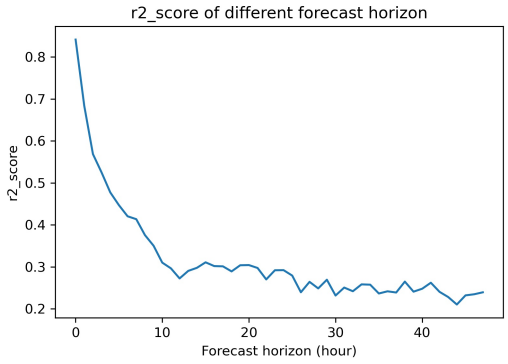

Figure 4: R2 score of different forecast horizon

the combinations on toy test data, as well as part of known results with the online stage 3 submission test.

Table 2: Model fusion results

Models	Score on toy	Score in Stage 3
Model 4 & 5	42.024	
Model 4 & 5 & 8	42.008	45.273
Model 4 & 5 & 8 & hourly	42.936	*45.237

*: the best model combination in stage 3, where hourly means predicted with a temporal scale of 1 hour.

From Table 2, we demonstrate that fusion multi models' prediction with different temporal scale may provide more accurate and robust results, even not always.

5 CONCLUSION

In this paper, we propose a multi-scale fusion method for wind power forecasting with spatiotemporal attention networks. We employ a seq2seq model to capture the temporal relevance, in which TCN module is optionally configured to extract more effective long-range dependency information, and neighbor turbines' features are argumentd with naïve concat or spatio attention strategies. What's more, to counteract the volatility and uncertainty that exist in multi-step forecasting of wind power, we also introduce the temporal aggregation mechanism to perform forecasting with different scales, and enhance the model performance slightly. Further improvement in prediction can be achieved by simply averaging or performing mean correction, which is analyzed with numerical results.

For future work, we would investigate our approach with variable temporal scale, and try to find an optimal temporal aggregation scale to make a better balance between coarse forecasting horizon and valuable trend information.

REFERENCES

[1] Y. Wang, R. Zou, F. Liu, L. Zhang, and Q. Liu. A review of wind speed and wind power forecasting with deep neural networks. *Applied Energy*, 304(1):117766, 2021.
[2] F. Petropoulos, D. Apiletti, V. Assimakopoulos, M. Z. Babai, and F. Ziel. Forecasting: theory and practice. 2020.
[3] J. Li and M. Armandpour. Deep spatio-temporal wind power forecasting. 2021.
[4] A Zn, A Zy, A Wt, A Qw, and C Mrb. Wind power forecasting using attention-based gated recurrent unit network. *Energy*, 196.
[5] X. Fu, F. Gao, J. Wu, X. Wei, and F. Duan. Spatiotemporal attention networks for wind power forecasting. 2019.
[6] Y. Hao and C. Tian. A novel two-stage forecasting model based on error factor and ensemble method for multi-step wind power forecasting. *Applied Energy*, 238(MAR.15):368–383, 2019.
[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *arXiv*, 2017.
[8] Wind power forecasting (kaggle). https://www.kaggle.com/datasets/theforcecoder/wind-power-forecasting. online; accessed 06 april 2022.
[9] Jingbo Zhou, Xinjiang Lu, Yixiong Xiao, Jiantao Su, Junfu Lyu, Yanjun Ma, and Dejing Dou. Sdwpf: A dataset for spatial dynamic wind power forecasting challenge at kdd cup 2022. *Techincal Report*, 2022.
[10] H. Wu, J. Xu, J. Wang, and M. Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. 2021.
[11] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
[12] J. Chung, C. Gulcehre, K. H. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *Eprint Arxiv*, 2014.
[13] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv:1803.01271*, 2018.
[14] M. Qi. Lightgbm: A highly efficient gradient boosting decision tree. In *Neural Information Processing Systems*, 2017.