Wind Power Forecasting with Deep Learning: Team didadida_hualahuala

Marcus Kalander*
Noah's Ark Lab, Huawei Technologies
Hong Kong, China
marcus.kalander@huawei.com

Zhongwen Rao*
Noah's Ark Lab, Huawei Technologies
Shenzhen, China
raozhongwen@huawei.com

Chengzhi Zhang
State Street Corporation
Hangzhou, China
zhangchengzhi@amss.ac.cn

ABSTRACT

Wind energy is an effective supplement to traditional energy sources. However, wind power generation is strongly weather-dependent and thus not only unsustainable in terms of production, but also highly volatile. This complex variability poses a huge challenge to the smooth operation of the grid system. In this scenario, a unique Spatial Dynamic Wind Power Forecasting dataset from Longyuan Power Group Corp. Ltd (SDWPF) is used for modern wind turbine power forecasting. In this work, we propose a framework for accurate wind power generation forecasting (WPF) based on deep learning. To obtain a more generalizable learner, we use ensemble learning to build our framework, which consists of the following two parts: a) a modified version of the DLinear model, which uses time series decomposition and linear layers for information aggregation, and b) an extreme time-gated network that adaptively captures fine-grained information and re-aggregates information by spatial location in the inference stage to obtain higher forecasting accuracy. The results indicate that our proposed combined model framework can capture long-term time series information well. Our code is available at https://github.com/shaido987/KDD_wind_power_forecast.

KEYWORDS

KDD Cup, wind power forecasting, time series, MDLinear, XTGN

ACM Reference Format:

Marcus Kalander, Zhongwen Rao, and Chengzhi Zhang. 2022. Wind Power Forecasting with Deep Learning: Team didadida_hualahuala. In *Proceedings of Baidu KDD CUP 2022 - Wind Power Forecast (KDD '22).* ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Wind power, clean and widely distributed, has been recognized as a promising renewable energy for future power generation [20]. According to the World Wind Energy Association (WWEA), the global wind industry had a capacity of 837 GW installed by the end of 2021 and 21,1 GW of offshore wind capacity was commissioned in 2021, which is an astounding level of growth with three times more than 2020. The increased penetration of wind power comes with challenges in reliably operating the electric grid system, mainly due to the high variability character of wind power generation [1, 13].

To deal with these challenges, efficient wind power forecasting has been widely considered, since accurate predictions make it possible to ensure reliable and economically efficient generation scheduling. Many studies have been devoted to investigating improved wind forecasting techniques, and most of them are based on either physical or statistical approaches. The physical WPF models draw on detailed mathematical expressions of the dynamic wind turbines/farms systems [5, 29]. Despite their advantageous physical interpretability, the physical methods often require huge amounts of computation due to their complex mathematical equations [28]. Alternatively, statistical WPF models, based on vast amounts of historical data, aim at finding the potential relationships between wind power generation and explanatory variables. Some traditional statistical methods such as ARMA, VAR, and Bayesian approaches have been modified to perform short-term wind generation forecasting [3, 14, 16]. A drawback of these approaches lies in their limited adaptability and learning capability, resulting in downgrading performance with increasing forecast horizons. More recently, deep learning WPF models have drawn attention for their powerful representation ability [18, 21, 25]. These methods aim to model mapping functions from input variables to wind power generation rather than relying on in-built physical assumptions. Despite an explosion of studies on designing deep learning WPF models, there is still considerable space for improvement in inductive biases due to the high uncertainty and high non-linear behaviour that exist in wind power generation [1, 9, 24].

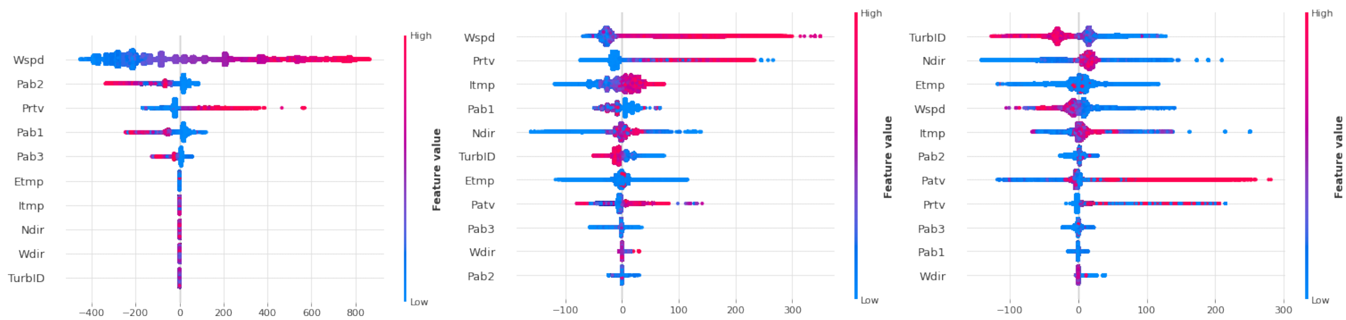
With the aim to examine the limitations of existing WPF models and exploring new advanced methods, Baidu presented a unique dataset, SDWPF, and launched the Baidu KDD Cup 2022 competition: Spatial Dynamic Wind Power Forecasting Challenge [30]. SDWPF contains comprehensive statistics on wind power generation collected from the Supervisory Control And Data Acquisition (SCADA) system of 134 wind turbines in a wind farm. For each turbine, the dynamic context factors are provided at 10 minutes intervals and the assigned task is to forecast the future 2 days (288 timesteps) of the active power output of the wind farm. Specifically, 13 features are given, consisting of 10 time series with internal status and external features such as wind speed, wind direction, temperature, nacelle direction, and three auxiliary features: day, time, and wind turbine ID. In addition to the monitoring data of each wind turbine, their relative location is provided. This allows easy usage of methods taking advantage of the spatial distribution, e.g., Graph Convolutional Networks (GCN) [8]. For a full introduction to the challenge, see the Baidu KDD Cup 2022 website[1].

*Both authors contributed equally.

[1] https://aistudio.baidu.com/aistudio/competition/detail/152/0/introduction

KDD '22, August 14–18, 2022, Washington DC, U.S.

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX



(a) Immediate forecast. **(b) One-day future forecast.** **(c) Two-day future forecast.**

Figure 1: Feature importance using SHAP values. Features at the top have a higher impact on the active power (*Patv*) while the color indicates the direction (hence best viewed in color).

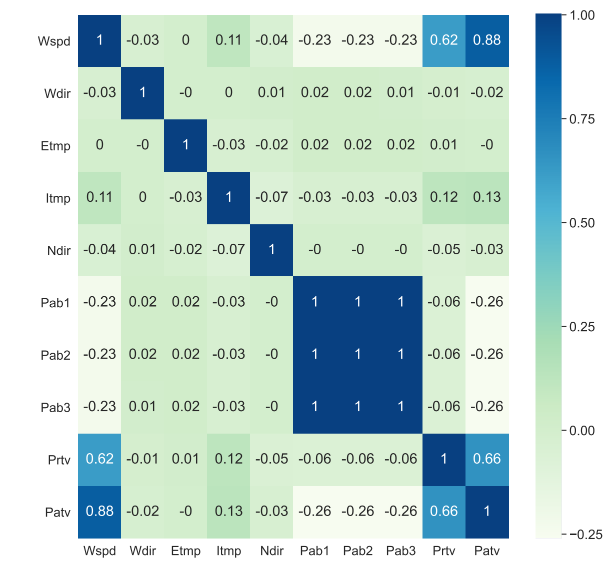


Figure 2: Feature correlation heatmap. The correlation coefficient is rounded to two decimal places.

Here we demonstrate improvements in the wind power forecasting capability. To create these more skillful predictions, we develop a data-driven approach ensembling two models, the Modified DLinear and the Extreme Temporal Gated Network. The ability of the proposed framework to learn from observational data as well as capture spatio-temporal uncertainty makes it a powerful method for operationally useful forecasting.

2 METHODOLOGY

We rely upon two separate approaches and fuse their prediction results to produce the final wind power forecast at each timestep. The first approach is an altered version of DLinear [27] (MDLinear) while the second method (XTGN) is based on Temporal Convolutional Networks (TCN) [10]. In the following, we introduce the shared data preprocessing and feature engineering steps, followed by the details of the two approaches. Finally, the fused model and its forecast merging strategy are presented.

2.1 Preprocessing and Feature Engineering

Initial Feature Analysis. There are a total of 13 features in the raw dataset, three of which are auxiliary ones: day, time, and wind turbine ID ($TurbID$). The day information is irrelevant due to how the evaluation is done (all test inputs start on day one) and is thus removed. For the time of day, it is correlated to the wind speed due to the increase in temperature during the day, however, both wind speed and outside temperature are within the provided feature set, making time of day largely redundant. It is therefore also removed.

The remaining ten features are from the internal monitoring SCADA system. The provided features are wind speed ($Wspd$), wind direction ($Wdir$), nacelle direction ($Ndir$), inside and outside temperature ($Itmp$ and $Etmp$), the pitch angle of the three blades ($Pab1$, $Pab2$, and $Pab3$), reactive power ($Prtv$), and active power ($Patv$, the target variable). We provide SHAP [15] feature importance scores in Figure 1. A key observation is that the importance of the features will change depending on the forecast horizon, for short-term predictions the wind speed is crucial, however, for predictions two days ahead, other features such as the ID (i.e., which turbine it is) are more important.

Feature Engineering. We begin by removing the wind and nacelle direction features. These two features have a minor influence on the active power while being subject to quick and abrupt changes and are consequently unreliable for longer-term forecasts. Moreover, we remove the two temperature features as their feature importance is relatively low, see Figure 1. From the analysis, the internal temperature has some positive correlation with $Patv$, however, we empirically found that removing it achieved better results. This could be due to overfitting to the training data. The feature correlations are illustrated in Figure 2. As can be observed, the three pitch angle features are perfectly correlated with each other and we thus merge these into a single feature by taking the maximum pitch angle,

$$Pab\_max = max([Pab1, Pab2, Pab3]). \quad (1)$$

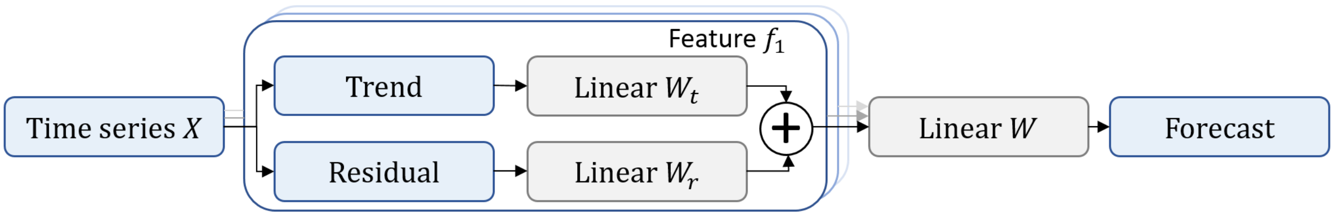


Figure 3: Overview of the MDLinear method.

For the active power *Patv*, we simply set any negative values to zero. However, our two methods differ slightly in the preprocessing of the reactive power *Prtv*, MDLinear uses its absolute value while XTGN uses the following logic

$$Prtv = \begin{cases} 0, & \text{if } Prtv < 0 \text{ and } Patv \leq 0, \\ NaN, & \text{if } Prtv < 0 \text{ and } Patv > 0, \\ Prtv, & \text{otherwise.} \end{cases} \quad (2)$$

Our final feature set is thus $\{TurbID, Wspd, Pab\_max, Prtv, Patv\}$. Any missing, unknown, or abnormal values in the raw data following the evaluation logic [30] are removed and subsequently replaced by linearly interpolated values.

2.2 Modified DLinear

We introduce our first proposed approach in this subsection, MDLinear. The method is based on DLinear [27] which is a simple but effective method that has been shown to outperform numerous transformer-based models on multiple time series forecasting tasks. By design, the method operates on univariate time series data. For multivariate time series, the forecasts of each feature are thus independent of the others. The method first decomposes the time series into trend and residual components. Two one-layer linear networks (W_t and W_r) are then applied to the respective component before the two results are merged into a final forecast output. Note that there are two variants of DLinear, one where the weights W_t and W_r are shared for all features and one where each feature learns separate weights. In our work, we use the variant with shared weights (DLinear-S).

The original DLinear method does not directly take advantage of information from supplementary features during forecasting and instead mainly relies on the historical values of the target time series. We modify DLinear to exploit all available information by adding an additional linear layer at the end to consolidate the information from all input features and denote our method MDLinear. Figure 3 illustrates the complete method design.

2.2.1 Additional Feature Engineering. In addition to the feature engineering introduced in subsection 2.1, MDLinear splits the *TurbID* feature into *cluster* and *ID*. The clusters are constructed by adhering to the wind turbines' x-coordinates, while the *ID* is assigned based on the y-coordinates. Within each cluster, the wind turbine with the largest y-coordinate is assigned an *ID* of 1, and the one with the lowest y-coordinate is assigned n_c, where n_c is the number of wind turbines within the cluster. Figure 4 shows the cluster and ID assignments for all 134 wind turbines.

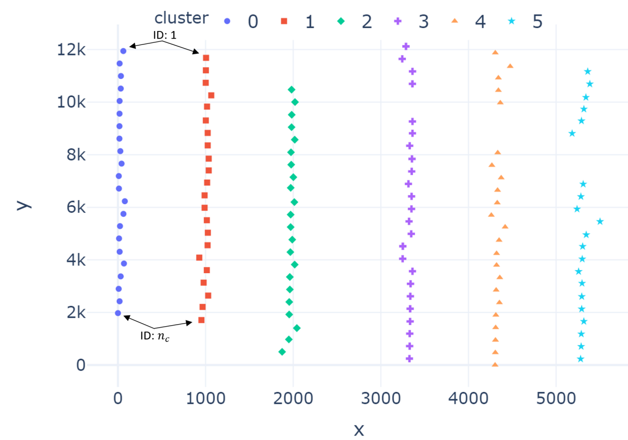


Figure 4: Wind turbine cluster and ID assignments.

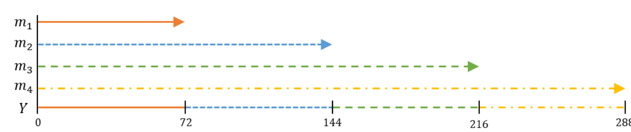


Figure 5: MDLinear train and forecast strategy.

We further create a new feature *cluster_avg* using the assigned clusters. For cluster c with a set of wind turbines C, we have

$$cluster\_avg_c = \frac{1}{n_c} \sum_{i=1}^{n_c} Patv^{C(i)}, \quad (3)$$

where n_c is the number of wind turbines in c and $Patv^{C(i)}$ the wind power of the i-th wind turbine in C. Note that *cluster_avg* has the same value for all wind turbines within the same cluster (for the same timestep).

2.2.2 Multi-horizon Training and Forecasting. Training a single model to make future forecasts is not always advantageous. Short-term and long-term patterns can vastly differ and a model that focuses on the full horizon may fail to capture important short-term patterns. Therefore, in MDLinear, we train four separate models with increasing forecast horizons [72, 188, 216, 288] and merge these to create a final prediction. The last predicted 72 timesteps of each model m are used. Figure 5 illustrates the procedure.

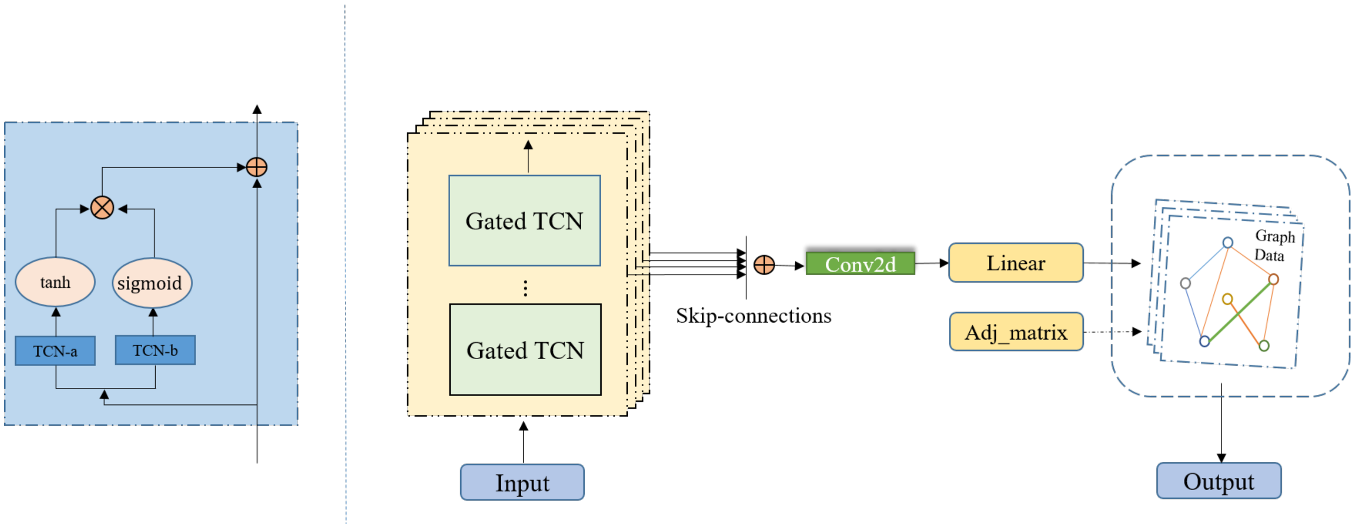


Figure 6: The framework of XTGN. The inputs are first transformed by the gated temporal convolution module (Gated TCN, detailed in the left figure) followed by a 2D convolutional layer and a linear layer. An information diffusion mechanism (shown in the dashed box) is performed only during the inference phase to get a reliable wind power prediction.

2.3 Extreme Temporal Gated Network

Inspired by the idea of converting a forecasting problem into a pattern-matching task [11, 12], we consider a robust learning framework, extreme Temporal Gated Network (XTGN), to extract implicit representative patterns in the observational data. We present the proposed method in Figure 6 and elaborate on it in the following.

The dynamical chaotic grid system produces complicated features with sudden jumps and uncertain wind power generation with strong randomness [23, 30], making it difficult to generate accurate forecasts. To quickly respond to varying wind power patterns and abruptly changes caused by external reasons such as wind turbine renovation or active power controlling, we apply a Temporal Convolutional Network (TCN) [10] based framework to acquire the necessary information on existing representative patterns and then generalize it for forecasting.

Although RNN-based approaches have been proven to be able to process sequential data, the recursive manner still suffers from defects such as vanishing gradient and being time-consuming. On the contrary, the stacked dilated causal convolution layers [26] allow TCN to achieve long temporal range receptive fields and alleviate the vanishing gradient problem. Mathematically, the temporal representation at layer i is calculated as,

$$x * W = \sum_{s=0}^{K-1} W(s)x(i - d \times s), \tag{4}$$

where $*$ denotes a convolution operator, $x \in \mathbf{R}^M$ is a 1D sequence input, $W \in \mathbf{R}^K$ is the learnable convolution filter and d is a hyperparameter that controls the skipping sparsity.

The gating mechanism was introduced in TCN to allow better information flow through layers [2], and it takes the form

$$h(x) = \tanh(x * \theta_1 + b) \odot \sigma(x * \theta_2 + c), \tag{5}$$

where $\odot$ is the element-wise product, and the tangent hyperbolic function $\tanh(\cdot)$ and the sigmoid function $\sigma(\cdot)$ are used to control the ratio of information to forget or allow into the next layer. To generate probable varying patterns that are directly related to the downstream forecasting task, the output of the Gated TCN layer is then passed to a 2D convolutional layer followed by a linear operator.

Moreover, an information propagation scheme is used during the inference phase to mitigate the negative effects of atypical data on the prediction error. To be specific, the obtained values from the last linear layer are corrected by a neighborhood aggregation to develop a more accurate prediction. Considering the observation that nearby wind turbines exhibit similar wind power patterns, it is natural to describe the underlying graph structure of the system using a distance matrix. Here, we generate a k nearest neighbor (k-NN) graph based on the cosine similarity [22] between the geographic location gl for each wind turbine pair (i, j),

$$S_{ij} = \frac{gl_i^T gl_j}{\|gl_i\|\|gl_j\|}. \tag{6}$$

For each wind turbine, the k nearest neighbors following the above cosine similarity are chosen to construct the k-NN graph. The adjacency matrix is denoted as A. Then we perform information diffusion over the graph to merge information from the node neighborhood to get the final forecasts. By considering the neighbors' state, the predictions can respond effectively to some irregular congestion and abruptly changing patterns. Despite there being various information diffusion mechanisms [4, 8], we aggregate information from only first-order neighbors to ensure high computational efficiency. In this way, we get a reliable wind power prediction,

$$Y^t = A(W_o \times \text{Conv2D}(h(x)) + b_o) \tag{7}$$

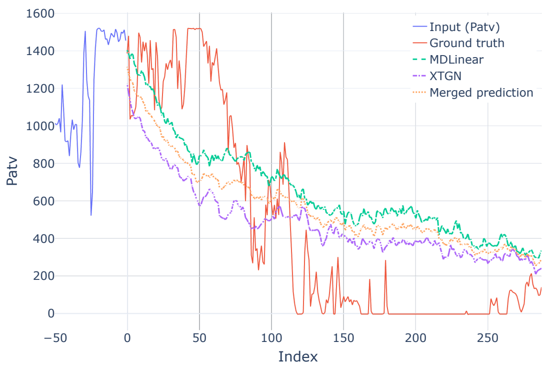


Figure 7: A single prediction result for a wind turbine.

where W_o and b_o are learnable parameters in the linear layer. Note that other forms of diffusion process can be easily fitted into our framework, such as Random walk [17, 19].

2.4 Fused Model

We consolidate the predictions from MDLinear and XTGN into a fused forecast. We denote the predictions for a single wind turbine at timestep t as $Y^m \in \mathbb{R}^{288\times1}$ and $Y^t \in \mathbb{R}^{288\times1}$ for MDLinear and XTGN, respectively. We empirically found that a simple averaging of the two forecasts at each timestep achieves robust results, see Section 3.2. The fused forecast for each wind turbine is thus

$$Y = \frac{Y^m + Y^t}{2}. \tag{8}$$

The process is illustrated in Figure 7 which depicts a typical 288-length prediction for a single timestep of a single wind turbine.

3 EXPERIMENTS

In this section, we present some results using the small provided test sample containing 14 days of data. We run a sliding window of length 288 and a step size of 10 to create a set of 145 forecast instances, each with a historical window of length 288. All scores in this section are from this offline test data.

3.1 Experimental Setup

3.1.1 Parameter Settings. In MDLinear, the length of the historical time window used as input is set to 50 and the time series decomposition is done by learning the trend using a moving average with a window size of 25. Four different models are trained using the multi-horizon strategy described in Section 2.2.2.

XTGN is directly trained for two days-ahead forecasting and uses two days as the historical time window, a.k.a. 288 consecutive historical data points ($M = 288$) are used to forecast wind power generation in the next two days ($T = 1, 2, \cdots, 288$). Note that the input feature set is different from MDLinear model and only four features are used: $\{Wspd, Pab\_max, Prtv, Patv\}$. To explicitly capture fine-grained temporal embedding from the input sequence, we apply 2 blocks in which each block contains 7 layers of Gated TCN with dilation factors $[1, 2, 4, ..., 64]$, respectively. In the inference phase, we utilize the cosine similarity function to construct an adjacency matrix A based on the $k = 50$ nearest neighbors.

For both methods, all features are z-score normalized based on the mean and standard deviation of the training data to ensure a stable training procedure. The first 90% of the data for each wind turbine is used as training data while the remaining 10% is used for validation. We apply a masked loss function where any missing, unknown, or abnormal values are masked away following the evaluation scheme logic [30]. MDLinear uses the average of RMSE and MAE while XTGN utilizes MAE loss.

The models are trained for a maximum of 100 epochs with a batch size of 32 and make use of early stopping with a patience of 3 to avoid overfitting. The Adam optimizer [7] is used for training, with an initial learning rate at 0.001 for MDLinear and 0.003 for XTGN, both with a weight decay of 0.0001. MDLinear further applies an adaptive learning rate strategy where the learning rate gets divided by 10 after 10 epochs.

3.1.2 Evaluation Metrics. For wind turbine i at time t_0, the RMSE and MAE are computed as

$$RMSE^i_{t_0} = \frac{1}{288}\sqrt{\sum_{t=1}^{288}(Y_{t_0+t} - \hat{Y}_{t_0+t})^2}, \tag{9}$$

$$MAE^i_{t_0} = \frac{1}{288}\sum_{t=1}^{288}|Y_{t_0+t} - \hat{Y}_{t_0+t}|, \tag{10}$$

where Y_{t_0+t} and $\hat{Y}_{t_0+t}$ is respectively the predicted and true power output at time $t_0 + t$ with $t \in [1, 288]$. Note that any values that are missing, unknown, or abnormal in the original data are ignored in the computation of RMSE and MAE by setting $Y_{t_0+t} - \hat{Y}_{t_0+t} = 0$. The final score is the average of RMSE and MAE over all wind turbines

$$score = \frac{1}{2}\sum_{k=0}^{K}\sum_{i=1}^{134} RMSE^i_{T(k)} + MAE^i_{T(k)}, \tag{11}$$

where K is the number of instances to evaluate the model with and T a set of K timestamps to predict.

3.1.3 Baselines for Comparison. In addition to our proposed fused model and its two component methods, we provide results on additional baselines. First, we include a naive baseline predicting the average train *Patv* value for each wind turbine and a baseline using a moving average with a window size of 288. Note that both these baselines predicts identical values for the whole 288 length forecast window for a single timestep t. LightGBM [6] is also added as a baseline as it has shown strong results in various prediction tasks. Finally, we also include the provided GRU and GNN baselines [30].

3.2 Results and Analysis

The offline scores of the baselines and our proposed method as well as its component methods are presented in Table 1. The inference times shown are for running the full test set. For reference, MDLinear using a single model with a forecast horizon of 288 is included which performs significantly worse than the complete method. From the table, we can observe that the offline scores of the fused model are worse than some of the baselines, however,

Table 1: Offline scores and inference times for the methods.

Method	RMSE	MAE	Score	Time (s)
Historical average	56.72	47.86	52.29	121
Moving average	61.56	50.62	56.09	127
GRU	55.13	45.77	50.45	409
GNN	55.39	47.15	51.27	245
LightGBM	53.05	44.89	48.97	4,035
MDLinear (single model)	56.74	48.32	52.53	4,63
MDLinear	53.40	45.53	49.46	1,384
XTGN	54.54	46.50	50.52	227
Fused model	53.74	45.86	49.80	1,420

Table 2: Different fusion strategies. For the time splits, the method in parentheses is used for the first timesteps.

Method	RMSE	MAE	Score
Time split 72:216 (MDLinear)	54.24	46.09	50.17
Time split 72:216 (XTGN)	53.74	45.94	49.84
Time split 188:188 (MDLinear)	53.95	45.90	49.92
Time split 188:188 (XTGN)	54.02	46.13	50.07
Average	53.74	45.86	49.80

the fused model is more robust and thus performed the best on the online evaluation.

How the merging is done in the fused model is critical to the final model performance. An assessment of various methods is presented in Table 2. In addition to using the average of the forecasts at each timestep, we evaluate using the prediction of one of the models for the initial segment and the other model for the remaining timesteps. As shown in the table, using the average forecast values achieves the best performance of the evaluated strategies.

4 CONCLUSION

We introduce an ensemble framework for wind power forecasting by combining two effective approaches. One considers the problem using a linear model MDLinear, and the other model XTGN converts the forecasting problem as a pattern-matching task. The MDLinear model is designed based on time series decomposition, which encourages the model to incorporate temporal dynamics in the predictions. In the XTGN model, we use a TCN-based module to explore implicit representative patterns in the observational data and design an information diffusion mechanism to create accurate predictions. Our method achieves excellent performance in the Baidu KDD Cup 2022 challenge and appears to get a reliable robust wind power forecast in dealing with atypical data.

REFERENCES

[1] Wen-Yeau Chang et al. 2014. A literature review of wind forecasting methods. *Journal of Power and Energy Engineering* 2, 04 (2014), 161.

[2] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *International conference on machine learning*. PMLR, 933–941.

[3] Jethro Dowell and Pierre Pinson. 2015. Very-short-term probabilistic wind power forecasts by sparse vector autoregression. *IEEE Transactions on Smart Grid* 7, 2 (2015), 763–770.

[4] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[5] Jaesung Jung and Robert P Broadwater. 2014. Current status and future advances for wind speed and power forecasting. *Renewable and Sustainable Energy Reviews* 31 (2014), 762–777.

[6] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017).

[7] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[8] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[9] Adam Kisvari, Zi Lin, and Xiaolei Liu. 2021. Wind power forecasting–A data-driven method along with gated recurrent neural network. *Renewable Energy* 163 (2021), 1895–1909.

[10] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. 2017. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 156–165.

[11] Chunggi Lee, Yeonjun Kim, Seungmin Jin, Dongmin Kim, Ross Maciejewski, David Ebert, and Sungahn Ko. 2019. A visual analytics system for exploring, monitoring, and forecasting road traffic congestion. *IEEE transactions on visualization and computer graphics* 26, 11 (2019), 3133–3146.

[12] H. Lee, S. Jin, H. Chu, H. Lim, and S. Ko. 2021. Learning to Remember Patterns: Pattern Matching Memory Networks for Traffic Forecasting. (2021).

[13] Ma Lei, Luan Shiyan, Jiang Chuanwen, Liu Hongling, and Zhang Yan. 2009. A review on the forecasting of wind speed and generated power. *Renewable and sustainable energy reviews* 13, 4 (2009), 915–920.

[14] Li Ling-ling, Jun-Hao Li, Peng-Ju He, and Cheng-Shang Wang. 2011. The use of wavelet theory and ARMA model in wind speed prediction. In *2011 1st International Conference on Electric Power Equipment-Switching Technology*. IEEE, 395–398.

[15] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).

[16] Jakob W Messner and Pierre Pinson. 2019. Online adaptive lasso estimation in vector autoregressive models for high dimensional wind power forecasting. *International Journal of Forecasting* 35, 4 (2019), 1485–1498.

[17] GH Mowbray and MV Rhoades. 1959. On the reduction of choice reaction times with practice. *Quarterly Journal of Experimental Psychology* 11, 1 (1959), 16–23.

[18] Zhewen Niu, Zeyuan Yu, Wenhu Tang, Qinghua Wu, and Marek Reformat. 2020. Wind power forecasting using attention-based gated recurrent unit network. *Energy* 196 (2020), 117081.

[19] Karl Pearson. 1905. The problem of the random walk. *Nature* 72, 1865 (1905), 294–294.

[20] Pierre Pinson, Christophe Chevallier, and George N Kariniotakis. 2007. Trading wind generation from short-term probabilistic forecasts of wind power. *IEEE Transactions on Power Systems* 22, 3 (2007), 1148–1156.

[21] Aqsa Saeed Qureshi, Asifullah Khan, Aneela Zameer, and Anila Usman. 2017. Wind power prediction using deep neural network based meta regression and transfer learning. *Applied Soft Computing* 58 (2017), 742–755.

[22] Stijn Van Dongen and Anton J Enright. 2012. Metric distances derived from cosine similarity and Pearson and Spearman correlations. *arXiv preprint arXiv:1208.3145* (2012).

[23] Shijun Wang, Chun Liu, Kui Liang, Ziyun Cheng, Xue Kong, and Shuang Gao. 2022. Wind Speed Prediction Model Based on Improved VMD and Sudden Change of Wind Speed. *Sustainability* 14, 14 (2022), 8705.

[24] Yun Wang, Houhua Xu, Runmin Zou, Lingjun Zhang, and Fan Zhang. 2022. A deep asymmetric Laplace neural network for deterministic and probabilistic wind power forecasting. *Renewable Energy* (2022).

[25] Yun Wang, Runmin Zou, Fang Liu, Lingjun Zhang, and Qianyi Liu. 2021. A review of wind speed and wind power forecasting with deep neural networks. *Applied Energy* 304 (2021), 117766.

[26] Fisher Yu and Vladlen Koltun. 2015. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122* (2015).

[27] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2022. Are Transformers Effective for Time Series Forecasting? *arXiv preprint arXiv:2205.13504* (2022).

[28] Jinhua Zhang, Jie Yan, David Infield, Yongqian Liu, and Fue-sang Lien. 2019. Short-term forecasting and uncertainty analysis of wind turbine power based on long short-term memory network and Gaussian mixture model. *Applied Energy* 241 (2019), 229–244.

[29] Xin Zhao, Shuangxin Wang, and Tao Li. 2011. Review of evaluation criteria and main methods of wind power forecasting. *Energy Procedia* 12 (2011), 761–769.

[30] 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).