

AutoWP: Automated wind power forecasts with limited computing resources using an ensemble of diverse wind power curves

Stefan Meisenbacher¹, Silas Aaron Selzer^{2,3}, Mehdi Dado¹,
Maximilian Beichter¹, Tim Martin¹, Markus Zdrallek², Peter
Bretschneider^{3,4}, Veit Hagenmeyer¹, Ralf Mikut¹

E-Mail: ralf.mikut@kit.edu, ¹Karlsruhe Institute of Technology, Germany

²University of Wuppertal, Germany ³Ilmenau University of Technology, Germany

⁴Fraunhofer IOSB, Germany

Abstract

Forecasting the locally distributed Wind Power (WP) generation is crucial for future energy systems, demanding scalable WP forecasting models to keep pace with the increasing number of smart grid applications. Therefore, we propose AutoWP, which is a weighted ensemble of WP curves that represent different site conditions. This representation is achieved by using a diverse set of WP curves from Original Equipment Manufacturers (OEMs) and optimizing their contribution to the weighted sum of curves using the least squares method. AutoWP is advantageous since physical limitations in WP generation are implicitly reflected in the considered WP curves, and the optimization to find the ensemble weights requires only a small amount of data and computational effort. Furthermore, AutoWP uses a rule-based and filter-based approach for training data cleaning to identify and filter samples representing abnormal operation due to shutdowns or partial load operation. It is shown on a real-world data set that AutoWP achieves competitive day-ahead forecasting performance against other methods based on WP curve modeling and outperforms the autoregressive Deep Learning (DL) method Temporal Fusion Transformer (TFT), which is based on the attention mechanism.

1 Introduction

The increasing generation capacity of decentral Wind Power (WP) turbines and PhotoVoltaic (PV) plants with weather-dependent and fluctuating generation poses new challenges for the stability of power grids, especially for frequency stability. However, grid bottlenecks occur not only at the transmission grid level but also at the distribution grid level, making smart grid applications indispensable in the future energy system, e. g., demand response to reduce peak-valley load differences [1], battery scheduling to optimize WP dispatch planning [2], and redispatch planning for predictive congestion management [3]. Since many smart grid applications require forecasts of decentralized WP generation, it is necessary to automate the design and operation of WP forecasting models to keep pace with the growing number of applications and the expansion of generation capacity. On the one hand, models achieving low forecasting errors are required that consider site-specific conditions like the wind direction and the mutual influence of the WP turbines (wake losses) in offshore wind farms. Because offshore wind turbines are established in a uniform terrain, the WP farm can be modeled holistically like in [4] using an autoregressive Deep Learning (DL) model. On the other hand, scalable forecasting models are required for decentrally located onshore WP turbines, where the terrain is heterogeneous (open fields, urban areas, hills, and forests). In such a case, the use of DL models is limited since they require a high computational effort for training the model to represent site-specific air turbulence caused by the terrain. Even though so-called WP curves are available for most turbines, which establish an empirical relationship between wind speed and WP generation, the WP curves provided by the Original Equipment Manufacturer (OEM) only apply to the site-specific conditions in which they were recorded.

Related work on site-specific WP curve modeling comprise parametric and non-parametric modeling approaches [5]. While the former is based on fitting an assumed mathematical expression [6] or using physical-inspired models [7], the latter comprises methods based on Machine Learning (ML), e. g., Support Vector Regression (SVR) [8–10], the MultiLayer Perceptron (MLP) [7, 9, 11, 12], and Decision Tree (DT)-based methods [8, 10, 13]. ML-based methods can consider further explanatory variables such as the atmospheric

pressure [14], the air temperature [12, 14] and the wind direction [11–13]. A critical challenge, however, is inconsistent training data due to shutdowns or partial load operation. Such inconsistencies can be classified into i) steady WP generation at a power below the peak power rating of the turbine, ii) no WP generation while above cut-in wind speed, and iii) WP generation during stop-to-operation transitions and vice versa [15]. Modeling normal operation, thus, requires data cleaning as proposed in the literature by labeling normal operation based on expert knowledge [10] or automating data cleaning by removing samples that significantly differ from an already fitted WP curve [9], or where the power generation is lower than a threshold near zero [9, 12]. The challenge of identifying inconsistent data is also present for autoregressive methods applied to WP forecasting. Such methods consider a horizon of past WP generation values to make a forecast. Most state-of-the-art autoregressive methods are based on DL architectures like the attention-based transformer architecture [16–20]. While complex relationships can be learned by such methods, their scalability is limited due to the high computational training effort.

Therefore, the present paper proposes the new method called AutoWP for day-ahead WP forecasting, which automates data pre-processing and is scalable by requiring low computational training effort while achieving competitive forecasting performance. The automated pre-processing uses a rule-based and an outlier detection-based approach, and AutoWP’s model design relies on an ensemble of diverse WP curves that is fitted to the site-specific conditions of the new WP turbine. Specifically, a diverse ensemble of OEM WP curves from the Open Energy Platform (OEP) wind turbine library [21] is created, and the optimal contribution of these curves to the weighted sum of curves is determined using the least squares method.

The remainder of this paper is organized as follows. We detail the methodological concept of AutoWP in Section 2, evaluate AutoWP in Section 3, discuss the results in Section 4, and provide a conclusion and outlook in Section 5.

2 AutoWP

The core of AutoWP is to represent a new WP turbine as an optimally weighted ensemble of WP curves, similar to the method AutoPV [22, 23].¹ We first introduce the WP curve before describing the three-step ensemble method.

The WP curve $y[k] = f(v_{\text{eff}}[k])$ is a function of the power generation $y[k]$ depending on the wind speed at a reference height $v_{\text{eff}}[k]$ (commonly the hub height). Because wind speed forecasts are usually available at 10m and 100m, forecasting using WP curves requires a height correction² to account for different wind speeds at different heights due to the atmospheric boundary layer [24]. For AutoWP, we perform the height correction of the wind speed forecast based on the straightforward wind power law

$$\frac{v_a}{v_b} = \left(\frac{h_a}{h_b} \right)^{\alpha_h}, \quad (1)$$

with v_a and v_b being the wind speeds at heights h_a and h_b above ground level, and α_h depends on the terrain [25].³ Using (1) allows to estimate the effective hub height

$$h_{\text{eff}} = 100 \text{ m} \cdot \left(\frac{\bar{v}_{\text{eff}}}{\bar{v}_{100}} \right)^{(1/\alpha_h)} \quad (2)$$

from the averages of the 100m wind speed forecast \bar{v}_{100} and the hub height wind speed measurement \bar{v}_{eff} over the samples in the training data sub-set. Then, wind speed forecasts $\hat{v}_{100}[k]$ can be height-corrected using

$$\hat{v}_{\text{eff}}[k] = \hat{v}_{100}[k] \cdot \left(\frac{h_{\text{eff}}}{100 \text{ m}} \right)^{\alpha_h}, \quad (3)$$

and used as inputs of the WP curve to make a forecast.

¹ AutoPV also uses an optimally weighted ensemble to represent new PV plants with differing site conditions and unknown mounting configurations (tilt and azimuth angles), as well as mixed-oriented PV plants.

² Except if the turbine's hub height is coincidentally 100m. If the wind speed forecast is available at a different height, for example 80m, the height in (2) and (3) must be adjusted accordingly.

³ For onshore and offshore WP turbines, typical estimates for α_h are 1/7, respectively 1/9 [26].

The three-step method AutoWP entails i) creating the ensemble, ii) computing the ensemble output by the optimally weighted sum, and iii) re-scaling the ensemble output, exemplified in Figure 1, and detailed in the following.

The first step creates the WP curve ensemble using OEM curves from the OEP wind turbine library [21]. Specifically, all WP curves are resampled to ensure a uniform sampling rate of the wind speed v_{eff} , and normalized with the turbine's peak power rating $P_{\max,n}$:

$$P_n^*[v] = \frac{P_n[v]}{P_{\max,n}}. \quad (4)$$

In order to limit the ensemble size without losing diversity, we sort all normalized WP curves according to the area under the curve, retaining the first and last WP curves and eight WP curves equally spaced in between, i. e., $N_m = 10$.

The second step computes the ensemble WP curve based on the weighted sum of $N_m = 10$ WP curves

$$\hat{y}^*[k] = \sum_{n=1}^N \hat{w}_n \cdot \hat{y}_n^*[k], \quad (5)$$

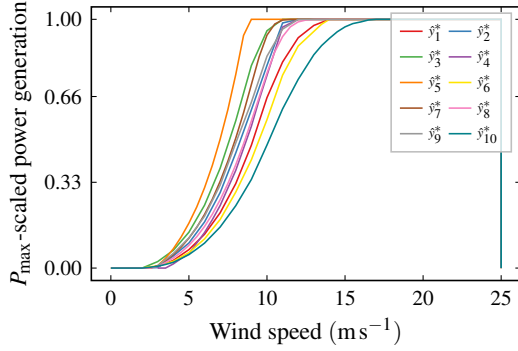
with the weight \hat{w}_n , $n \in \mathbb{N}_1^N$ and the output \hat{y}_n of the n -th normalized WP curve. The optimal weights are determined by minimizing the mean squared error

$$\min_{\hat{\mathbf{w}}} \frac{1}{K} \sum_{k=1}^K (\hat{y}^*[k] - y^*[k])^2 \quad \text{s.t.} \quad \hat{\mathbf{w}} \in [0, 1], \sum_{n=1}^N \hat{w}_n = 1, \quad (6)$$

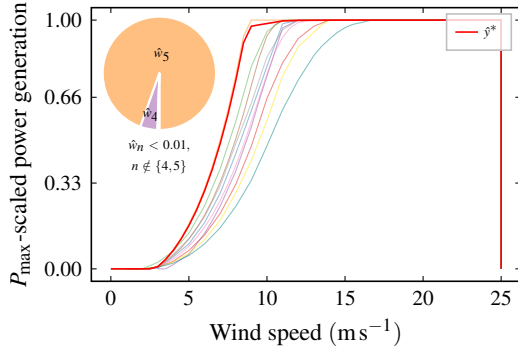
between ensemble output \hat{y}^* (5) and the target time series y^* , which is normalized analogously to (4).⁴ We solve the optimization problem (6) with the least squares algorithm of the Python package SciPy [27] and normalize the weights to ensure a convex linear combination.⁵

⁴ $\hat{w}_n = 1/N$, $\forall n \in [1, N]$ can be used for initialization if no data is available (cold-start problem), to be adapted during operation as measurements become available, as shown for AutoPV [22, 23].

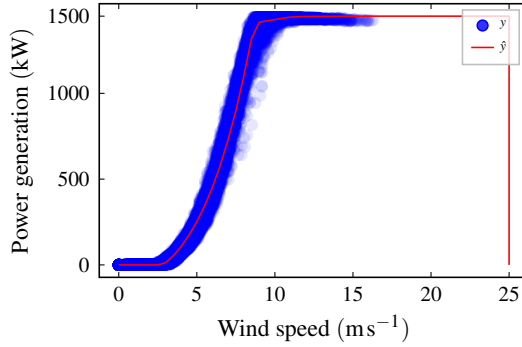
⁵ Alternatively, the constraints in (6) could be relaxed, i. e., weights greater than one are accepted, and the sum of the weights does not have to be one. Consequently, the re-scaling in (7) is obsolete as the sum of weights represent the new WP turbine's peak power rating, making AutoWP also suitable for WP turbines with unknown peak power ratings.



(a) The first step creates the ensemble pool using $N_m = 10$ normalized WP curves \hat{y}^* , $n \in \mathbb{N}_1^m$ of the OEP wind turbine library [21]. The selection reduces redundancy and preserves diversity.



(b) The second step forms the normalized ensemble WP curve \hat{y}^* as convex linear combination of the pool's curves, with the weights \hat{w}_n , $n \in \mathbb{N}_1^m$ adapted to optimally fit the new WP turbine.



(c) The third step re-scales the ensemble WP curve \hat{y} to the peak power rating of the new WP turbine, in this example $P_{\max, \text{new}} = 1500 \text{ kW}$. Additionally, the measurement \hat{y} used to fit the ensemble weights is shown.

Figure 1: The three steps of AutoWP's automated design exemplified for the real-world WP turbine no. 2.

The third step re-scales the ensemble output (5) using the new WP turbine's peak power rating $P_{\max, \text{new}}$:

$$\hat{y}[k] = \hat{y}^*[k] \cdot P_{\max, \text{new}}. \quad (7)$$

3 Evaluation

In this section, we benchmark AutoWP against the OEM WP curve, ML-based WP curve modeling methods, and an autoregressive DL method.

Data We evaluate AutoWP on a real-world data set of the quarter-hourly energy metering (kWh) in 2019 and 2020 from two WP turbines with a peak power rating of 1500kWp.⁶ This data set is useful for the evaluation, as the WP turbines are subject to unique shutdown patterns, as shown below. The energy metering time series of both WP turbines are transformed into mean power generation time series

$$\bar{P}[k] = \frac{\Delta E[k]}{t_k}, \quad (8)$$

to increase interpretability, with the energy generation $\Delta E[k]$ (kWh) metered within the sample period t_k (h). Additionally, average wind speed measurements at hub height, as well as day-ahead weather forecasting data from the European Centre for Medium-Range Weather Forecasts (ECMWF) [28] with the origin at 00:00 are available. We split the data into a training data sub-set (2019) and a test data sub-set (2020).⁷

Both WP turbines are located in southern Germany and are subject to many shutdowns. Abnormal operating states in the data are identified according to [9] and [15]: All data points with a wind speed measurement greater than the cut-in speed and a power generation lower than the cut-in power are considered

⁶ Dynamics below the 15-minute resolution like stop-to-operation transitions and vice versa, as well as wind gusts, cannot be considered.

⁷ The forecasting method TFT additionally holds-out validation data of 20% of the training data to terminate training if the validation loss increases (early stopping).

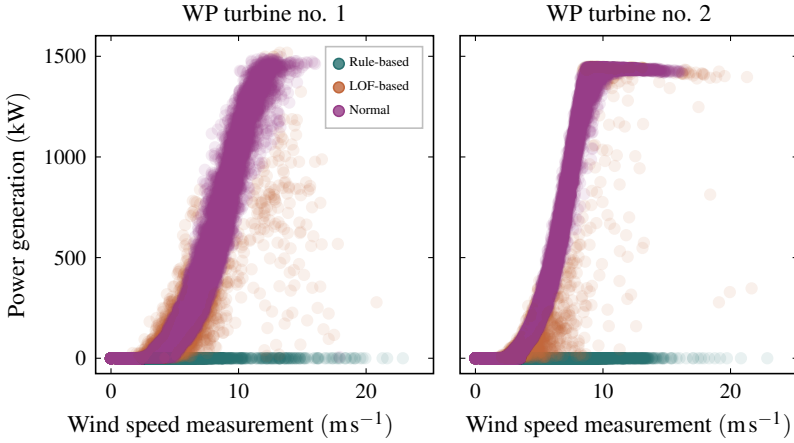


Figure 2: Data pre-processing to identify abnormal operational states. Rule-based filtering identifies turbine shutdowns and LOF-based filtering identifies turbine stop-to-operation transitions and vice versa. The samples for which the wind speed at hub height is below the cut-in speed $v_{\text{cut-in}} = 2.5 \text{ m s}^{-1}$ amount to 16.7%, while the cut-out speed $v_{\text{cut-out}} = 25 \text{ m s}^{-1}$ at which the WP turbine would be shut down for safety reasons is not reached.

as shutdowns [9], and the Local Outlier Factor (LOF) algorithm is applied to identify outliers due to WP turbine stop-to-operation transitions and vice versa [15], see Figure 2. Since we aim to model the normal operating state of the WP turbine, abnormal operating states are dropped from the training data sub-set.

Benchmarks We compare the forecasting error of AutoWP against the OEM WP curve, ML-based WP curve modeling methods, and an autoregressive DL method. Regarding WP curve modeling, we consider established non-linear regression methods to directly learn the weather-dependent WP curve using the explanatory variables air temperature and atmospheric pressure, as well as wind speed and direction. Specifically, we consider the established methods eXtreme Gradient Boosting (XGB), SVR, and the MLP that work well in a variety of regression tasks. Regarding autoregressive DL-based forecasting, we consider the TFT [29]. This method combines recurrent Long Short-Term Memory (LSTM) layers with the attention-based transformer architecture. Recurrent

layers allow the output from neurons to affect the subsequent input to the same neuron, which is beneficial for capturing short-term dependencies. Attention units compute importance scores for each element in the input sequence, which is useful for capturing long-term dependencies. For the day-ahead forecast with the origin at 00:00, we consider past values up to one day (96 values), and covariate time series of air temperature and atmospheric pressure, as well as wind speed and direction.

The TFT is implemented with the Python package PyTorch Forecasting [30], SVR and MLP with the Python package scikit-learn [31], and XGB with the authors' [32] Python package. For all methods, we use the default hyperparameter configuration of the respective Python package, and methods that use a stochastic training algorithm (TFT and MLP) are run five times.

Evaluation strategy In the evaluation, we assume knowing the times at which the WP turbine will be shut down, which is reasonable for day-ahead forecasts, as the turbine operator should be aware of regular shutdowns like maintenance work or bat protection. That is, we calculate assessment metrics only for samples identified as normal operation. As assessment metrics, we consider the normalized Mean Absolute Error (nMAE)

$$\text{nMAE} = \frac{\sum_{k=1}^K |\hat{y}[k] - y[k]|}{\sum_{k=1}^K y[k]}, \quad (9)$$

and the normalized Root Mean Squared Error (nRMSE)

$$\text{nRMSE} = \frac{\sqrt{\frac{1}{K} \sum_{k=1}^K (\hat{y}[k] - y[k])^2}}{\frac{1}{K} \sum_{k=1}^K y[k]}, \quad (10)$$

where $\hat{y}[k]$ is the forecast and $y[k]$ the realized value at time point k .

Results Table 1 shows the results on the test sub-set of the evaluation data. We observe that the OEM WP curve achieves comparably higher forecasting errors for WP turbine no. 1. Among methods based on WP curve modeling, AutoWP, MLP, and SVR achieve a similar forecasting error. While AutoWP achieves

Table 1: The forecasting errors on the test data sub-set of AutoWP compared to other methods based on WP curve modeling (OEM curve, MLP, SVR, XGB), and to the autoregressive DL method TFT.

Turbine no.	Error	AutoWP	OEM curve	MLP	SVR	XGB	TFT
1	nMAE	0.69 ± 0.00	0.94 ± 0.00	0.83 ± 0.01	0.70 ± 0.00	0.84 ± 0.00	0.74 ± 0.00
2		0.61 ± 0.00	0.67 ± 0.00	0.69 ± 0.01	0.64 ± 0.00	0.74 ± 0.00	0.67 ± 0.00
1	nRMSE	1.2 ± 0.00	1.57 ± 0.00	1.15 ± 0.01	1.11 ± 0.00	1.22 ± 0.00	1.26 ± 0.00
2		0.93 ± 0.00	1.07 ± 0.00	0.92 ± 0.00	0.93 ± 0.00	1.01 ± 0.00	0.98 ± 0.00

the lowest nMAE for both turbines, the SVR achieves the lowest nRMSE for WP turbine no. 1, and the MLP for no. 2. The autoregressive DL method TFT achieves competitive performance but requires a high computational training effort, see Table 2.⁸ In contrast, the WP curve modeling approaches require substantially less training time. Since the OEM WP curve only requires height correction, its training effort is the lowest. For XGB and AutoWP, the training time is still less than one second, whereas SVR and MLP are in the region of half a minute, and the TFT already requires well over an hour for training.

Table 2: The computational training effort in seconds of AutoWP compared to other methods based on WP curve modeling (OEM curve, MLP, SVR, XGB), and to the autoregressive DL method TFT. For the OEM curve, the effort refers to the computing time for the height correction.

AutoWP	OEM curve	MLP	SVR	XGB	TFT
0.6 ± 0.3	0.001 ± 0.000	19.3 ± 6.2	31.6 ± 2.4	0.2 ± 0	6206 ± 1183

Insights Figure 3 shows a visual comparison of AutoWP (WP curve modeling) and TFT (autoregressive DL) for WP turbine no. 1. Additionally, the wind speed measurement at hub height and the height-corrected wind speed forecast from ECMWF are shown. This exemplary comparison shows that the static WP curve modeling methods (no temporal context) cannot forecast intervention-related WP turbine shutdowns, while the TFT captures them with a delay. Specifically, on the day-ahead forecast no. 4, the TFT captures the shutdown in the past

⁸ For training, we use an Intel Xeon Platinum 8368 CPU with 76 cores and an NVIDIA A100-40 GPU with 40GB memory (the GPU is only used for the DL-based TFT), and 256GB RAM provided by the HAICORE HPC.

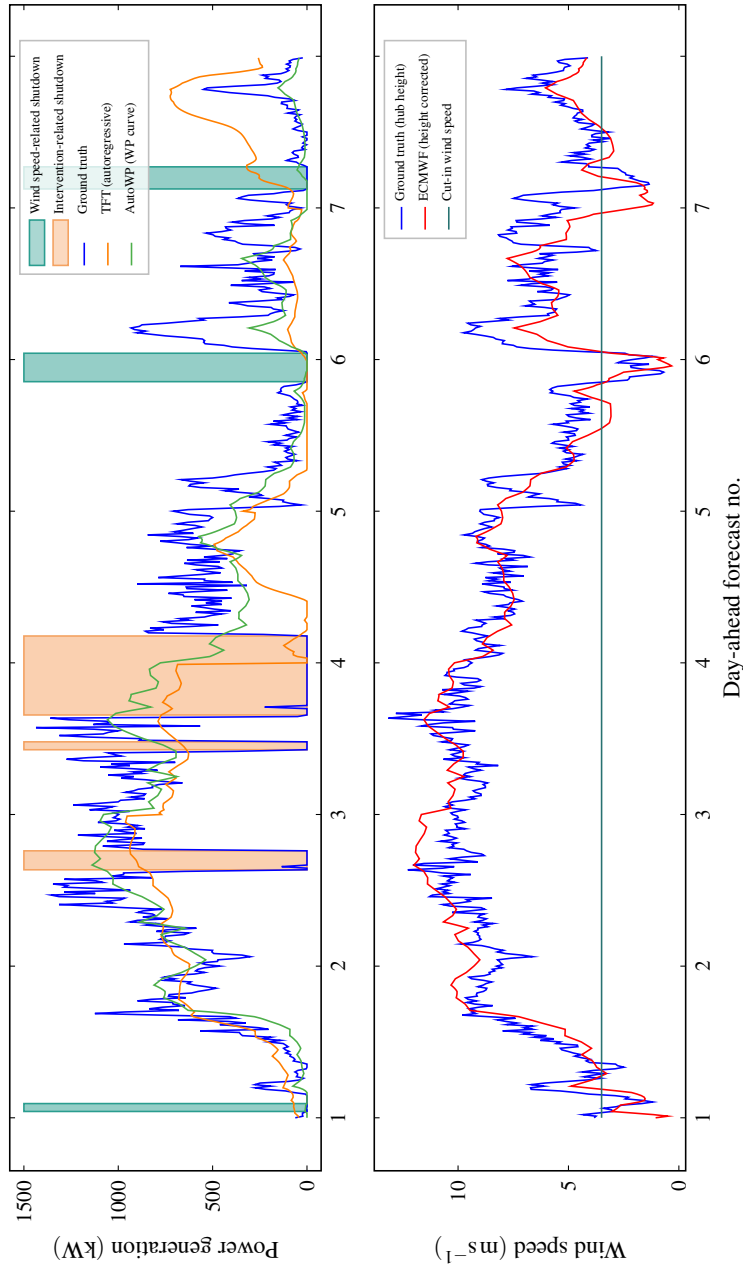


Figure 3: Exemplary comparison of seven day-ahead WP forecasts with origin at 00:00 for WP turbine no. 1 using AutoWP (WP curve modeling) TFT (autoregressive). While AutoWP is only based on wind speed forecasts (ECMWF), the TFT also considers past WP generation measurements.

horizon of the previous day. However, the shutdown in the past horizon also leads to a delayed capture of the WP turbine activation.

4 Discussion

This section discusses the benchmarking results, the limitations, and benefits of AutoWP.

Benchmarking We discuss three aspects: First, the results show that all considered methods outperform the OEM WP curve. This can be explained by the fact that the OEM WP curve does not consider site-specific conditions characterized by terrain-related air turbulence. As these conditions may be very different from those under which the OEM WP curve was determined, it is worth training the forecasting model to these conditions using measured data from the WP turbine. Second, the computational effort required to train the model varies greatly. Although the OEM curve requires the least amount of computational effort (0.001 s), the forecasting error can be reduced considerably with a small amount of effort using AutoWP (0.6 s). This is because the computational effort for solving the least squares optimization problem to determine AutoWP's optimal ensemble weights is similar to linear regression, but AutoWP can represent the non-linear progression of the curve through the ensemble of WP curves. Third, it is apparent that the methods based on WP curve modeling cannot forecast interventions into the WP generation capability. Although in the autoregressive DL method TFT, past shutdowns contribute to a certain extent to forecast future shutdowns, shutdowns are only captured with a time delay. On the one hand, this means that additional information is required if future shutdowns *must* be reliably forecasted and, on the other hand, that shutdowns in the past horizon require data imputation if future shutdowns *must not* be forecasted.

Limitations We discuss three aspects: First, although we consider two years of measurements from two WP turbines, the data set is rather small to draw

generalizable conclusions. However, the results show that AutoWP can significantly reduce the forecasting error while requiring only a small amount of computational effort. While the performance of the benchmarks could potentially be improved by hyperparameter optimization, it would further increase their computational effort. Second, recurring shutdown patterns are visible in the data (see Figure 4), which can potentially be forecasted using additional information. Although the results show that shutdowns in the past horizon of the autoregressive DL method TFT contributes to the forecast, further evaluation is necessary, e. g., shutdown imputation in the past horizon, additional features for encoding the seasonality, and considering the theoretical WP generation during shutdowns. Third, the measured wind speed at hub height does not reach the cut-out speed $v_{\text{cut-out}} = 25 \text{ ms}^{-1}$, at which the WP turbine would be shut down for safety reasons. However, as AutoWP also accounts for these shutdowns in the ensemble WP curve (see Figure 1), application to WP turbines in regions with higher wind speeds is possible.

Benefits AutoWP is beneficial since it can represent various site conditions using the ensemble of diverse WP curves. Specifically, the site conditions are

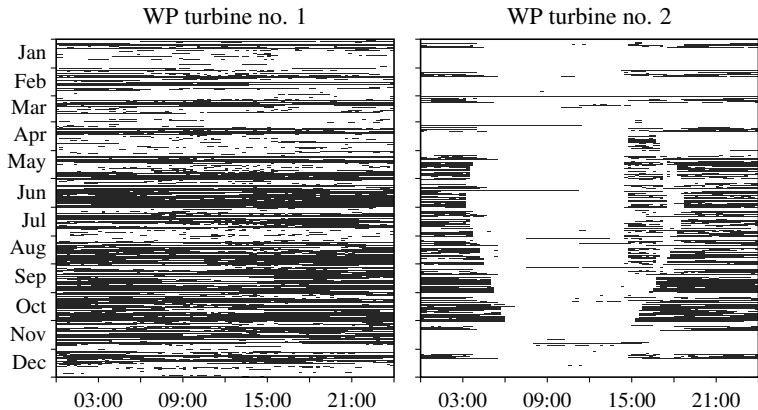


Figure 4: Identified WP turbine shutdowns with rule-based filtering. Black fields represent data points at a measured wind speed greater than the cut-in speed with a power generation lower than the cut-in power.

characterized by terrain-related air turbulence, which affects the wind speed where the WP reaches its peak power output. Although ML methods can also be used to train a site-specific WP curve, AutoWP implicitly considers physical limitations in WP generation and requires only a few data to train the model. This scalability is an essential prerequisite for deploying the model to hundreds of decentral located onshore WP turbines to serve smart grid applications like redispatch planning.

5 Conclusion and outlook

Many smart grid applications require forecasts of locally distributed WP generation. In order to handle the increasing demand for WP forecasting models, scalable methods are required that can be trained efficiently to represent the site-specific conditions characterized by the terrain-dependent air turbulence, while being robust in terms of inconsistent training data. AutoWP addresses these challenges, providing a scalable day-ahead forecasting method that includes the identification and filtering of inconsistent training data and implicitly considers physical limitations in WP generation. While achieving competitive performance against other WP curve modeling methods, the evaluation shows that such methods require knowledge about scheduled interventions into the WP generation capabilities to consider future shutdowns in the forecast. This limitation is overcome to some extent by the benchmarking method TFT, which is a state-of-the-art autoregressive DL method that considers past WP generation values, also including shutdowns. The TFT, however, captures shutdowns with a time delay and requires an enormous training effort.

Therefore, future work could evaluate WP turbines from different regions, and different shutdown handling methods for autoregressive WP forecasting methods. Furthermore, since only the TFT was evaluated, it is an open question how other DL-based autoregressive methods perform in WP forecasting in the presence of regular and irregular shutdowns.

Acknowledgements

This project is funded by the Helmholtz Association under the Program “Energy System Design”, the German Research Foundation (DFG) as part of the Research Training Group 2153 “Energy Status Data: Informatics Methods for its Collection, Analysis and Exploitation” and the Helmholtz Association’s Initiative and Networking Fund through Helmholtz AI, and is supported by the HAICORE@KIT partition. Furthermore, the authors thank Stadtwerke Karlsruhe Netzservice GmbH (Karlsruhe, Germany) for the data required for this work.

References

- [1] H. Xu, Y. Chang, Y. Zhao, and F. Wang. A new multi-timescale optimal scheduling model considering wind power uncertainty and demand response. *International Journal of Electrical Power & Energy Systems*, vol. 147, p. 108832, 2023. doi:10.1016/j.ijepes.2022.108832.
- [2] Z. Cai, C. Bussar, P. Stöcker, L. Moraes, D. Magnor, and D. U. Sauer. Optimal Dispatch Scheduling of a Wind-battery-System in German Power Market. *Energy Procedia*, vol. 99, pp. 137–146, 2016. doi:10.1016/j.egypro.2016.10.105.
- [3] F. Salm, M. Oettmeier, and P. Rönsch. The new redispatch and the impact on energy management in districts. In *Proceedings of the 2022 18th International Conference on the European Energy Market (EEM)*, pp. 1–4, 2022. doi:10.1109/EEM54602.2022.9921014.
- [4] R. Meka, A. Alaeddini, and K. Bhaganagar. A robust deep learning framework for short-term wind power forecast of a full-scale wind farm using atmospheric variables. *Energy*, vol. 221, p. 119759, 2021. doi:10.1016/j.energy.2021.119759.
- [5] M. Lydia, S. S. Kumar, A. I. Selvakumar, and G. E. Prem Kumar. A comprehensive review on wind turbine power curve modeling techniques.

Renewable and Sustainable Energy Reviews, vol. 30, pp. 452–460, 2014. doi:10.1016/j.rser.2013.10.030.

- [6] J. Cidrás, E. Díaz-Dorado C. Carrillo, A. F. Obando Montaña, J. Cidrás, J. and E. Díaz-Dorado. Review of power curve modelling for wind turbines. *Renewable and Sustainable Energy Reviews*, vol. 21, pp. 572–581, 2013. doi:10.1016/j.rser.2013.01.012.
- [7] A. Arnold, S. König, R. Mikut, and P. Bretschneider. Application of data mining methods for power forecast of wind power plants. In *Proceedings of the 9th International Workshop on Large-Scale Integration of Wind Power into Power Systems as well as on Transmission Networks for Offshore Wind Power Plants*, pp. 655–60, Quebec City, Canad, 2010. Energynautics GmbH, Langen, Germany. ACM, New York, USA.
- [8] R. K. Pandit, D. Infield, and A. Kolios. Comparison of advanced non-parametric models for wind turbine power curves. *IET Renewable Power Generation*, vol. 13, no. 9, pp. 1503–1510, 2019. doi:10.1049/iet-rpg.2018.5728.
- [9] Y. Wang, Q. Hu, D. Srinivasan, and Z. Wang. Wind power curve modeling and wind power forecasting with inconsistent data. *IEEE Transactions on Sustainable Energy*, vol. 10, no. 1, pp. 16–25, 2019. doi:10.1109/TSTE.2018.2820198.
- [10] S. R. Moreno, L. S. Coelho, H. V. H. Ayala, and V. C. Mariani. Wind turbines anomaly detection based on power curves and ensemble learning. *IET Renewable Power Generation*, vol. 14, no. 19, pp. 4086–4093, 2020. doi:10.1049/iet-rpg.2020.0224.
- [11] S. Li, D. C. Wunsch, E. A. O’Hair, and M. G. Giesselmann. Using neural networks to estimate wind turbine power generation. *IEEE Transactions on Energy Conversion*, vol. 16, no. 3, pp. 276–282, 2001. doi:10.1109/60.937208.
- [12] M. Schlechtingen, I. F. Santos, and S. Achiche. Using data-mining approaches for wind turbine power curve monitoring: A comparative

- study. *IEEE Transactions on Sustainable Energy*, vol. 4, no. 3, pp. 671–679, 2013. doi:10.1109/TSTE.2013.2241797.
- [13] U. Singh, M. Rizwan, M. Alaraj, and I. Alsaidan. A machine learning-based gradient boosting regression approach for wind power production forecasting: A step towards smart grid environments. *Energies*, vol. 14, no. 16, article 5196, 2021. doi:10.3390/en14165196.
 - [14] M. Á. Rodríguez-López, E. Cerdá, and P. del Río. Modeling wind-turbine power curves: Effects of environmental temperature on wind energy generation. *Energies*, vol. 13, no. 18, article 4941, 2020. doi:10.3390/en13184941.
 - [15] R. Morrison, X. Liu, and Z. Lin. Anomaly detection in wind turbine SCADA data for power curve cleaning. *Renewable Energy*, vol. 184, pp. 473–486, 2022. doi:10.1016/j.renene.2021.11.118.
 - [16] S. Sun, Y. Liu, Q. Li, T. Wang, and F. Chu. Short-term multi-step wind power forecasting based on spatio-temporal correlations and transformer neural networks. *Energy Conversion and Management*, vol. 283, p. 116916, 2023. doi:10.1016/j.enconman.2023.116916.
 - [17] L. van Heerden, C. van Staden, and H. J. Vermeulen. Temporal Fusion Transformer for day-ahead wind power forecasting in the South African context. In *Proceedings of the 2023 IEEE International Conference on Environment and Electrical Engineering and 2023 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*, pp. 1–5, 2023. doi:10.1109/EEEIC/ICPSEurope57605.2023.10194737.
 - [18] Y. Hu, H. Liu, S. Wu, Y. Zhao, Z. Wang, and X. Liu. Temporal collaborative attention for wind power forecasting. *Applied Energy*, vol. 357, p. 122502, 2024. doi:10.1016/j.apenergy.2023.122502.
 - [19] S. Xu, Y. Wang, X. Xu, G. Shi, Y. Zheng, H. Huang, and C. Hong. A multi-step wind power group forecasting seq2seq architecture with spatial-temporal feature fusion and numerical weather prediction correction. *Energy*, vol. 291, p. 130352, 2024. doi:10.1016/j.energy.2024.130352.

- [20] S. Mo, H. Wang, B. Li, Z. Xue, S. Fan, and X. Liu. Powerformer: A temporal-based transformer model for wind power forecasting. *Energy Reports*, vol. 11, pp. 736–744, 2024. doi:10.1016/j.egy.2023.12.030.
- [21] M. Petersen, J. Huber, and C. Hofmann. Wind turbine library. Further contributors: Ludee, jh-RLI, 2019. https://openenergyplatform.org/dataedit/view/supply/wind_turbine_library,
- [22] S. Meisenbacher, B. Heidrich, T. Martin, R. Mikut, and V. Hagenmeyer. AutoPV: Automated photovoltaic forecasts with limited information using an ensemble of pre-trained models. In *Proceedings of the 14th ACM International Conference on Future Energy Systems (e-Energy '23)*, pp. 386–414, Orlando, USA, 2023. ACM, New York, USA. doi:10.1145/3575813.3597348.
- [23] S. Meisenbacher, T. Martin, B. Heidrich, R. Mikut, and V. Hagenmeyer. Automating day-ahead forecasting of photovoltaic power generation: Model design, monitoring, and adaption. In *ETG Congress 2023*, pp. 1–8, 2023.
- [24] D. Etling. Theoretische Meteorologie: Eine Einführung. Chapter 21: Die atmosphärische Grenzschicht, pp. 297–340. Springer-Verlag, Berlin, Heidelberg, Germany, 2004. doi:10.1007/978-3-540-75979-9_21.
- [25] G. M. Masters. Wind power systems. Chapter 6 in *Renewable and efficient electric power systems*, pp. 307–383. John Wiley & Sons, Ltd, Chichester, UK, 2004. doi:10.1002/0471668826.ch6.
- [26] S. A. Hsu, E. A. Meindl, and D. B. Gilhousen. Determining the power-law wind-profile exponent under near-neutral stability conditions at sea. *Journal of Applied Meteorology and Climatology*, 33(6):757–765, American Meteorological Society, Boston, USA, 1994. doi:10.1175/1520-0450(1994)033<0757:DTPLWP>2.0.CO;2.
- [27] P. Virtanen et al.. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17:261–272, 2020. doi:10.1038/s41592-019-0686-2.

- [28] European Centre for Medium-Range Weather Forecasts (ECMWF). Meteorological Archival and Retrieval System (MARS). <https://confluence.ecmwf.int/display/CEMS/MARS>. Accessed: 2024-06-03, 2023.
- [29] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister. Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021. doi:10.1016/j.ijforecast.2021.03.012.
- [30] J. Beitner. PyTorch Forecasting. <https://towardsdatascience.com/introducing-pytorch-forecasting-64de99b9ef46>. Accessed: 2024-06-03, 2020.
- [31] F. Pedregosa et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(85):2825—2830, 2011.
- [32] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '23)*, pp. 785–794, San Francisco, USA, 2016. ACM, New York, USA. doi:10.1145/2939672.2939785.