Analysis of Wind Energy Time Series with Kernel Methods and Neural Networks

Oliver Kramer
Bauhaus-University Weimar
Institute of Structural Mechanics
99423 Weimar, Germany
oliver.kramer@uni-weimar.de

Fabian Gieseke
Bauhaus-University Weimar
Institute of Structural Mechanics
99423 Weimar, Germany
fabian.gieseke@uni-weimar.de

***Abstract*—Wind energy has an important part to play as renewable energy resource in a sustainable world. For a reliable integration of wind energy the volatile nature of wind has to be understood. This article shows how kernel methods and neural networks can serve as modeling, forecasting and monitoring techniques, and, how they contribute to a successful integration of wind into smart energy grids. First, we will employ kernel density estimation for modeling of wind data. Kernel density estimation allows a statistically sound modeling of time series data. The corresponding experiments are based on real data of wind energy time series from the NREL western wind resource dataset. Second, we will show how prediction of wind energy can be accomplished with the help of support vector regression. Last, we will use self-organizing feature maps to map high-dimensional wind time series to colored sequences that can be used for error detection.**

I. Introduction

Sustainability is of great importance due to increasing demands and limited resources. In particular, in the field of energy production and consumption, methods are required that improve efficiency. The extension of renewable energy sources, and the growing information structure allow a fine screening of energy resources, but also require the development of tools for the analysis and understanding of huge datasets about the energy grid. Wind has an important part to play in sustainable energy systems. An essential aspect is to understand *when* and *how strong* the wind is blowing for a single windmill, or a whole wind park. A precise understanding and forecast of wind energy allows to balance the grid, e.g., to control spinning reserves. Advanced monitoring techniques for wind time series data are required to understand changes, and to recognize errors in the grid.

In this article we will show that kernel methods and neural networks can play a key role in modeling and forecasting of wind energy time series. We hope to demonstrate the potential of these techniques in contributing to a sustainable system development.

II. Wind Data

The wind time series data that is basis of our analysis is taken from the NREL western wind resources dataset [9], [14]. The western wind resources dataset is part of the Western Wind and Solar Integration Study, which is a large regional wind and solar integration study in the United States of America. The data has been designed to perform temporal and spatial comparisons like load correlation, or estimation of production from hypothetical wind farms for demand analysis and planning of storage based on wind variability, or economic calculations comparing in-state versus out-of-state costs of delivered energy. The data was sampled every ten minutes (resulting in 52,560 measurements a year), and every two kilometers; 1.2 million grid points have been aggregated to 32,043 locations. Each grid point is estimated to hold ten Vestas 3 MW turbines, and therefore the 32,043 locations in total exhibit more than 960 GW of capacity. The set contains data of 2004, 2005 and 2006. The GUI of the western wind dataset allows to select grid points and download corresponding times series data.

For forecasting and sequence visualization the wind data is aggregated to a time series training set in a preprocessing step. This training set has a vertical, and a horizontal component:

- vertical (rows): the vertical axis corresponds to the choice of grid points, the number of grid points determines the dimension of the dataset
- horizontal (lines): the horizontal axis corresponds to the chosen time window (e.g. aggregate hours, days, etc.)

In the following we treat the wind data as a time series of N wind measurements of K grid points, i.e,. $\mathbf{x}_1, \ldots, \mathbf{x}_N$ with $\mathbf{x}_i = (x_1, \ldots x_K)^T$.

III. Wind Modeling with Kernel Smoothing Methods

For simulation of smart grids modeling of wind is an essential task. Modeling wind resources simply based on raw measured data is a possible, but restrictive way. In this section we employ an alternative: modeling wind with a kernel density estimation approach, more precisely the Parzen window estimator [13]. It is a kernel density method that can be used for statistically sound sampling based on observed data. We will introduce the Parzen window estimator, and will model observed data both from a single grid point in Tehachapi, and a wind park near Salt Lake City.

A. Kernel Density Estimation

Kernel smoothing methods are based on kernel density functions that are introduced in this section. Kernel density

978-1-4244-9952-6/11/$26.00 ©2011 IEEE

functions measure densities in data space. They are closely related to histograms [3]. Kernel density estimation is a step towards smoother density estimates. Let $\mathcal{D} = \{x_1, \ldots, x_N\}$ be the observations, and we seek for the density function $p(x)$ of the data distribution. Kernel density estimation is a non-parametric approach. The kernel density approximation of the probability density function is

$$\hat{p}(x) = \frac{1}{Nh} \sum_{i}^{N} K\left(\frac{x - x_i}{h}\right) \tag{1}$$

with kernel K and the so-called bandwidth parameter h. The bandwidth h has a similar meaning like the width of histogram bins, as they define the width of the influence of kernel K.

A typical kernel function is the Gaussian kernel:

$$K_G(z) = \frac{1}{(2\pi)^{q/2}\det(\mathbf{H})} \exp\left(-\frac{1}{2}\left|\mathbf{H}^{-1}z\right|^2\right). \tag{2}$$

Here, the bandwidth $\mathbf{H} = \mathrm{diag}(h_1, h_2, \ldots, h_d)$ is a diagonal matrix with entries that determine the standard deviation for each dimension.

Sampling from the Parzen estimator is possible by solving

$$(Nh)^{-1} \sum_{i=1}^{N} \int_{-\infty}^{a} K\left(\frac{x - x_i}{h}\right) dx = r \tag{3}$$

for a uniformly distributed number $r \in [0, 1]$. Simpler than computing the integral, and solving the equation is an alternative way to approximate the same density function. Randomly choosing one of N points, and sampling from the corresponding single kernel function with its assigned bandwidth, which is easy to implement for a Gaussian kernel, approximates the same probability density function.

B. Wind Resource Modeling

The Parzen window estimator allows to model the wind resource time series of a single wind grid point or even of a whole wind park. This can be accomplished by treating each time series of a specific time span $\mathbf{x} = (x_1, \ldots, x_N)^T$ as N-dimensional observation (data sample). By putting all N observations into the Parzen estimator, it can model the observations, and in turn allows statistically sound sampling. We will demonstrate how this model can be built on the level of a grid point, and a whole wind park.

We demonstrate the process modeling the wind time series based on samples $\mathbf{x}_i$ from a set of K original time series $\mathcal{S} = (\mathbf{x}_i)$, $i = 1, \ldots, K$. Sampling from the kernel density estimate of all observed time series results in a statistically plausible model of the observations.

We model the time series of a single wind grid point in the Tehachapi park (ID 11743) from January 2006, with the Parzen window estimator. For this sake, we aggregate the 1,440 observations to 24 hourly observations for each day, i.e., six observations are aggregated to one point in the time series. The resulting 31 observations, i.e., 24-dimensional data vectors of 31 days of January 2006, are modeled with the Parzen window estimator.

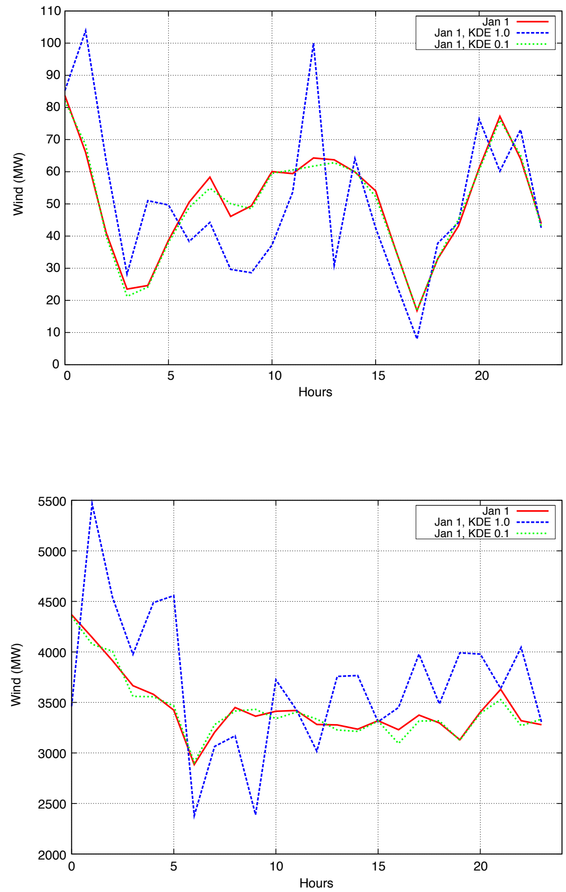

Fig. 1. Wind resources based sampled time series for the wind grid point (upper part), and wind park (lower part), that are closest to the corresponding original time series in January 1, 2006.

Figure 1 shows a sampled time series that is closest to the original time series of January 1, 2006. The curves named KDE[1] 1.0 denote the kernel density estimate with bandwidths based on Silverman's rule-of-thumb [18], KDE 0.1 scaled by 0.1 respectively. *Jan 1* shows the original time series. The sampled time series with scaling factor are closer to the original training instances than the sequences based on Silverman's rule. The lower part of Figure 1 demonstrates the modeling process for a whole wind park. For this sake we use the data of 28 grid points of the park southeast of Salt Lake city. For each time step the grid point data of 28 points, and six time steps is aggregated to one measurement. Similar to the single wind grid point case the model is close to the original time series. Stronger oscillations of wind in case of the Silverman-setting are due to the fact that each time step is independently sampled from the *whole* distribution of wind time series.

[1] KDE: short for kernel density estimation

IV. Wind Forecasting with Support Vector Regression

Wind energy forecasting is an important aspect for balancing authorities in a smart grid. Up to now, the integration of decentralized energy into the grid is as good as ignored. It is estimated that the stability of the energy grid decreases, if the amount of ignored renewable energy exceeds about 15 to 20%. But wind resources are steadily increasing. For a reasonable integration of volatile resources like wind, a precise prediction for subhourly scheduling becomes necessary. Precise forecast will allow balancing and integrating of multiple volatile power sources at all levels of the transmission and distribution grid [10]. Consequently, the forecast of renewable resources plays an important role for a stable energy grid. The amount of wind power crucially depends on the speed of the wind, as the power is proportional to the cube of the wind speed. Hence, small differences in the wind speed result in significant differences in the produced power.

In this section, we investigate the questions if prediction of wind energy can exclusively be based on the existing infrastructure of windmills, and their wind speed measurements. To answer these questions, we will conduct experiments based on the real-world NREL data employing a state-of-the art kernel regression method, i.e., support vector regression (SVR) proposed by Vapnik [22], which is related to the classification concept of support vector machines [21], [16], [20]. Our analysis will be based on a direct mapping of wind speed measurements on produced wind energy.

We formulate the wind forecasting task as regression problem. Again, we assume that a time series $\mathbf{x}_1, \ldots, \mathbf{x}_N$ of N wind measurements of K wind grid points, and corresponding measurements $y_1, \ldots, y_N$ of wind energy production of a target point is given. The task is to predict the production $\mathbf{y}_t$ at time $t = t_i + \theta$ based on past wind measurements at time $t_i, t_i - 1, t_i - 2, \ldots, t_i - \mu$, with $\mu \in \mathbb{N}$ past observations. The following questions arise:

- how much data from the past do we need (i.e., how to choose μ to reduce the validation error), and
- how far can we look into the future (i.e., how does the validation error depend on θ).

We concentrate on the production on the large-scale level of a whole wind park in Section IV-C.

A. Support Vector Regression

The SVR approach is one of the state-of-the-art methods in regression. The goal of the learning process is to find a prediction function $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$ that assigns "good" predictions to unseen patterns $x \in \mathcal{X}$, where $\mathcal{X}$ is an arbitrary set (e.g., $\mathcal{X} = \mathbb{R}^d$). We refer to, e.g, Smola and Schölkopf [19] for a comprehensive overview of SVR. The SVR technique can be seen as a special case of regularization problems of the form

$$\inf_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} L\big(y_i, f(\mathbf{x}_i)\big) + \lambda ||f||_{\mathcal{H}}^2, \qquad (4)$$

where $\lambda > 0$ is a fixed user-defined real value, $L : \mathbb{R} \times \mathbb{R} \rightarrow [0, \infty)$ a loss function and $||f||_{\mathcal{H}}^2$ the squared norm in a so-called *reproducing kernel Hilbert space*

$$\mathcal{H} \subseteq \mathbb{R}^{\mathcal{X}} = \{f : \mathcal{X} \rightarrow \mathbb{R}\}$$

induced by an associated kernel function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ [15].

B. Related Work

As wind forecasting is an important task, a lot of publications can be found in literature applying different methods. Costa *et al.* [2] review 30 years of short-term prediction concentrating on forecasting methods, mathematical, statistical and physical models, as well as meteorology. Negnevitsky *et al.* [12] review forecasting techniques used for power system applications with focus on electricity load, price forecasting and wind power prediction. Milligan *et al.* [10] discuss, if wind is a capacity resource. They state that for a single wind power plant, predictions on a one or two hours basis can achieve an accuracy level of approximately 5-7% mean absolute error to installed wind capacity, increasing to 20% for day-ahead forecasts.

Preliminary work on SVR, and wind forecasting has recently been introduced. Mohandes *et al.* [11] compared an SVR approach for wind speed prediction to a multi-layer perceptron. The approach is based on mean daily wind speed data from Saudi Arabia. Shi *et al.* [17] proposed an approach that combines an evolutionary algorithm for parameter tuning with SVR-based prediction. The technique allows a six hours prediction, and is experimentally evaluated on wind data from North China. Recently, Zhao *et al.* [24] compared SVR to backpropagation for a ten minutes prediction of wind speed.

Further work concentrates on special aspects like prediction and diagnosis of wind turbine faults. Kusiak and Li [7] introduced an approach based on fault prediction on three levels, e.g., fault category and specific fault prediction in a five minutes to one hour approach.

C. Wind Park Energy Forecast

Large-scale forecasting on the level of wind parks has an important part to play for global control strategies. Besides the approach to aggregate the forecasts of all grid points of the whole park, the sum of energy can be taken into account. We conduct the prediction analysis on the level of a wind park near Salt Lake City that consists of 28 grid points. For the forecasts we employ a set of 100 randomly chosen points in the whole western area.

Figure 2 shows a visualization of two randomly selected sequences, and the corresponding ten minutes and six hours ahead forecasts. The curves show the real wind that was blowing, and the forecasts, each based on two past time steps. The plots show that all forecasts achieve a relatively high prediction accuracy that could be satisfying for most balancing activities in a smart grid. The predictions based on the last two hours are even more reliable based on a ten minutes forecast than the predictions based on the last 30 minutes. Also for the

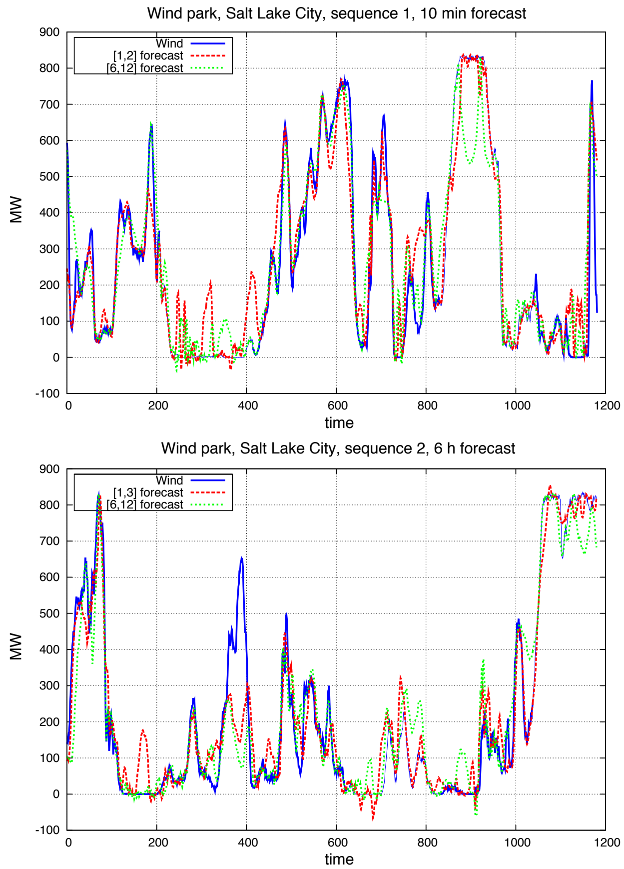

Fig. 2. Ten minutes wind energy forecasts for a wind park southeast of Salt Lake City on two randomly selected sequences.

six hours ahead forecast the prediction based on the [6,12]-dataset results in the best curve. Local deviations from the true curve are more frequent in the case of the [1,3]-dataset forecast.

V. Wind Monitoring with Self-Organizing Feature Maps

As wind is a volatile energy source, state observation has an important part to play for grid management, fault analysis and planning strategies of grid operators. We demonstrate how two approaches from unsupervised neural computation help to understand high-dimensional wind resource time series. The first approach for the visualization of multivariate sequences is based on self-organizing feature maps. The output sequence allows the monitoring of the overall system state with a low-dimensional and linear visualization that reflects the topological characteristics of the original wind data. We demonstrate the visualization on real-world wind resource measurements.

In this section we propose a sequence analysis approach for high-dimensional wind time series based on a quantization process with Kohonen's self-organizing map (SOM) [5]. It generates a topological low-dimensional representation of the high-dimensional data allowing a *smoother* quantization with regard to the data sample distribution than simply dividing the feature space into grids, or distributing codebook vectors equidistantly. A SOM is able to capture the intrinsic structure of the data. It can be used to visualize sequences, or to perform further symbol-oriented analysis like string matching.

A. Related Work

SOM-based visualization has been applied to various applications ranging from financial time series data [23] to GSM Network Performance Data [8]. We have employed a related approach for the visualization of music sequences [4], and employed the SOM-based discretization for recognition of sequences with a dynamic time warping approach for gesture recognition [6]. Using SOMs for monitoring and modeling of complex industrial processes like pulp process, steel production, and paper industry are described by Alhoniemi *et al.* [1].

B. Visualization of Multivariate Wind Time Series

Task of the SOM is to compute a smooth mapping of the high-dimensional time series data to the neurons of the map. By smoothness we denote that the quantization maintains the topology of the data: neighbored high-dimensional data is also neighbored on the low-dimensional map. This characteristic is important for the state monitoring, as smoothly changing colors alleviate the recognition of state changes. We employ a squared two-dimensional SOM, whose size and number of training cycles are stated in each case.

With a trained SOM the sequence of high-dimensional data can be translated to a corresponding sequence of symbols (winner neurons). Let $\mathbf{x}_1, \ldots, \mathbf{x}_N$ be a sequence of high-dimensional feature vectors. Fed to a trained SOM the smooth quantization is generated by computing a sequence of symbols $\sigma_1, \ldots, \sigma_N$ with

$$\sigma_i = \arg \min_{k=1 \ldots K} \|\mathbf{w}_k - \mathbf{x}_i\|_2. \quad (5)$$

The symbols of the sequence are colorized with regard to the corresponding winner neuron n^*, whose color has to be user-defined. According to the topological characteristics of the SOM neighbored neurons are assigned to similar colors in the sequence. A function $f : \mathcal{S} \rightarrow \mathbb{R}^3, \quad \sigma \rightarrow (r_1, r_2, \lambda)$ maps a symbol σ to a 3-dimensional RGB-vector. Values r_1 and r_2 depend on neuron n with corresponding coordinates on the map, while for the third value λ an arbitrary constant can be chosen. With function f a sequence of symbols corresponding to a sequence of winner neurons of the SOM can be translated into a sequence of RGB-vectors.

We have developed an efficient implementation of the SOM-based quantization and visualization method in C. The developed software tool makes use of an XML-file for specification of the SOM training parameters. The features of the training set have to be stored in a simple CSV-format.

C. Identification of Alert States

The upper part of Figure 3 shows an alert state that has artificially been integrated to the data of January 2006. The 28-th data element of the wind resource has been set to 0.0 for all grid points, simulating a sudden blackout of all wind turbines. This alert state can clearly be recognized, it has a suspicious bright color like the area on the right hand side of the plot that represents times of low wind energy production. A training with representative alert states that have to be selected by the grid operator is possible to identify specific situations of the system. The lower part of Figure 3 shows

Fig. 3. Upper sequence: Simulated blackout of wind resources in Tehachapi. The alert state at the 28-th time step can clearly be recognized with a similar color like the low-wind level on at the end of the period. Lower sequence: Training of the SOM with states of interest, and corresponding visualization of a 100 step times series.

the visualization of a sequence that contains training motifs – sequences of special interest – at the beginning. These four segments have artificially been generated and represent (1) a state where no wind blows in the whole system, (2) a state where the wind blows with full speed in the northern half of the park, and not in the southern part, (3) the inverse state of (2), and (4) the situation that the wind blows everywhere with full speed. The subsequent parts of the sequence show where the corresponding situation can be found in the remainder of the sample sequences. Again, similar colors represent states that are similar to the reference motifs. The other colors (like the dark blue sections) do not have much in common with the reference motifs. The marked states can easily be identified, e.g., as alert states.

VI. Conclusion

Kernel methods and neural techniques have proven well in analysis and modeling dynamic systems. This seems natural as neural techniques are motivated by mechanisms of natural neural systems that have to cope with dynamic environments. Wind is a very dynamic system. An important aspect is to manage its volatile and dynamic nature. The integration of wind energy into smart grids affords balancing capabilities, and balancing affords understanding and forecasting.

The examples presented in this work have shown how kernel techniques can help to cope with the dynamics of wind time series data. They turn out to be successful methods in modeling, forecasting and monitoring of wind energy time series data. Efficient implementations allow their application in real-time scenarios. It is subject to future projects to show the success of these and other kernel methods in real-world energy applications.

References

[1] E. Alhoniemi, J. Hollmen, O. Simula, and J. Vesanto. Process monitoring and modeling using the self-organizing map. *Integrated Computer Aided Engineering*, 6:3–14, 1999.

[2] A. Costa, A. Crespo, J. Navarro, G. Lizcano, H. Madsen, and E. Feitosa. A review on the young history of the wind power short-term prediction. *Renewable and Sustainable Energy Reviews*, 12(6):1725–1744, August 2008.

[3] W. Härdle and L. Simar. *Appplied Multivariate Statistical Analysis*. Springer, Berlin, Heidelberg, 2007.

[4] T. Hein and O. Kramer. Recognition and visualization of music sequences using self-organizing feature maps. In *KI*, pages 160–167, 2010.

[5] T. Kohonen. *Self-Organizing Maps*. Springer, 2001.

[6] O. Kramer, B. Satzger, and J. Lässig. Power prediction in smart grids with evolutionary local kernel regression. In *HAIS (1)*, pages 262–269, 2010.

[7] A. Kusiak and W. Li. The prediction and diagnosis of wind turbine faults. *Renewable Energy*, 36(1):16–23, 2011.

[8] P. Lehtimaeki and K. Raivio. A som based approach for visualization of gsm network performance data. In *In Proceedings of the 18th Internation Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, 2005.

[9] D. Lew, M. Milligan, G. Jordan, L. Freeman, N. Miller, K. Clark, and R. Piwko. 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.

[10] M. Milligan, K. Porter, E. DeMeo, P. Denholm, H. Holttinen, B. Kirby, N. Miller, A. Mills, M. O'Malley, M. Schuerger, and L. Soder. Wind power myths debunked. *IEEE Power and Energy Society*, February, 2009.

[11] M. Mohandes, T. Halawani, S. Rehman, and A. A. Hussain. Support vector machines for wind speed prediction. *Renewable Energy*, 29(6):939–947, May 2004.

[12] M. Negnevitsky, P. Mandal, and A. Srivastava. Machine learning applications for load, price and wind power prediction in power systems. In *Intelligent System Applications to Power Systems (ISAP)*, pages 1–6, 2009.

[13] E. Parzen. On the estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33:1065–1076, 1962.

[14] C. W. Potter, D. Lew, J. McCaa, S. Cheng, S. Eichelberger, and E. Grimit. Creating the dataset for the western wind and solar integration study (u.s.a.). In *7th International Workshop on Large Scale Integration of Wind Power and on Transmission Networks for Offshore Wind Farms*, 2008.

[15] B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In D. P. Helmbold and B. Williamson, editors, *Proceedings of the 14th Annual Conference on Computational Learning Theory*, pages 416–426, 2001.

[16] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[17] J. Shi, Y. Yang, P. Wang, Y. Liu, and S. Han. Genetic algorithm-piecewise support vector machine model for short term wind power prediction. In *Proceedings of the 8th World Congress on Intelligent Control and Automation*, pages 2254–2258, 2010.

[18] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*, volume 26 of *Monographs on Statistics and Applied Probability*. Chapman and Hall, London, 1986.

[19] A. J. Smola, B. Schölkopf, and B. Scholkopf. A tutorial on support vector regression, 1998.

[20] I. Steinwart and A. Christmann. *Support Vector Machines*. Springer-Verlag, New York, NY, USA, 2008.

[21] J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999.

[22] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[23] D. Yu, Y. Qi, Y.-H. Xu, and J.-Y. Yang. Kernel-som based visualization of financial time series forecasting. In *ICICIC (2)*, pages 470–473, 2006.

[24] P. Zhao, J. Xia, Y. Dai, and J. He. Wind speed prediction using support vector regression. In *Industrial Electronics and Applications (ICIEA)*, pages 882 – 886, 2010.