Wind Power Prediction with Machine Learning

Nils André Treiber, Justin Heinermann and Oliver Kramer

Abstract Better prediction models for the upcoming supply of renewable energy are important to decrease the need of controlling energy provided by conventional power plants. Especially for successful power grid integration of the highly volatile wind power production, a reliable forecast is crucial. In this chapter, we focus on short-term wind power prediction and employ data from the National Renewable Energy Laboratory (NREL), which are designed for a wind integration study in the western part of the United States. In contrast to physical approaches based on very complex differential equations, our model derives functional dependencies directly from the observations. Hereby, we formulate the prediction task as regression problem and test different regression techniques such as linear regression, k-nearest neighbors and support vector regression. In our experiments, we analyze predictions for individual turbines as well as entire wind parks and show that a machine learning approach yields feasible results for short-term wind power prediction.

1 Introduction

The strong increase in renewable energy causes some problems induced by high fluctuations in production. A precise prediction is the key technology for successful integration of the wind power into the grid because it allows planning reserve plants, battery loading strategies and scheduling of consumers.

Generally, there are two model classes for prediction tasks. Most prediction approaches are based on physical models employing numerical weather simulations, see e.g., [3]. These models are used for short- and long-term forecasts in the

N.A. Treiber (✉) · J. Heinermann · O. Kramer
University of Oldenburg, 26111 Oldenburg, Germany
e-mail: nils.andre.treiber@uni-oldenburg.de

J. Heinermann
e-mail: justin.philipp.heinermann@uni-oldenburg.de

O. Kramer
e-mail: oliver.kramer@uni-oldenburg.de

J. Lässig et al. (eds.), *Computational Sustainability*,
Studies in Computational Intelligence 645,
DOI 10.1007/978-3-319-31858-5_2

range of hours. The other class of prediction methods is formed by machine learning algorithms that are implemented more frequently in recent years. Since these statistical models derive functional dependencies directly from the observations, they are also known as *data-driven* models. They can be used for predictions that target a time horizon from seconds to hours and therefore are important for balancing the electrical grid with its different authorities.

In this chapter, we build models that are exclusively based on wind power time series measurements. We formulate the prediction task as a regression problem and compare the accuracy of different regression techniques by employing the two simple regression methods *linear regression* and *k-nearest neighbors* (*k*NN), and the state-of-the-art technique *support vector regression* (SVR). In our studies we make predictions for individual turbines and then for entire wind parks. The latter predictions can be made with various feature aggregation combinations of the corresponding time series that we also compare. This chapter extends our preliminary workshop paper [21] on the workshop *Data Analytics for Renewable Energy Integration* on the *European Conference on Machine Learning (ECML 2013).*

2 Related Work

In their review, Costa et al. [2] present a broad overview of various methods and mathematical, statistical and physical models employed in the last 30 years for short-term prediction. Soman et al. [19] give an extensive survey of the possible techniques for different forecast horizons. Past results have shown that methods from statistical learning are powerful approaches for short-term energy prediction. For example, Juban et al. [8] presented a kernel density estimation approach for a probabilistic forecasting for different wind parks. Foresti et al. [5] employed multiple kernel learning regression as an extended support vector model that autonomously detects the relevant features for wind speed predictions. Also neural networks have been applied to wind power prediction in the past, e.g., by Mohandes et al. [13], who compared an autoregressive model with a classical backpropagation network. In this line of research, Catalao et al. [1] trained a three-layered feedforward network with the Levenberg-Marquardt algorithm for short-term wind power forecasting, which outperformed the persistence model and ARIMA approaches. Further, Han et al. [7] focused on an ensemble method of neural networks for wind power prediction. Regarding the aggregation of wind turbines, Focken et al. [4] studied the decrease of the prediction error of an aggregated power output caused by spatial smoothing effects. From the perspective of electrical engineers, Pöller and Achilles [16] explored, how different wind turbines can be aggregated to a single generator.

The spatio-temporal wind power prediction approach that is basis of our line of research has been introduced in [10] with a more extensive depiction in [11]. In [20], we presented an approach for preselection of turbines for *k*NN-based prediction. As the optimization problem is difficult to solve, we proposed an evolutionary blackbox method for an efficient feature selection, which corresponds to a selection of appropriate turbines. In [6], we proposed an ensemble approach for SVR, where small subsets of training data are randomly sampled and the predictions of multiple SVRs are combined to a strong classifier. As wind power ramps are difficult events for the integration into the grid, we considered this problem in a separate work [12]. We treat ramp prediction as classification problem, which we solve with SVMs. Recursive feature selection illustrates how the number of neighbored turbines affects this approach. The problem of imbalanced training and test sets is analyzed with regard to the number of no-ramp events. In practice, sensors might fail for various reasons and the prediction models cannot be applied. In [17], we compared various missing data methods for the imputation problem. A new contribution of this work is a *k*NN-based regression method, which is used as geo-imputation preprocessing step by taking into account the time series of the neighbored turbines. Last, in [22] we extended the repertoire of prediction methods with a cross-correlation weighted k-nearest neighbor regression (x-*k*NN) variant. The *k*NN-based similarity measure employs weights that are based on the cross-correlation of the time series of the neighboring turbines and the target. If the cross-correlation coefficient is high, the turbine gets a major influence for the prediction by expanding the corresponding dimension in the regression model.

3 Wind Data Set

The models are evaluated based on the *National Renewable Energy Laboratory* (NREL) western wind resources data set [14], which is part of the *Western Wind and Solar Integration Study*, a large regional wind and solar integration study initiated by the United States. The data set has been designed to perform temporal and spatial comparisons like load correlation or estimation of production from hypothetical (i.e., simulated) wind turbines for demand analysis and planning of storage based on wind variability. The data set consists of three years of wind energy data from numerical simulations that are mainly based on real-world wind measurements. It consists of 32,043 turbines in the western area of the US, and can be downloaded from the NREL website.[1] The whole model employs a total capacity of 960 GW of wind energy. A GUI allows to select turbines, and to download their corresponding time series data. Based on a time-resolution of ten minutes, 52,560 entries per year and per turbine are available for 2004, 2005, and 2006, respectively.

In Fig. 1, four different wind situations in a park near Tehachapi are illustrated. One can observe spatio-temporal correlations between the wind speed of the turbines. But it can also be noticed, that occasionally high wind speeds can occur locally.

[1] http://www.nrel.gov/.

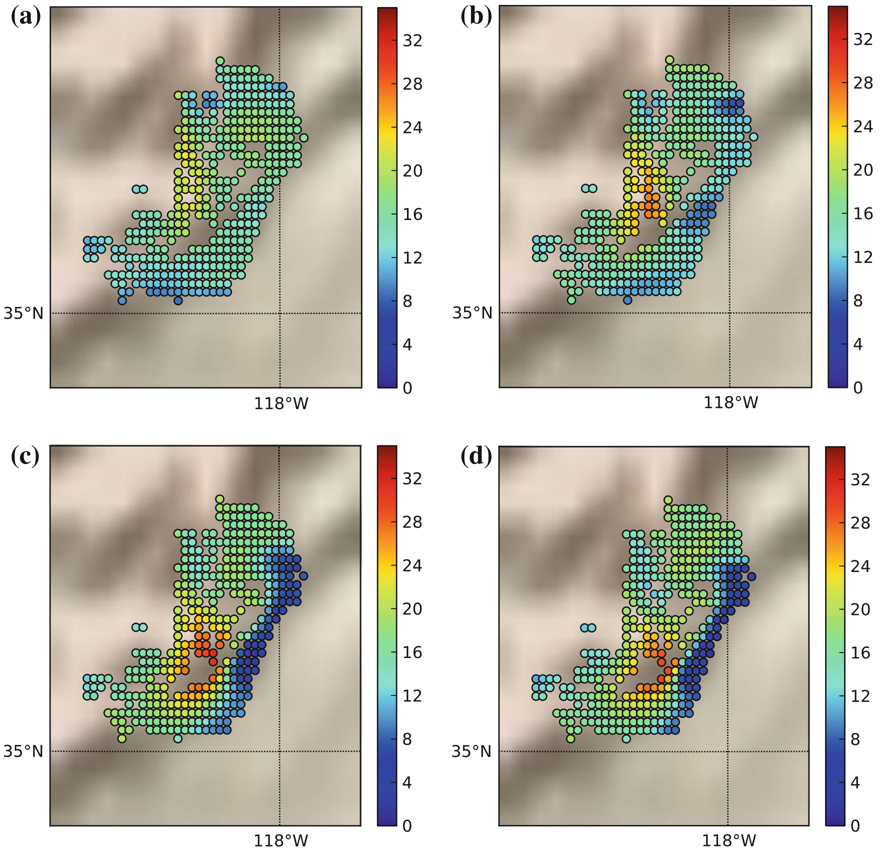

Fig. 1 Visualization of the wind speeds (m/s) in a park near Tehachapi (California) at four time steps with a temporal difference of 20 min. The turbines are colorized with regard to the wind strength, from low in *blue* to strong in *red*. **a** 2004-01-25 00:40, **b** 2004-01-25 01:00, **c** 2004-01-25 01:20, **d** 2004-01-25 01:40

4 WindML

For addressing the challenge to couple machine learning and data mining methods to wind power time series data, we developed the Python-based framework WindML [9], which aims at minimizing the obstacles for data-driven research in the wind power domain. It allows the simplification of numerous steps like loading and preprocessing large scale wind data sets or the effective parameterization of machine learning and data mining approaches. With a framework that bounds specialized mining algorithms to data sets of a particular domain, frequent steps of the data mining process chain can be re-used and simplified. The WindML framework is released under the open source *BSD 3-Clause License*.

The WINDML framework provides a data server, which automatically downloads requested data sets to a local cache when used for the first time. The system only downloads the data for the requested wind turbines and associated time range. The local copies on the user's hard disk are stored in the NUMPY [24] binary file format, allowing an efficient storage and a fast (recurrent) loading of the data. The data set interface allows the encapsulation of different data sources, resulting in a flexible and extendible framework. A complete documentation describing the modules and the overall framework in more detail is openly available. Further, documentations and examples are available, e.g., for the tasks of wind power prediction or visualization, whose outputs are rendered with text and graphics on the WINDML project website.

The application of machine learning and data mining tools to raw time series data often requires various preprocessing steps. For example, wind power prediction for a target wind turbine using turbines in the neighborhood affords the composition of the given power values as feature vector matrix. The WINDML framework simplifies such tasks and offers various methods for assembling corresponding patterns. It also effectively supports the imputation of missing data as well as other sophisticated processes including the computation of high-level features. In addition, the framework also provides various small helper functions, which, for instance, address the tasks of calculating the Haversine distance between given coordinates or selecting the wind turbines in a specified radius. The WINDML framework contains various supervised and unsupervised learning models. Most of the employed machine learning and data mining implementations are based on SCIKIT-LEARN [15], which offers a wide range of algorithms. The methods are continuously extended with own developments, see Sect. 2.

5 General Times Series Model

Our model makes predictions with past wind power measurements. For this task, we formulate the prediction as regression problem. Let us first assume, we intend to predict the power production of a single turbine with its time series. The wind power measurement $\mathbf{x} = p(t)$ (pattern) is mapped to the power production at a target time $y = p_T(t + \lambda)$ (label) with $\lambda \in \mathbb{N}^+$ being the forecast horizon. For our regression model we assume to have N of such pattern-label pairs that are the basis of the training set $T = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ and allow, via a regression, to predict the label for a unknown pattern $\mathbf{x}'$.

One can assume that this model yields better predictions if more information of the time series will be used. For this reason, we extend the pattern $\mathbf{x}$ by considering past measurements $p_T(t-1), \ldots, p_T(t-\mu)$ with $\mu \in \mathbb{N}^+$, see Fig. 2. In addition, it might be helpful to take differences of measurements $p_T(t) - p_T(t-1), \ldots p_T(t - (\mu - 1)) - p_T(t-\mu)$ into account. Since we aim to catch spatio-temporal correlations, we add further information to our patterns from m neighboring turbines, see Fig. 3. Their attributes can be composed in the same way like for the target turbine

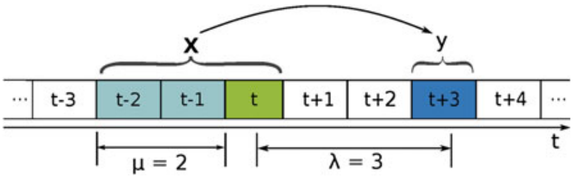

Fig. 2 Section of a time series. The pattern **x** is mapped to label y. The time horizon of the prediction is λ, and the number of additional past measurements is μ

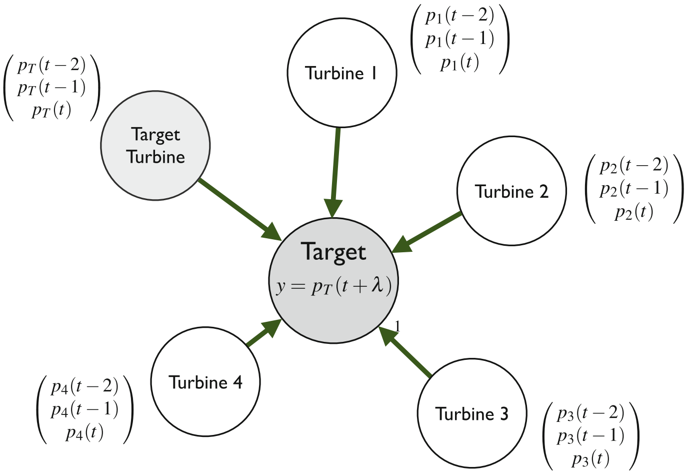
$\begin{pmatrix} p_T(t-2) \\ p_T(t-1) \\ p_T(t) \end{pmatrix}$
Target Turbine
Turbine 1
$\begin{pmatrix} p_1(t-2) \\ p_1(t-1) \\ p_1(t) \end{pmatrix}$
Turbine 2
$\begin{pmatrix} p_2(t-2) \\ p_2(t-1) \\ p_2(t) \end{pmatrix}$
Target
$y = p_T(t+\lambda)$
1
Turbine 4
$\begin{pmatrix} p_4(t-2) \\ p_4(t-1) \\ p_4(t) \end{pmatrix}$
Turbine 3
$\begin{pmatrix} p_3(t-2) \\ p_3(t-1) \\ p_3(t) \end{pmatrix}$

Fig. 3 Setup with four neighboring turbines and the target turbine itself. For each turbine, the current and two past measurements are considered, resulting in a $(4+1) \cdot 3 = 15$-dimensional input vector **x**

(e.g., power values and differences). Finally, we determine the accuracy of the prediction model by computing the error E of the prediction, which is measured as mean squared error (MSE):

$$E = \frac{1}{N}\sum_{i=1}^{N}(f(\mathbf{x}_i) - y_i)^2 \tag{1}$$

6 Regression Techniques

In this section, we introduce the implemented regression techniques. Generally, the goal is to find a function f that provides appropriate predictions to unseen patterns $\mathbf{x}'$. In the following, we explain linear regression, the basic idea of SVR and k-nearest neighbor regression.

6.1 Linear Regression

First, we focus on linear regression. In this model, the prediction value $f(\mathbf{x})$ is expected to be a linear combination of the input variables:

$$f(\mathbf{w}, \mathbf{x}) = w_0 + w_1 x_1 + \cdots + w_N x_N \tag{2}$$

The goal is to find coefficients $\mathbf{w} = (w_1, \ldots, w_N)^T$ that minimize the residual sum of the squares between the observed labels y and the responses predicted by the linear approximation. The problem has the form:

$$\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - y\|^2 \quad \text{with} \quad \mathbf{X} = \begin{pmatrix} x_{1,1} & \ldots\ldots & x_{1,N} \\ \ldots & \ldots\ldots & \ldots \\ x_{d,1} & \ldots\ldots & x_{d,N} \end{pmatrix} \tag{3}$$

6.2 Support Vector Regression

Support vector regression is one of the state-of-the-art techniques for prediction tasks. It is based on support vector machines (SVMs) that were proposed by Vapnik [23] in 1995. For the training of the regressor, we aim at finding weights $\mathbf{w}$ by minimizing the following problem, formulated by Vapnik with an ε-sensitive loss function:

$$\text{minimize } \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{N} (\xi_i + \xi_i^*) \tag{4}$$

$$\text{subject to} \begin{cases} y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b & \leq \varepsilon + \xi_i \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b - y_i & \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* & \geq 0 \end{cases}$$

In this formulation, $C > 0$ is a constant chosen by the user that is used as a parameter that penalizes only those errors which are greater than ε. The so-called slack variables ξ_i^* are introduced to provide a soft margin instead of a hard decision border.

To give good results on non-linear separable data as well, kernel functions are used. A kernel function can be seen as a similarity measure between patterns and is especially useful for non-linear regression tasks. In our experiments, we employ an RBF-kernel:

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\sigma^2}\right) \tag{5}$$

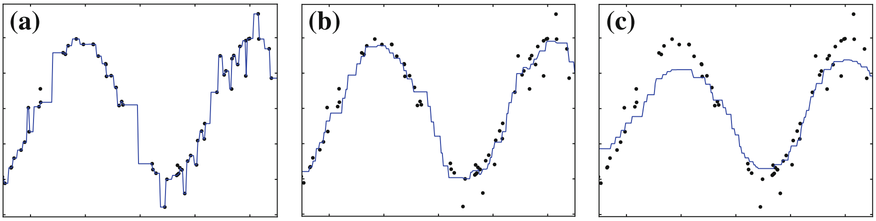

Fig. 4 Visualization of kNN regression. The figures show how the choice of the neighborhood size k determines the character of the regression. **a** $k = 1$, **b** $k = 6$, **c** $k = 18$

6.3 k-Nearest Neighbor Regression

In opposition to the SVR, the kNN model is one of the simplest of all machine learning algorithms. Nevertheless, this technique is very effective for many applications and often offers a better performance than using SVR. The outcome of kNN given a new pattern $\mathbf{x}'$ depends on the k-nearest neighbored patterns in a training set. These patterns are found by calculating the distance between the pattern and all existing patterns $dist(\mathbf{x}, \mathbf{x}')$ using the Euclidean metric:

$$dist(\mathbf{x}, \mathbf{x}') = \left(\sum_{i=1}^{d} (x_i - x'_i)^2 \right)^{1/2} \tag{6}$$

With set $\mathcal{N}_k(\mathbf{x}')$ that contains the indices of the k-nearest neighbors of $\mathbf{x}'$, the target value is given by:

$$f_{k\mathrm{NN}}(\mathbf{x}') = \frac{1}{k} \sum_{i \in \mathcal{N}_k(\mathbf{x}')} y_i, \tag{7}$$

if we intend to calculate the arithmetic average of the other k target values. Here, the number of considered neighbors determines the form of the resulting regression function. Given a small value, the model fits the data but is also strongly influenced by outliers, whereas large k might build up models that are too simple, see Fig. 4.

7 Wind Power Prediction for a Single Turbine

In this first experimental section, we want to predict the power output of a single wind turbine for a time horizon of 30 min.

7.1 Setup and Evaluation

For this prediction, neighboring turbines must be selected whose time series features are used in our multivariate time series model. For this sake, we arbitrarily pick 15 turbines that surround the target turbine within a radius of 10 km. For the determination of the distance between turbines, we use the Haversine formula [18]. Finally, every model is trained by using the data from the year of 2005. To accelerate the training process, only every fourth time-step is taken into account. Despite a smaller training set, it is guaranteed that wind conditions at different seasons are included. For the evaluation we test our models on the year 2006 by determining the MSE of the forecasts $f(\mathbf{x})$ with the measured power outputs y_i for N forecasts, see Eq. 1. In our experiments we use five turbines at different locations that are distributed over the western part of the United States.

7.2 Persistence Model

In our studies, forecasts are compared to the persistence model (PST). This model is based on the assumption that the wind power in the time horizon λ does not change and is as strong as at the present time, i.e. $f(p_\alpha(t)) = p_\alpha(t + \lambda)$. Although this naive approach seems to be trivial, its predictions are quite successful for short time horizons and temporally relative constant wind situations, shown for example by Wegley et al. [25]. In our studies, this model achieves mean squared errors, shown in Table 1.

Since it is an important challenge for balancing the grid to predict ramps of changing wind, our models should outperform the persistence model, which understandably fails in such situations.

7.3 Accuracy of the Linear Regression Prediction Model

The experimental results of the linear regression are shown in Table 2. Here the mean squared errors are given for predictions that, on the one hand, are achieved

Table 1 Mean squared error E of the PST model in $[MW^2]$ for the five target turbines

Turbine	$E\ [MW^2]$
Tehachapi	9.499
Lancaster	11.783
Palmsprings	7.085
Cheyenne	9.628
Casper	12.555

It is noticeable that the errors of the individual turbines are quite different

Table 2 Mean squared errors E of the linear regression model in $[MW]^2$

Turbine	$E\,[MW^2]$					
	Absolute			Absolute + changes		
	$\mu = 2$	$\mu = 3$	$\mu = 4$	$\mu = 2$	$\mu = 3$	$\mu = 4$
Tehachapi	7.441	7.334	**7.321**	7.470	7.348	7.365
Lancaster	8.610	**8.479**	8.488	8.631	8.536	8.536
Palmsprings	5.533	5.372	5.347	5.533	5.374	**5.346**
Cheyenne	**7.666**	7.691	7.701	7.702	7.704	7.716
Casper	9.984	10.026	10.067	**9.982**	10.039	10.108

The results shown on the left side are achieved with features only presenting past absolute measurements of the wind power. On the right side, the corresponding differences of the measurements are also taken into account

only with the absolute measurements of the generated power and, on the other hand, are produced with the absolute values and their corresponding changes. From the table one cannot decide which of the two feature vectors is more suitable for the prediction. It is also not clear how many past steps should be considered. The main result is that our linear regression predictions can clearly outperform the persistence model.

7.4 Results of the SVR Prediction

The SVR model requires an appropriate choice of its parameters. We implement a grid search with a three-fold cross validation and test values of the trade-off parameter $C = 0.1,\ 10,\ 100,\ 1000,\ 2000$ and the kernel bandwidth $\sigma = 10^{-i}$ with $i = 2, 3, \ldots, 7$. The Fig. 5 shows an arbitrarily chosen section of the time series of a turbine near Tehachapi and allows to compare the persistence and the SVR prediction. The quantitative results are concluded in Table 3.

The results show that the SVR technique even achieves better results than the linear regression if one takes only the absolute measurements as features and selects the number of past steps carefully.

7.5 Results of the kNN Model

The kNN model is tested with three different values for k. The mean squared errors of the predictions are shown in Table 4. The accuracy of kNN model is not as high as in the other two methods. In contrast to the SVR technique, this model requires inputs for its best predictions that also contain the corresponding differences of the measurements.

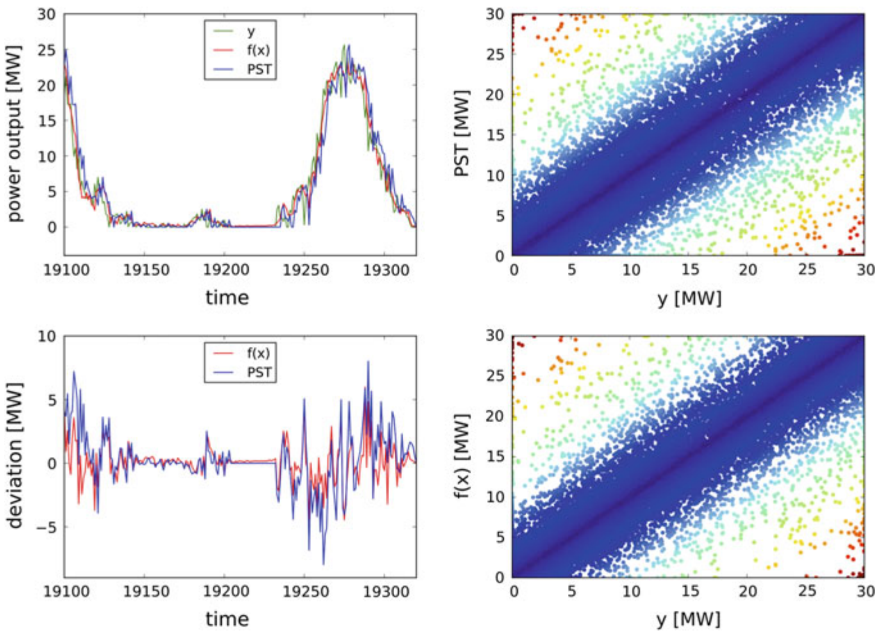

Fig. 5 Visualization of the SVR prediction for a turbines in Tehachapi. The *top left plot* compares the SVR and the persistence predictions to the actual wind power measurements. The *bottom left plot* shows the differences of the corresponding models. The plots on the *right hand side* show the actual and predicted measurement pairs. The absolute prediction errors is the deviation to the main diagonal

Table 3 Results of the prediction using the SVR model

Turbine	$E\ [MW^2]$					
	Absolute			Absolute+changes		
	$\mu = 2$	$\mu = 3$	$\mu = 4$	$\mu = 2$	$\mu = 3$	$\mu = 4$
Tehachapi	7.093	7.074	**6.984**	7.131	7.096	7.012
Lancaster	8.364	**8.108**	8.157	8.397	8.339	8.342
Palmsprings	5.303	5.231	**5.209**	5.387	5.251	5.256
Cheyenne	7.670	7.713	**7.646**	7.679	7.702	7.694
Casper	9.644	**9.493**	9.507	9.798	9.628	9.539

In comparison to the previous results, a higher accuracy is achieved

8 Prediction for a Wind Park

Since the power output of a single turbine is generally not high enough to cause relevant effects on the electric grid, we want to analyze the power output of an entire wind park. The objective of this section is to compare different features and various feature aggregation combinations of the turbines belonging to the park.

Table 4 Prediction errors of the kNN model

Turbine	k	$E\ [MW^2]$					
		Absolute			Absolute + changes		
		$\mu = 2$	$\mu = 3$	$\mu = 4$	$\mu = 2$	$\mu = 3$	$\mu = 4$
Tehachapi	5	9.957	10.101	10.115	9.380	9.785	9.932
	25	9.296	9.720	9.906	**8.887**	9.399	9.725
	50	9.643	10.117	10.339	9.241	9.799	10.133
Lancaster	5	11.340	11.266	11.541	10.569	10.665	11.099
	25	10.837	10.848	11.016	**10.058**	10.296	10.698
	50	10.929	11.216	11.488	10.229	10.628	11.115
Palmsprings	5	7.292	7.403	7.628	6.903	7.176	7.396
	25	6.812	7.136	7.404	**6.521**	6.864	7.250
	50	6.900	7.340	7.666	6.649	7.102	7.523
Cheyenne	5	9.749	10.014	10.171	9.526	9.897	10.181
	25	9.041	9.388	9.633	**8.864**	9.308	9.636
	50	9.135	9.539	9.860	9.010	9.480	9.858
Casper	5	13.033	13.371	13.698	12.409	12.951	13.440
	25	12.440	12.985	13.414	**11.943**	12.619	13.189
	50	12.835	13.440	13.925	12.348	13.127	13.723

It is expected that a tuning of parameter k should partially lead to slightly better predictions

8.1 Definition of a Wind Park

In our experiments, we intend to predict the power production of an entire wind park. Since parks are not explicitly given in the data set, we define a park by a central turbine, identified with a certain ID, and some neighboring turbines. To get these further turbines, we determine all turbines in a particular radius r_{in} around the central turbine. Since at least one of the considered aggregations should benefit from spatio-temporal relationships in a park, we choose a radius in an interval of $r_{in} = 7 - 12\,\text{km}$ to ensure that the park covers substantial region. Finally, we select $z = 25$ from all turbines within the selected radius.

8.2 Feature Aggregations

We test the three feature aggregation types.

8.2.1 Park Prediction as Sum of All Single Predictions

One possibility to make predictions for a park is to sum up the predictions of all z wind turbines. While here many different approaches for the individual predictions are possible, we only concentrate on two. First, we use the implementation for each turbine in the park by considering only some features of its own time series, resulting in a pattern with a dimension d_{st}. This setup is called AG-1.

Second, we also respect additional information of m arbitrarily selected neighboring turbines in the park for the individual predictions to get spatio-temporal information. Hereby, the extra features x_j are generated in the same way as in our studies of the individual prediction in the sections before. Eventually, one has a pattern with a dimension $d = (1 + m)\, d_{st}$, which is still mapped only to the power output $p_T(t + \lambda)$ of the target turbine T. This implementation is defined as setup AG-2.

8.2.2 Park Prediction with All Features of All Turbines

In this implementation, all time series features of all z turbines in the park are considered at once. The high-dimensional pattern includes the features of each turbine and thus has a dimension of $d = z \cdot d_{st}$. The label y is no longer the output power of an individual turbine, but rather the accumulated power of the park $y^{park} = \sum_{j=1}^{z} p_j(t + \lambda)$. This setup defines AG-3.

8.2.3 Park Prediction with One Aggregated Time Series

Finally, we use the aggregated power not only as label, but also for the patterns by computing one overall time series whose features are extracted as before. We label this model with AG-4:

$$\mathbf{x}^{park} = \begin{pmatrix} \sum_{j=1}^{z} p_j\big(t\big) \\ \cdots \\ \sum_{j=1}^{z} p_j\big(t - \mu\big) \end{pmatrix} \quad \longrightarrow \quad y^{park} = \sum_{j=1}^{z} p_j\big(t + \lambda\big)$$

A question is which feature aggregations yield the best predictions with regard to the MSE (see Eq. 1). We will experimentally compare the introduced aggregations after the explanation of the evaluation details.

8.3 Prediction with Support Vector Regression

For the SVR training process we employ a three-fold cross-validation and a grid search for the parameters in the interval $C = 10^i$ with $i = 2, 3, 4$ and $\sigma = 10^{-i}$ with $i = 4, 5$. These intervals have been manually restricted to accelerate the training process. In our first studies, we employ the SVR model with different aggregations. We compare the precision of the forecast with patterns that consist only of absolute values of wind measurements by using the actual measurement ($\mu = 0$), the actual and two past measurements ($\mu = 2$), and the actual and six past measurements ($\mu = 6$).

Table 5 Employing a 30 min forecast for a park with 25 turbines with the SVR model

	$E^* \, [MW^2]$					
	Absolute			Absolute + changes		
	$\mu = 0$	$\mu = 2$	$\mu = 6$	$\mu = 1$	$\mu = 2$	$\mu = 3$
AG-1	280.7	282.3	275.8	289.1	280.6	275.7
AG-2	281.1	237.9	237.4	243.1	238.0	238.0
AG-3	278.6	223.8	223.4	224.0	220.9	218.6
AG-4	282.1	213.7	213.3	219.3	212.4	**211.2**

The values show the validation errors with regard to the square loss $E^* = E/10^5$ [MW^2]. The persistence model achieves an error of $E^* = 280.6$

Furthermore, we test patterns that again contain the differences of the corresponding measurements for $\mu = 1, 2, 4$. In this way, it is possible to compare the results for $\mu = 2$ with and without wind power changes. In addition, we can compare the error when the patterns have the same dimensionality, which is the case for $\mu = 6$ and $\mu = 3$ when using only absolute values and changes respectively.

We train every model using the first nine months of 2006. For the quantitative evaluation on the last quarter of 2006, we determine the MSE E of the forecasts $f(\mathbf{x}^{park})$ with the measured power outputs y^{park}. Again, we compare our results with the persistence model PST that achieves an error of $E = 280.6 \cdot 10^5 [MW^2]$.

The two parts of the Table 5 show that the prediction is more precise for the aggregations AG-1, AG-3 and AG-4 and for ($\mu \geq 2$) if the patterns include the corresponding changes. It appears that the sum of the individual simple predictions (AG-1) leads to the worst results, comparable with those of the persistence model. If the individual predictions are made with neighboring turbines (AG-2), the persistence model is clearly outperformed. The aggregation considering all features at once (AG-3) leads to an even better accuracy than AG-2. The best results for prediction is achieved by aggregation AG-4, which builds up one time series for the entire park by summing the power values of the single turbines. As we focus on prediction models for a park, this aggregation step pays off. Another advantage is the fact that the data set is reduced and the used regression technique needs a much shorter runtime than using all individual features.

8.4 Prediction with k-Nearest Neighbors

The best parameter for the kNN model is determined with a four-fold cross-validation by testing neighborhood sizes of $k = \{3, 10, 20, 40, 80\}$. Table 6 serves as a direct comparison to Table 5. For the aggregations AG-1, AG-3 and AG-4, including the changes of measurements into the patterns is helpful when using kNN regression. While the results for AG-2 are slightly worse than for AG-1 with $\mu \geq 2$, no good predictions are achieved by the setup AG-3.

Table 6 Employing a 30 min forecast for a park with 25 turbines with *k*NN

	E^* $[MW^2]$					
	Absolute			Absolute + changes		
	$\mu = 0$	$\mu = 2$	$\mu = 6$	$\mu = 1$	$\mu = 2$	$\mu = 3$
AG-1	292.1	281.4	290.6	283.0	278.8	280.4
AG-2	275.0	295.1	339.0	269.3	281.5	296.0
AG-3	288.8	332.5	394.1	292.9	315.2	335.9
AG-4	282.7	215.7	241.5	220.0	**213.5**	216.6

The table shows the validation error $E^* = E/10^5$ [MW^2]. For the error of the persistence model, see caption of Table 5

In conclusion, one should note that here the AG-4 achieves the highest accuracy, too. The errors achieved are almost as low as for the SVR technique. This is remarkable, since *k*NN performed significantly worse in the prediction for a single turbine, see Sect. 7.

9 Conclusions

A precise short-term wind power prediction is important for a safe and sustainable balancing of the electricity grid. This work focuses on the statistical wind power forecast for an individual turbine and an entire park with a horizon of 30 min. The most important result is that predictions with the highest accuracy are achieved for both setups with the SVR technique. For the park, it is with respect to the accuracy and the performance of the prediction advantageous to build one aggregated times series, formed by the summation of the power outputs of all individual turbines.

A direct comparison of our results to other models is a quite difficult undertaking, in particular, because other models would have to use exactly the same data. Further, there is no standard for measuring the performance of wind energy prediction models, which has often been criticized in literature [2]. However, the objective of prediction models is to generally outperform the persistence model in terms of MSE. We do this in a convincing way: Our final approach for the park achieves an accuracy that is 24 % better than the persistence model. We expect that tuning parameters of the SVR models further improves the results. This will be subject to our future research activities.

Acknowledgments We thank the presidential chair of the University of Oldenburg, the EWE research institute NextEnergy, and the Ministry of Science and Culture of Lower Saxony for partly supporting this work. Further, we thank the US National Renewable Energy Laboratory (NREL) for providing the wind data set.

References

1. Catalao, J.P.S., Pousinho, H.M.I., Mendes, V.M.F.: An artificial neural network approach for short-term wind power forecasting in Portugal. In: 15th International Conference on Intelligent System Applications to Power Systems (2009)
2. Costa, A., Crespo, A., Navarro, J., Lizcano, G., Feitosa, H.M.E.: A review on the young history of the wind power short-term prediction. Renew. Sustain. Energy Rev. **12**(6), 1725–1744 (2008)
3. Ernst, B., Oakleaf, B., Ahlstrom, M., Lange, M., Moehrlen, C., Lange, B., Focken, U., Rohrig, K.: Predicting the wind. Power Energy Mag. **5**(6), 78–89 (2007)
4. Focken, U., Lange, M., Mönnich, K., Waldl, H., Beyer, H., Luig, A.: Short-term prediction of the aggregated power output of wind farms—a statistical analysis of the reduction of the prediction error by spatial smoothing effects. J. Wind Eng. Ind. Aerodyn. **90**(3), 231–246 (2002)
5. Foresti, L., Tuia, D., Kanevski, M., Pozdnoukhov, A.: Learning wind fields with multiple kernels. Stoch. Env. Res. Risk Assess. **25**(1), 51–66 (2011)
6. Heinermann, J., Kramer, O.: Precise wind power prediction with SVM ensemble regression. In: Artificial Neural Networks and Machine Learning—ICANN 2014, pp. 797–804. Springer, Switzerland (2014)
7. Han, S., Liu, Y., Yan, J.: Neural network ensemble method study for wind power prediction. In: Asia Pacific Power and Energy Engineering Conference (APPEEC) (2011)
8. Juban, J., Fugon, L., Kariniotakis, G.: Probabilistic short-term wind power forecasting based on kernel. In: Density Estimators. European Wind Energy Conference, pp. 683–688. IEEE (2007)
9. Kramer, O., Gieseke, F., Heinermann, J., Poloczek, J., Treiber, N.A.: A framework for data mining in wind power time series. In: Proceedings of ECML Workshop DARE (2014)
10. Kramer, O., Gieseke, F.: Short-term wind energy forecasting using support vector regression. In: 6th International Conference on Soft Computing Models in Industrial and Environmental Applications (2011)
11. Kramer, O., Gieseke, F.: Analysis of wind energy time series with kernel methods and neural networks. In: 7th International Conference on Natural Computation (2011)
12. Kramer, O., Treiber, N.A., Sonnenschein, M.: Wind power ramp event prediction with support vector machines. In: 9th International Conference on Hybrid Artificial Intelligence Systems (2014)
13. Mohandes, M.A., Rehmann, S., Halawani, T.O.: A neural networks approach for wind speed prediction. Renew. Energy **13**(3), 345–354 (1998)
14. Lew, D., Milligan, M., Jordan, G., Freeman, L., Miller, N., Clark, K., Piwko, R.: How do wind and solar power affect grid operations: the western wind and solar integration study. In: 8th International Workshop on Large Scale Integration of Wind Power and on Transmission Networks for Offshore Wind Farms (2009)
15. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
16. Pöller, M., Achilles, S.: Aggregated wind park models for analyzing power system dynamics. In: 4th International Workshop on Large-scale Integration of Wind Power and Transmission Networks for Offshore Wind Farms, Billund (2003)
17. Poloczek, J., Treiber, N.A., Kramer, O.: KNN regression as geo-imputation method for spatio-temporal wind data. In: 9th International Conference on Soft Computing Models in Industrial and Environmental Applications (2014)
18. Robusto, C.C.: The Cosine-Haversine formula. Am. Math. Mon. **64**(1), 38–40 (1957)
19. Soman, S.S., Zareipour, H., Malik, O., Mandal, P.: A review of wind power and wind speed forecasting methods with different time horizons. In: North American Power Symposium (NAPS), pp. 1–8 (2010)
20. Treiber, N.A., Kramer, O.: Evolutionary turbine selection for wind power predictions. In: 37th Annual German Conference on AI, pp. 267–272 (2014)

21. Treiber, N.A., Heinermann, J., Kramer, O.: Aggregation of features for wind energy prediction with support vector regression and nearest neighbors. In: European Conference on Machine Learning, DARE Workshop (2013)
22. Treiber, N.A., Kramer, O.: Wind power prediction with cross-correlation weighted nearest neighbors. In: 28th International Conference on Informatics for Environmental Protection (2014)
23. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
24. Van der Walt, S., Colbert, S.C., Varoquaux, G.: The numpy array: a structure for efficient numerical computation. Comput. Sci. Eng. **13**(2), 22–30 (2011)
25. Wegley, H., Kosorok, M., Formica, W.: Subhourly wind forecasting techniques for wind turbine operations. Technical report, Pacific Northwest Lab., Richland, WA (USA) (1984)

Springer

http://www.springer.com/978-3-319-31856-1

Computational Sustainability
Lässig, J.; Kersting, K.; Morik, K. (Eds.)
2016, VI, 276 p. 98 illus., 75 illus. in color., Hardcover
ISBN: 978-3-319-31856-1