Very Short-Term Wind Speed Forecasting Using Spatio-Temporal Lazy Learning

Annalisa Appice[1(✉)], Sonja Pravilovic[2], Antonietta Lanza[1],
and Donato Malerba[1]

[1] Dipartimento di Informatica, Università degli Studi di Bari Aldo Moro
via Orabona, 4, 70126 Bari, Italy
{annalisa.appice,antonietta.lanza,donato.malerba}@uniba.it
[2] Faculty of Information Technology, Mediterranean University
Vaka Djurovica b.b., 81000 Podgorica, Montenegro
sonja.pravilovic@unimediteran.net

Abstract. A wind speed forecast corresponds to an estimate of the upcoming production of a wind farm. The paper illustrates a variant of the Nearest Neighbor algorithm that yields wind speed forecasts, with a fast time resolution, for a (very) short time horizon. The proposed algorithm allows us to monitor a grid of wind farms, which collaborate by sharing information (i.e. wind speed measurements). It accounts for both spatial and temporal correlation of shared information. Experiments show that the presented algorithm is able to determine more accurate forecasts than a state-of-art statistical algorithm, namely auto. ARIMA.

1 Introduction

The growing integration of wind farms into the power grid requires precise forecasts of upcoming energy productions at different time scales, depending on the intended application. Very short-term forecasts (≤ 6 h) can be used for the turbine active control, short-term forecasts (48–72 h) may serve for wind power scheduling, as well as for economic dispatch, while longer time scales (up to 5–7 days ahead) may be considered for planning the maintenance of wind farms and transmission lines. Depending on the nature information processed, wind forecasting approaches can be classified into physical, statistical and hybrid approaches. A physical approach [1,6] uses weather forecast, while a statistical approach [2,8–10,13]) is based on vast amount of historical data (time series) without considering meteorological conditions. A hybrid approach [4,7]) uses both weather forecasts and time series analysis.

In this paper, we address the problem of very short-term wind speed forecasting by resorting to a statistical approach. We describe a spatio-temporal lazy learning-based algorithm, called WiNN (spatio-temporal WInd Nearest Neighbor algorithm), which is based on a variant of the Nearest Neighbor algorithm, which accounts for two forms of autocorrelation: the temporal autocorrelation,

which may exist in wind speed time series data, and the spatial autocorrelation, which may exist between wind speed data measured at near wind farms. For every wind farm, lazy learning is performed to produce forecasts from wind speed values observed in a given spatio-temporal neighborhood. The main advantage of lazy learning is that the computation of a complex forecasting model of the historical data *à la* ARIMA is avoided. In addition, it can be easily extended to account for spatial autocorrelation as well.

The paper is organized as follows. In Sect. 2, we illustrate the data scenario and formulate the learning problem considered. In Sect. 3, we present the proposed algorithm, while in Sect. 4, we discuss the dataset and the relevant results. Finally, Sect. 5 draws some conclusions and outlines some future work.

2   Data Setting and Learning Problem

Data Scenario A wind farm grid is defined as a geophysical streaming system (K, Z, T), where: (1) K is the set of wind farms spanned on a bi-dimensional[1] XY representation of the geographic space, (2) Z is the wind speed variable and (3) T is the time line discretized in equally spaced time points denoted as $1, 2, \ldots, t, \ldots$. In this data scenario, $z(k, t)$ denotes a measure of Z collected from a certain wind farm $k \in K$, at a specific time point $t \in T$. A *wind speed stream* $z(k)$ is the stream of measures $z(k, t)$ collected at wind farm $k \in K$ for each time point $t \in T$, that is, $z(k) = z(k, 1), z(k, 2), \ldots, z(k, t), \ldots$. Following the *sliding window model* [3], $z(k)$ is decomposed into consecutive sliding windows of equal size w, namely $z(k) = z(k, 1 \rightarrow w), z(k, 2 \rightarrow w + 1), \ldots, z(k, t - w + 1 \rightarrow t), \ldots$, where $z(k, t - d - w + 1 \rightarrow t - d)$ denotes the *backward data window* of wind farm k at time t, with backward horizon w and temporal delay $d = 0, 1, \ldots, t - w$.

Forecasting Problem. Given a wind farm grid (K, Z, T) and a time horizon w, a forecasting service aims at producing, at each time point t and for every wind farm $k \in K$, the predictions of upcoming w values of Z, which are henceforth denoted as $\hat{z}(k, t + 1), \hat{z}(k, t + 2), \ldots, \hat{z}(k, t + w)$. In this study, the forecasting service for the upcoming w measurements of Z is based on the backward w measurements of Z collected over the grid. Backward data can be selected with a possible temporal delay that is at worst w-sized.

3   WiNN

WiNN is a lazy learning algorithm that allows us to yield (very-) short term forecasting of wind speed in a wind farm grid (K, Z, T). Input parameters of the algorithm are a spatial radius r, a window size w and a similarity threshold δ. The top-level description of the algorithm is reported in Algorithm 1.

First, for every target wind farm $k \in K$, WiNN applies a spatial filter, in order to determine a spatial neighborhood with center k and radius r (Algorithm 1, lines 1–3).

---

[1] Multi-dimensional representation of geographic space can be equally dealt.

Algorithm 1. WiNN(K, Z, T)

1: **for** $k \in K$ **do**
2:     compute $\sigma(k, r)$ {Definition 31}
3: **end for**
4: **for** $t \in T$ **do**
5:     **for** $k \in K$ **do**
6:         **for** $d \in 1, \ldots, w$ **do**
7:             compute $\tau(k, t, -d)$ {Definition 32}
8:             $\eta(k, t, -d) \leftarrow \sigma(k) \cap \tau(k, t, -d)$ {Definition 33}
9:         **end for**{Forecasting}
10:         **for** $f \in 1, \ldots, w$ **do**
11:             compute $\mathcal{L}(k, t + f)$ {Definition 34}
12:             $\hat{z}(k, t + f) \leftarrow \text{knn}(\mathcal{L}(k, t + f))$ {Definition 35}
13:         **end for**
14:     **end for**
15: **end for**

Definition 1 (Spatial Neighborhood). *The spatial neighborhood $\sigma(k, r)$ is the set of reference wind farms of K ($\sigma(k, r) \subseteq K$) such that:*

$$\sigma(k) = \{h \in K | h \neq k \text{ and } geoDistance(k, h) \leq r\}, \tag{1}$$

where $geoDistance(k, h)$ is the geographic distance computed between the spatial coordinates $xy(k)$ and $xy(h)$, respectively.

This phase is performed when the processing of the wind data streams produced by (K, Z, T) starts. It is repeated only when changes occur in the grid structure, i.e. a farm is either deleted from or added to the grid. After this initialization phase, WiNN processes data (wind speed measurements) as they are produced by the grid. At every streaming time point, it applies a temporal filter to every target farm (Algorithm 1, line 7), in order to identify neighbor farms whose backward data are correlated, with a temporal delay, to the backward data measured by the target farm at the present time (Algorithm 1, line 8). The temporal delay d ranges between 1 and w.

Definition 2 (Temporal Neighborhood). *Let d be a positive, integer-valued temporal delay with $d \leq w$. The temporal neighborhood $\tau(k, t, -d)$ is the set of reference wind farms of K ($\tau(k, t, -d) \subseteq K$) such that:*

$$\tau(k, t, -d) = \{h \in K | h \neq k \text{ and } dataDistance(k, h, t, -d, w) \leq \delta\}, \tag{2}$$

where $dataDistance(k, h, t, -d, w)$ is the Euclidean distance computed between the target backward data $z(k, t - w + 1 \rightarrow t)$ and the reference, d-delayed backward data $z(h, t - d + w + 1 \rightarrow t - d)$, that is

$$dataDistance(k, h, t, -d, w) = \sum_{i=1}^{w} \left(z(h, t - d - w + i) - z(k, t - w + i) \right)^2.$$

In this study, the definition of this kind of neighborhood with a temporal delay is motivated by the characteristics of the physical variable (wind speed) that we are considering for the forecasting problem. Wind can be considered as a moving object over space, so it is reasonable that the wind speed measured from the target wind farm k at time point t is more similar (i.e. higher correlated) to the wind speed measured from a reference wind farm h at a time point before t (i.e. $t - d$) rather than to the wind speed measured from h at t. In this study, we construct neighborhoods with various temporal delay values, in order to be able to properly model wind as a moving object also without accounting for information concerning the wind direction.
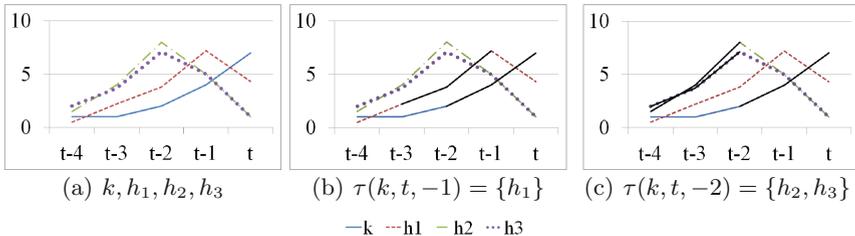
Fig. 1. Example of temporal neighborhoods constructed with window size $w = 3$, time delay $d = 1$ (1(b)) and $d = 2$ (1(c)) for target wind farm k by considering reference wind farms h_1, h_2 and h_3, respectively (1(a)).

Example 1. *Let us consider one target wind farm k, as well as three reference wind farms, namely, h_1, h_2 and h_3 which measure wind speed data processed with window size $w = 3$ (Fig. (1(a))). We can construct a temporal neighborhood of k with time delay $d = -1$ by comparing $z(k, t - w + 1 \rightarrow t)$ to $z(h_1, t - w \rightarrow t - 1)$, $z(h_2, t - w \rightarrow t - 1)$ and $z(h_3, t - w \rightarrow t - 1)$ (Fig. 1(b)), while we can construct the temporal data neighborhood of k with time delay $d = -2$ by comparing $z(k, t - w + 1 \rightarrow t)$ to $z(h_1, t - w - 1 \rightarrow t - 2)$, $z(h_2, t - w - 1 \rightarrow t - 2)$ and $z(h_3, t - w - 1 \rightarrow t - 2)$ (Fig. 1(c)).*

A spatial-temporal neighborhood is built by applying a spatial filter and a temporal filter in cascade. WiNN builds, for every wind farm $k \in K$, w spatial-temporal neighborhoods, one for every delay $d = 1, 2, \ldots, w$ (Algorithm 1, line 8).

Definition 3 (Spatio-Temporal Neighborhood). *The spatio-temporal neighborhood $\eta(k, t, -d)$ is the set of reference wind farms of K ($\eta(k, t, -d) \subseteq K$), which satisfy both the spatial neighborhood filter $\sigma(k)$ (Definition 1) and the temporal neighborhood filter $\tau(k, t, -d)$ (Definition 2) simultaneously, that is, $\eta(k, t, -d) = \sigma(k) \cap \tau(k, t, -d)$.*

Subsequently, spatio-temporal neighborhoods constructed with temporal delays $d = 1, 2, \ldots, w$ (Algorithm 1, line 11) are processed, in order to populate d learning datasets $\mathcal{L}(k, t + 1), \mathcal{L}(k, t + 2), \ldots, \mathcal{L}(k, t + w)$.

Definition 4 (Lazy Learning Data Set). *Let* $\{\eta(k,t,-1)\}_{d=1,2,\ldots,w}$ *be the set of spatio-temporal neighborhoods (Definition 3) associated with $k \in K$ at time t and constructed with the temporal delay $d = 1, 2, \ldots w$, respectively. The learning data set $\mathcal{L}(k,t+d)$ is the set of timestamped data points (reference farm, timestamp, measured wind speed), that is defined as follows:*

$$\mathcal{L}(k,t+d) = \bigcup_{f \geq d} \{(h, t-f+d, z(h, t-f+d)) | h \in \eta(k, t-f)\}.$$

Every learning set $\mathcal{L}(k, t+d)$, with $d = 1, 2, \ldots, w$, is constructed in order to forecast $\hat{z}(k, t+d)$. Lazy learning is performed by resorting to a spatio-temporal version of the Nearest Neighbour formula (Algorithm 1, lines 12). A spatio-temporal distance is computed, in order to estimate the weight according to any sampled backward data point can contribute to the forecast value.

Definition 5 (k-NN). *The forecast value $\hat{z}(k, t+d)$ is determined as follows:*

$$\hat{z}(k, t+d) = \frac{\displaystyle\sum_{(h,t',z)\in\mathcal{L}(k,t+d)} \omega((k,t+d),(h,t'))z(h,t')}{\displaystyle\sum_{(h,t',z)\in\mathcal{L}(k,t+d)} \omega((k,t+d),(h,t'))}, \qquad (3)$$

where $\omega((k,t+d),(h,t')) = \frac{1}{st((k,t+d),(h,t'))^3}$ *and* $st((k,t+d),(h,t')) = \frac{1}{2}scaled_{01}$ $(d(k,h)) + \frac{1}{2}scaled_{01}d(t+d,t'))$.
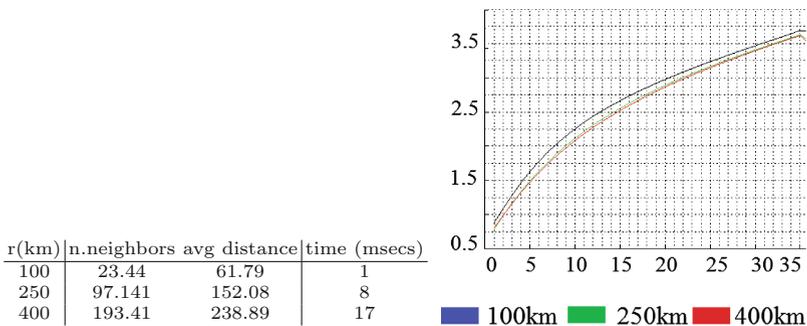
It is noteworthy that $st(\cdot,\cdot)$ is computed as the sum of the scaling in the interval [0,1] of the distance $(d(k,h) = geoDistance(k,h))$ computed between the geographic coordinates of target wind farm k and neighbor reference farm h, as well as of the scaling in the interval [0,1] of the distance $(d(t+d,t') = t+d-t')$ computed between the timestamps associated to the forecast value $(t+d)$ and to the sampled neighbor (t').

4   Experimental Study

The experiments have been carried out using real world data publicly provided by the DOE/NREL/ALLIANCE3 (http://www.nrel.gov/). The data consist of wind speed measurements from 1326 different locations at 80 m of height in the Eastern region of the US. The data were collected in 10 min intervals during the year of 2004 (time line). This wind farm grid is able to produce 580 GW, and each farm produces between 100 MW and 600 MW. Experiments are run on an Intel(R) Core(TM) i7 920 @2.67GHz running Windows 7 Professional. In this study, we have evaluated the sensitivity of both accuracy and efficiency of WiNN to the set-up of the spatial radius. In addition, we have analyzed the accuracy of WiNN compared to that of auto.ARIMA [5]. The selected competitor is a state-of-art statistical forecasting algorithm. Both algorithms have been evaluated in

(very-) short forecasting setting. Forecasts have been produced by considering a time horizon of six hours with wind speed forecasting performed every 10 min ($w = 36$). δ is automatically determined for each considered temporal delay d, as a percentage ($\delta\% = 10\%$) of the maximum Euclidean distance computed between each pair of backward time series, selected for every farm of the grid, at time t and with delay d.

Evaluating WiNN. We have considered the entire wind farm grid, in order to analyze the size of constructed spatial neighborhoods, the average learning time spent, at each time point t, to forecast upcoming wind speed values (over six hours), as well as the accuracy of produced forecasts. Learning time is measured in milliseconds and averaged on the number of time points, as well as on the number of wind farms processed. WiNN is run repeatedly with a radius r varying between 100 km, 250 km and 400 km. Figure 2 (left side) reports the average learning time spent by every wind farm, in order to complete the considered forecasting task. We observe that by increasing the radius, the number of reference wind farms processed, as well as the average distance between neighbor farms increase. Learning times increase accordingly, but they are greatly lower than 10 min. Thus, the forecasting service deployed with WiNN can work in (near-) real time. Figure 2 reports the root mean squared error of the forecasts produced from the wind farm grid. Errors are calculated from the forecasts produced at each time point of the considered time line, for the forward 36 time points (6 h). We observe that the forecasting error decreases by increasing the number of reference neighbor farms processed. However, the reduction of forecasting error is negligible when the radius of spatial neighborhoods is enlarged from 250 Km to 400 km, while the average learning time doubles (from 8 msec to 17 msec) in the same case. Hence, we consider that the choice $r = 250$ km can guarantee an acceptable trade-off between accuracy and efficiency.

r(km)	n.neighbors	avg distance	time (msecs)
100	23.44	61.79	1
250	97.141	152.08	8
400	193.41	238.89	17

Fig. 2. WiNN: radius r varying among 100 km, 250 km and 400 km. Left side: size of constructed spatial neighborhoods (column 2), average geographic distance between spatial neighbors (column 3) and learning time spent in millisecs (column 4) to complete forecasting. Right side: average rmse (axis Y) of forecasts produced from the entire grid of wind farms at time $t+d$ with $t \in T$ and $d = 1, 2, \ldots, 36$ (axis X) (Color figure online).
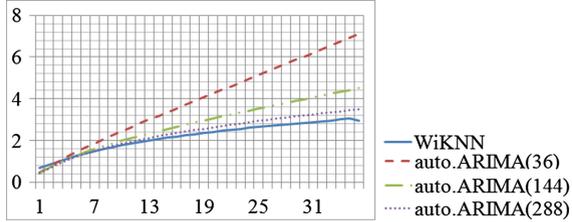
Fig. 3. WiNN vs auto.ARIMA: analysis of the forecasting accuracy measured per forecasting time point.

Comparing WiNN with auto.ARIMA. We have considered 15 wind farms randomly selected across the grid. For every selected wind farm, we have compared forecasting errors of WiNN with forecasting errors of auto.ARIMA. WiNN is run by setting radius r equal to 250 Km and window size w equal to 36. This means that it forecasts, at each time point of the considered time line, the upcoming 36 values of the wind speed and uses backward windows of size equal to 36, in order to determine these forecasts. Similarly to WiNN, the competitor auto.ARIMA forecasts, at each time point of the considered time line, upcoming 36 values of wind speed. These forecasts are determined by considering backward historical data with size ranging between 36 (six hours), 144 (24 h) and 288 (48 h). Figure 3 reports the root mean squared error of the forecasts produced from the fifteen selected wind farms for the upcoming 36 time points (6 h). Errors are calculated from the forecasts produced at each time point of the considered time line. Results show that auto.ARIMA outperforms WiNN if we consider the forecasts of the wind speed associated with the future time points close to t (i.e. $t+1, t+2, t+3, t+4$ and $t+5$), but WiNN outperforms auto.ARIMA if we consider the forecasts associated with the future time points distant from t ($t+6, t+7, \ldots, t+36$). Additionally, auto.ARIMA can produce lower errors, which are closer to the errors produced by WiNN, only by augmenting the amount of historical data to be learned. These results confirm the efficacy of dealing with spatial information when forecasting wind speed in a wind farm grid. Results also show the feasibility of the lazy learning approach that we have formulated for this forecasting task.

5   Conclusion

This paper presents a data mining algorithm that resorts to a spatio-temporal variation of Nearest Neighbor algorithm in order to produce accurate forecasting of very short term wind speed values in a wind farm grid. The efficacy of the proposed algorithm is compared to that of a state-of-art statistical model. As future work, we plan to investigate the combination of the neighborhood construction described in this paper with the spatial aware versions of the ARIMA model described in [11,12].

References

1. Albert, W., Chi, S., J.H., C.: An improved grey-based approach for electricity. Elect. Power Syst. Res. **67**, 217–224 (2003)
2. Almeida, V., Gama, J.: Collaborative wind power forecast. In: Bouchachia, A. (ed.) ICAIS 2014. LNCS, vol. 8779, pp. 162–171. Springer, Heidelberg (2014)
3. Gaber, M.M., Zaslavsky, A., Krishnaswamy, S.: Mining data streams: a review. ACM SIGMOD Rec. **34**(2), 18–26 (2005)
4. Giebel, G., Badger, J., Perez, I.M., Louka, P., Kallos, G.: Short-term forecasting using advanced physical modelling-the results of the anemos project. Results from mesoscale, microscale and CFD modelling. In: EWEA 2006 (2006)
5. Hyndman, R., Khandakar, Y.: Automatic time series forecasting: the forecast package for R. J. Stat. Softw. **26**(3), 1–22 (2008)
6. Lange, M., Focken, U.: New developments in wind energy forecasting. Power Eng. Soc. IEEE Gen. Meet. **2008**, 1–8 (2008)
7. Negnevitsky, M., Johnson, P., Santoso, S.: Short term wind power forecasting using hybrid intelligent systems. Power Eng. Soc. IEEE Gen. Meet. **2007**, 1–4 (2007)
8. Ohashi, O., Torgo, L.: Wind speed forecasting using spatio-temporal indicators. In: ECAI 2012 - PAIS 2012 (System Demonstrations Track). Frontiers in Artificial Intelligence and Applications, vol. 242, pp. 975–980. IOS Press (2012)
9. Potter, C., Negnevitsky, M.: Very short-term wind forecasting for tasmanian power generation. IEEE Trans. Power Syst. **21**(2), 965–972 (2006)
10. Pravilovic, S., Appice, A., Lanza, A., Malerba, D.: Wind power forecasting using time series cluster analysis. In: Džeroski, S., Panov, P., Kocev, D., Todorovski, L. (eds.) DS 2014. LNCS, vol. 8777, pp. 276–287. Springer, Heidelberg (2014)
11. Pravilovic, S., Appice, A., Malerba, D.: An intelligent technique for forecasting spatially correlated time series. In: Baldoni, M., Baroglio, C., Boella, G., Micalizio, R. (eds.) AI*IA 2013. LNCS, vol. 8249, pp. 457–468. Springer, Heidelberg (2013)
12. Pravilovic, S., Appice, A., Malerba, D.: Integrating cluster analysis to the ARIMA model for forecasting geosensor data. In: Andreasen, T., Christiansen, H., Cubero, J.-C., Raś, Z.W. (eds.) ISMIS 2014. LNCS, vol. 8502, pp. 234–243. Springer, Heidelberg (2014)
13. Sideratos, G., Hatziargyriou, N.: An advanced statistical method for wind power forecasting. IEEE Trans. Power Syst. **22**(1), 258–265 (2007)

Springer