

Nowcasting of Wind Speed using Support Vector Regression. Experiments with Time Series from Gran Canaria

I. Espino, M. Hernández

SIANI

Universidad de Las Palmas de Gran Canaria
Campus of Tafira, 35017 Las Palmas, Canary Islands (Spain)
e-mail: {iespino, mhernandez}@siani.es

Abstract. The aim of this paper is to describe and evaluate a proposal for nowcasting wind speed for wind farm locations from historical time series, based on the method of regression by support vectors. To show the improvement over other methods, we used the ANEMOS Project standard evaluation protocol for forecasting against three reference models to compare, referred to a statistical approach: persistence, autoregressive and autoregressive moving average models. The obtained results show a good performance of the proposed method and how beat the classical reference models.

Keywords

Wind speed forecasting, nowcasting, support vector regression.

1. Introduction

In last times, environmental, economic and geostrategic considerations have prompted the use of wind power as a renewable energy resource. The biggest challenge in integrating wind power into the electric grid is its intermittency. As a consequence of the presence of an increasing fraction of renewable energy into the electric grid, its influence has grown into the grid security, system operation, and market economics problems.

Although wind energy may not be dispatched, the negative impact of wind energy onto the grid can be reduced if it can be scheduled using wind forecasting. So, the improvement of the wind power forecasting performance has significant impact on the system operation, obtaining as a consequence increasing wind power penetration without degradation in security or quality of service.

Classification of wind forecasting methods can be organized by time-scale in the following categories [1] with its related main applications:

- *Very short-term forecasting*: range from few seconds to 30 minutes ahead. Its main applications are in electric market clearing or regulation actions.
- *Short-term forecasting*: range from 30 minutes to 6 hours ahead. Its main applications are in economic load dispatch planning or load increment/decrement decisions.
- *Medium-term forecasting*: range from 6 hours to 1 day ahead. Its main applications are in generator online/offline decisions, operational security in day-ahead or electric market.

- *Long-term forecasting*: range from 1 day to 1 week ahead. Its main applications are in unit commitment decisions, reserve requirement decisions or maintenance scheduling to obtain optimal operating costs.

These previous limits of forecasting terms are not strictly defined and some relaxation may be granted depending on the application of the prediction model. Other authors [2] define *nowcasting* to short lead time weather forecast and, particularly wind forecast. The US National Weather Service specifies that zero to three hours ahead forecasting is considered nowcasting. For other agencies, forecasts up to six hours are called nowcasting.

The main wind power forecasting methods developed and reported in literature can be classified into the following approaches:

- *Persistence Approach*, where it is assumed that wind speed at time $t+k$ will be the same at time t . It is more accurate than most of another approaches for nowcasting, effective for very short-term and short-term forecast. Hence, it is considered as a reference and so, it is used as a benchmark to check the improvements of new forecasting solutions [3].
- *Statistical Approach*, which is based on tuning the parameters of a model, training it with historical measurement data. In this approach, the difference between the predicted and the actual model allows to tune the parameters [4]. It includes time-series based models and neural network based methods. Most popular models of time-series approach are autoregressive moving average (ARMA) and variations (ARIMA, ARMAX, ARX, ...). Neural network models can be feedforward networks, multilayer perceptrons, recurrent neural networks, radial basis function networks, etc.
- *Physical Approach*, that uses a numerical model of the physical description of the atmosphere (Numerical Weather prediction, NWP). Usually wind speed is computed on a coarse grid by a weather service at mesoscale and transformed by downscaling at the location of wind farm [5] under forecasting. Customarily, NWP are run several times a day due to the difficulties and high costs associated to gain information in short-time. This limits its usefulness to long-term forecasts (greater than 6 hours ahead). Moreover, the most stable the weather conditions are, the most accurate the predictions are.

- *Hybrid Approach* or combination of different approaches such as mixing physical and statistical, short-term and medium term models, etc. These models can outperform each of them individually.

This paper is oriented to a study of very short-time and short-time forecasting, under statistical approach using Support Vector Regression technique (SVR) and over time series of wind speed, in order to predict future values. To evaluate this technique, a comparative study with other methods related in literature using the standard proposed in [6] is included.

The paper is organized from here in the following sections. It contains a Reference Methods section where the different prediction techniques, from a statistical point of view, utilized for the evaluation experiments are defined (specifically, there is a brief description of the persistence, autoregressive and autoregressive moving average models), and a Support Vector Regression section where the method is fully described. Results section shows the collected results of experiments achieved by using time series of wind speed, and finally, Conclusions section summarizes the main contributions and remarkable results.

2. Reference Methods

The experiments exposed in this paper are orientated to forecast speed wind in the southeast of Gran Canaria Island using methods related to regression process. These methods try to predict time series values in future k times ahead by a function that approximates its values thanks to historical training samples. They are discussed below. Previously we will present the models that will be used to evaluate comparatively the proposal based on SVR.

Persistence is the simplest model for forecasting and it is known as *Naive Predictor*. It is based on the assumption of high inertia in the subjacent physical model. If $y(t)$ is the value at time t of a time series, in persistence model the predicted value for k times ahead is

$$\hat{y}(t+k) = y(t) \quad (1)$$

It is considered a reference model hard to beat for short k values (reference).

A predictive autoregressive method (AR) is simply a linear regression that tries to reconstruct time series supposing that the present value $y(t)$ depends on recent past values $y(t-i)$ and a random component $\eta(t)$ with zero mean and constant variance. Formally,

$$y(t) = c + \sum_{i=1}^p \varphi_i y(t-i) + \eta(t) \quad (2)$$

where p is the order of the autoregressive model, φ_i the configuration parameters and c is a constant related with the process mean. The p order is identified by autocorrelation function (ACF) and partial autocorrelation function (PACF) using the Bayesian information criterion (BIC) and the Akaike information criterion (AIC) in a complementary form. An initial estimation of the φ_i parameters values is obtainable by

means of the Yule Walker equations which can be further improved by recursive iteration [7].

So, identified the order of model and parameters values, we can forecast the predicted value for a horizon time k by successive iterations from $\hat{y}(t+1)$ to $\hat{y}(t+k)$.

$$\hat{y}(t+k) = \sum_{i=1}^p \alpha_i \hat{y}(t+k-i) \quad (3)$$

By simplification, we have assumed a constant c equals to zero and the expectation of $\eta(t)$.

An improvement of an AR model was proposed by [8], it is called autoregressive moving average model (ARMA). The forecast in this model depends not only on the values it has had in the more or less recent past according to the autoregressive component, but it can also be a function of the residuals of past forecasts. The mathematical expression of the general ARMA(p, q) model that is in this case to the series is the following equation:

$$y(t) = c + \sum_{i=1}^p \varphi_i y(t-i) + \sum_{i=1}^q \theta_i \varepsilon(t-i) \quad (4)$$

where p is the order and φ_i the parameters of the autoregressive component and, q and θ_i the order and the parameters of the moving average component, respectively.

Briefly related, the construction of the model consists in identifying the p and q indices of the models, determining the φ_i and θ_i parameters contained in them, and finally validating them. The forecast is done the same way than AR model.

$$\hat{y}(t+k) = \sum_{i=1}^p \varphi_i \hat{y}(t+k-i) + \sum_{i=1}^q \theta_i \varepsilon(t+k-i) \quad (5)$$

3. Support Vector Regression

At this point, Regression using Support Vectors (SVR) is discussed. Support Vector (SV) [9] is firmly grounded in the framework of statistical learning theory or VC Theory (Vapnik-Chervonenkis Theory), developed over last fourth decades [10, 11, 12]. In a nutshell, VC theory characterizes properties of learning machines which enable them to generalize well to unseen data. In its present form, SV Machine was developed at AT&T Laboratories by Vapnik et al. and due to this industrial context, SV research has up to date a sound orientation towards real-world applications [13]. In regression and time series prediction applications, excellent performances were soon obtained [9, 14, 15, 16].

The goal in SVR is to find a function $f(x)$ that has at most ε deviation from the set of targets y_i ($i=1, \dots, l$) for all the training data, and at the same time, is as flat as possible. Suppose we have a training set $\{(x_1, y_1), \dots, (x_l, y_l)\} \subset X \times \mathbb{R}$ where X denotes the space of the input patterns of dimension \mathbb{R}^n and desire to approximate its values by a linear function $f: X \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$,

$$f(x) = \langle w \cdot x \rangle + b = x^T w + b = \sum_{i=1}^n w_i x_i + b \quad (6)$$

where $\langle \cdot \rangle$ is the dot product in X , w the weights vector and b the bias. One way to obtain *flatness*, i.e. small w , in (6) is minimizing the Euclidean norm $\|w\|^2$. If we want to allow for some errors we can introduce slack variables ξ_i, ξ_i^* to cope with infeasible constraints of the optimization problem. Therefore we arrive at the formulation (7) that corresponds to dealing with a ε -insensitive loss function, further described.

$$\begin{aligned} & \text{minimize } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \\ & \text{subject to } \begin{cases} y_i - (\langle w \cdot x_i \rangle + b) \leq \varepsilon + \xi_i \\ y_i - (\langle w \cdot x_i \rangle + b) \geq -\varepsilon - \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned} \quad (7)$$

The positive constant C determines the trade off between the flatness of f and the amount up to which deviations larger than ε are tolerated.

Finding a solution to the optimization problem is hard when training data is so large. The standard dualization method utilizing Lagrange multipliers solves it easier. The key idea is to construct a Lagrange function from both the objective function and the corresponding constraints, by introducing a dual set of variables denominated Lagrange multipliers. This technique yields the dual formulation of the optimization problem. Applying to the above case, the dual optimization problem is

$$\begin{aligned} & \text{maximize } \begin{cases} -\frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle x_i, x_j \rangle \\ -\varepsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) + \sum_{i=1}^l y_i (\alpha_i - \alpha_i^*) \end{cases} \\ & \text{subject to } \begin{cases} \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases} \end{aligned} \quad (8)$$

It is shown that

$$w = \sum_{i=1}^l (\alpha_i - \alpha_i^*) x_i \quad (9)$$

therefore

$$f(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \langle x_i, x \rangle + b \quad (10)$$

This is the so-called *Support Vector expansion*, i.e. w can be completely described as a linear combination of the training patterns x_i .

Computing b is done by exploiting the Karush-Kuhn-Tucker conditions. This state that at the optimal solution the product between dual variables and constraints has to vanish. So, b can be computed as follows:

$$\begin{aligned} b &= y_i - \langle w \cdot x_i \rangle - \varepsilon \quad \text{for } \alpha_i \in (0, C) \\ b &= y_i - \langle w \cdot x_i \rangle + \varepsilon \quad \text{for } \alpha_i^* \in (0, C) \end{aligned} \quad (11)$$

It describes that only for $|f(x_i) - y_i| \geq \varepsilon$ the Lagrange multipliers may be nonzero and therefore we do not need all x_i to describe w . The training samples that come with nonvanishing coefficients are denominated *Support Vectors (SV)*.

Normally, the function $f(x)$ that fits best to the training data is nonlinear, hence it is necessary a way to make the SV algorithm nonlinear. This could be achieved

by a mapping $\Phi: X \rightarrow \mathcal{F}$ that preprocesses the training patterns x_i into a greater dimension feature space \mathcal{F} and then applying the standard SV regression algorithm. Therefore, our goal would transform in finding a function $f(x)$ that satisfies

$$f(x) = \sum_{i=1}^n w_i \phi_i(x) + b \quad (12)$$

The main problem of this solution is that it becomes infeasible for both polynomial features of higher order and higher dimensionality. Properties of dual formulation may solve this inconvenience.

If we apply the transformation $\phi(x)$ in (10), we obtain

$$f(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \langle \phi(x_i), \phi(x) \rangle + b \quad (13)$$

As noted, the SV algorithm only depends on dot products between the various patterns. Hence it suffices to know and use $k(x_i, x) = \langle \phi(x_i), \phi(x) \rangle$ instead of $\phi(\cdot)$ explicitly. This allows to rewrite the Support Vector algorithm as follows:

$$\begin{aligned} & \text{maximize } \begin{cases} -\frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) k(x_i, x_j) \\ -\varepsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) + \sum_{i=1}^l y_i (\alpha_i - \alpha_i^*) \end{cases} \\ & \text{subject to } \begin{cases} \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases} \end{aligned} \quad (14)$$

The function $k(x_i, x)$ is denominated *kernel function* and must satisfies several conditions given by [17, 18, 19]. Examples of them are shown below.

Table 1. Examples of kernel function.

Kernel	Formulation
Linear	$K(x, z) = \langle Ax \cdot Az \rangle = x^T A^T A z = x^T B z$
Polynomial	$K(x, z) = (\langle x, z \rangle + c)^d$
Radial base function (RBF)	$K(x, z) = e^{-\frac{\ x-z\ ^2}{\sigma^2}}$

So far, we have solved the optimization problem when exist samples (x_i, y_i) with estimation errors $f(x_i) - y_i$ that follows a linear distribution. However, this distribution may vary depending the problem we meet. The concept of functional risk covers it.

Consider a training data $\{x_i, y_i\}_{i=1}^n$ with a probability distribution $P(x, y)$. Our goal will be to find a function f that minimizes a risk functional

$$R(\omega) = \int L(y_i, f(x_i)) dP(x, y) \quad (15)$$

where $L(y_i, f(x_i))$ denotes a cost function, or *loss function*, determining how we will penalize estimation errors. Given that the probability measure $dP(x, y)$ is unknown, we can only use X for estimating a function f that minimizes $R(\omega)$. An approximation consists in replacing the integration by the empirical estimate to get the *empirical risk functional* described as

$$R_{emp}(\omega) = \frac{\lambda}{2} \|\omega\|^2 + \frac{1}{n} \sum_{i=1}^n L(y_i, f(x_i)) \quad (16)$$

The term $\|\omega\|^2$ controls the capacity, i.e., the generalization error and $\lambda > 0$ is a regularization constant. This empirical function is well-approximated to the risk functional if we choose the terms correctly. Hence, the equation (16) will be the new goal function of the optimization problem. Some of the most common loss functions are described in the table 2.

Table 2. Examples of loss function.

Loss function	Formulation
Quadratic	$L(y, f(x)) = (f(x) - y)^2$
Laplacian	$L(y, f(x)) = f(x) - y $
Huber	$L(y, f(x)) = \begin{cases} \frac{1}{2}(f(x) - y)^2 & \text{if } f(x) - y < \mu \\ \mu f(x) - y - \frac{\mu^2}{2} & \text{otherwise} \end{cases}$
ϵ -insensitive	$L(y, f(x)) = \begin{cases} 0 & \text{if } f(x) - y < \epsilon \\ f(x) - y - \epsilon & \text{otherwise} \end{cases}$

To conclude this section, we expose a brief summary about steps to make a prediction using SV algorithm.

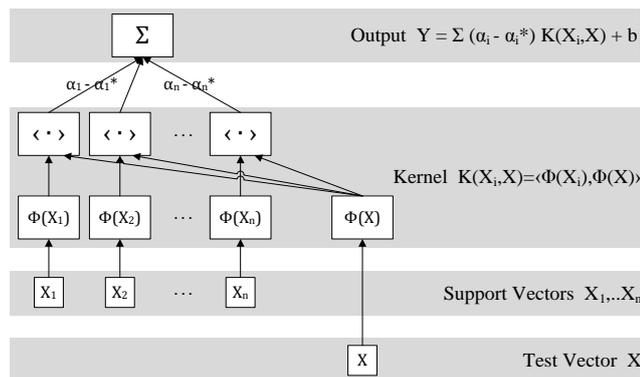


Figure 1. Prediction by SV algorithm.

The pattern for which a prediction is made X is mapped into a feature space by a map Φ . Then dot products are computed with the training patterns X_i under the map Φ . This corresponds to evaluating the kernel k function at location $k(x_i, x)$. Finally the dot products are added up using the weights $\alpha_i - \alpha_i^*$. This, plus the constant term b yields the final prediction output.

4. Results

We have used a wind data series acquired in Pozo Izquierdo, Gran Canaria Island (Spain). The wind speed series comprise about 20 weeks data from a meteorological tower in time steps of one minute and were obtained at 20 meters high. It corresponds to 203305 samples measured in meters per second. The data set was split in two subsets: train and test. The training data is a 60% (121983 samples) of the global data.

About time windows utilized in the experiments, we forecasted for a 10, 15, 30, 60, 90, 120, 180 and 240 minutes horizon time. To evaluate the forecasting performance it was used the standard protocol suggested

by [6]. It includes the definition of the Evaluation Criteria (EC): BIAS, MAE, RMSE and SDE, and also the improvement over a reference model which is computed in percent value as:

$$Imp_{ref, EC}(\%) = 100 \frac{EC_{ref} - EC}{EC_{ref}} \quad (17)$$

The reference model chosen for this improvement measure was the persistence one. Remark that the values of BIAS and MAE are related to the first moment of the error, therefore related to the wind speed, but the values of RMSE and SDE are related to the second order moment and the variance of the error.

The *System Identification Toolbox* of *MatLab*[®] software was used to estimate and evaluate the optimal parameters of the AR and ARMA models. It also includes several procedures to calculate ACF, PACF, discrete Fourier transform (DFT) and cumulative periodograms, needed to tune the different orders of models. In the case of SVR, we utilized the *libsvm* software [20] to obtain the support vectors and predict. To estimate the optimal configuration parameters in this model, a brief technical optimization study was performed using genetic algorithms provided by the *Optimization Toolbox*. The output of this process indicated that the least EC value for a 10 minutes time horizon was achieved with the configuration parameters given by table 3.

Table 3. Configuration parameters of SVR.

Parameter	Value
Dimension of input space	60
Loss function	ϵ -insensitive $\epsilon = 0.1$
Kernel function	RBF $\sigma = 30.151$
Regularization constant	$\lambda = 5.001$

The results of EC criteria for each proposed method and time horizon and the relative improvements respect the persistence model are shown in the following tables.

Table 4. Comparative results for 10 minutes ahead prediction.

	Persist.	AR	ARMA	SVR
BIAS	-0.00163	-0.07857	-0.05333	-0.02817
MAE	0.80383	0.75261	0.75079	0.71811
RMSE	1.10785	1.02271	1.02022	0.99028
SDE	1.10785	1.01969	1.01883	0.98989
Imp_MAE		6.372 %	6.598 %	10.664 %
Imp_RMSE		7.685 %	7.910 %	10.612 %
Imp_SDE		7.958 %	8.035 %	10.648 %

Table 5. Comparative results for 15 minutes ahead prediction.

	Persist.	AR	ARMA	SVR
BIAS	-0.00247	-0.10269	-0.07382	-0.03294
MAE	0.87065	0.82669	0.82428	0.78572
RMSE	1.19542	1.11893	1.11588	1.08101
SDE	1.19543	1.11422	1.11344	1.08051
Imp_MAE		5.049 %	5.326 %	9.755 %
Imp_RMSE		6.399 %	6.654 %	9.571 %
Imp_SDE		6.793 %	6.859 %	9.613 %

Table 6. Comparative results for 30 minutes ahead prediction.

	Persist.	AR	ARMA	SVR
BIAS	-0.00539	-0.14625	-0.10151	-0.06807
MAE	1.01350	0.98882	0.98352	0.93065
RMSE	1.39477	1.33291	1.32303	1.28518
SDE	1.39477	1.32487	1.31914	1.28339
Imp_MAE	2.435 %	2.958 %	8.175 %	
Imp_RMSE	4.435 %	5.144 %	7.857 %	
Imp_SDE	5.012 %	5.422 %	7.986 %	

Table 7. Comparative results for 60 minutes ahead prediction.

	Persist.	AR	ARMA	SVR
BIAS	-0.01125	-0.23753	-0.16622	-0.13874
MAE	1.20762	1.22497	1.21057	1.14396
RMSE	1.65428	1.62842	1.60624	1.56797
SDE	1.65425	1.61101	1.59763	1.56183
Imp_MAE	-1.437 %	-0.244 %	5.272 %	
Imp_RMSE	1.563 %	2.904 %	5.217 %	
Imp_SDE	2.614 %	3.423 %	5.587 %	

Table 8. Comparative results for 90 minutes ahead prediction.

	Persist.	AR	ARMA	SVR
BIAS	-0.01642	-0.28582	-0.22964	-0.20055
MAE	1.37970	1.40750	1.39021	1.33294
RMSE	1.88036	1.84867	1.82693	1.82098
SDE	1.88030	1.82646	1.81245	1.80991
Imp_MAE	-2.015 %	-0.762 %	3.389 %	
Imp_RMSE	1.685 %	2.841 %	3.158 %	
Imp_SDE	2.863 %	3.608 %	3.744 %	

Table 9. Comparative results for 120 minutes ahead prediction.

	Persist.	AR	ARMA	SVR
BIAS	-0.02120	-0.29865	-0.25748	-0.26970
MAE	1.53890	1.55273	1.53520	1.49954
RMSE	2.09242	2.02520	2.00925	2.04282
SDE	2.09232	2.00307	1.99270	2.02495
Imp_MAE	-0.899 %	0.240 %	2.558 %	
Imp_RMSE	3.213 %	3.975 %	2.370 %	
Imp_SDE	4.266 %	4.761 %	3.220 %	

Table 10. Comparative results for 180 minutes ahead prediction.

	Persist.	AR	ARMA	SVR
BIAS	-0.03046	-0.42464	-0.32271	-0.37906
MAE	1.78897	1.74479	1.74321	1.73844
RMSE	2.41498	2.25451	2.25064	2.34139
SDE	2.41481	2.21417	2.22740	2.31052
Imp_MAE	2.470 %	2.558 %	2.825 %	
Imp_RMSE	6.645 %	6.805 %	3.047 %	
Imp_SDE	8.309 %	7.761 %	4.319 %	

Table 11. Comparative results for 240 minutes ahead prediction.

	Persist.	AR	ARMA	SVR
BIAS	-0.04072	-0.51830	-0.50998	-0.44665
MAE	1.99938	1.90018	1.89221	1.94194
RMSE	2.65418	2.43315	2.42800	2.56516
SDE	2.65389	2.37733	2.37385	2.52599
Imp_MAE	4.962 %	5.360 %	2.873 %	
Imp_RMSE	8.328 %	8.522 %	3.354 %	

Imp_SDE	10.421 %	10.552 %	4.819 %
---------	----------	----------	---------

All tested methods perform worse on BIAS values such as increase the level in relation to the persistence model. Moreover, the greater the time horizon is; the greater the difference of this error between each method and the persistence one is.

However, in MAE, RMSE and SDE criteria the persistence value is beaten by all methods. For short time window, the SVR method presents better results but not for longer time. It is due, as we comment before, to the fact that the SVR parameters were fixed by genetic algorithm for 10 min. horizon and after used, for all horizons from 10 to 240 min. In this last case, ARMA method performs the smallest errors values.

Remark that differences between MAE values are greater for short time window whereas in the case of RMSE values are greater for the greatest time horizons. The reason is that RMSE value is related to variance of prediction error and the greater time horizon is, the greater this error is.

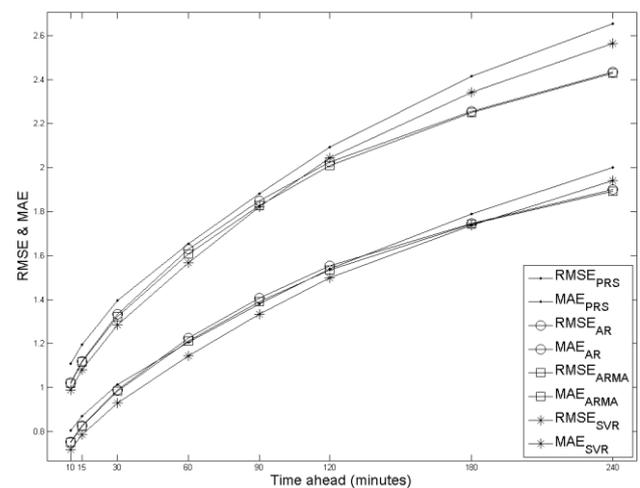


Figure 2. Comparative of MAEs and RMSEs between methods.

Note that the improvement of the autorregressive methods (AR and ARMA) regarding to persistence model is not always positive. For a time window between 60 and 90 minutes, the MAE value of persistence is smaller; however, its RMSE value is greater. This is due to the properties of autorregressive models that consider a constant variance of forecasting error.

Although in the previous experiments ARMA model performed better results than SVR in time horizon greater than 120 minutes, this statement is not completely true because SVR was fitted for a very short time forecast. The table 12 shows the results for a SVR method fitted to a 240 minutes time horizon prediction. Hence, also in this case, SVR performs better EC than ARMA model.

Table 12. SVR improvement for 240 minutes ahead prediction.

	Persist.	ARMA	SVR
BIAS	-0.04072	-0.50998	-0.57606
MAE	1.99938	1.89221	1.88426
RMSE	2.65418	2.42800	2.40935

SDE	2.65389	2.37385	2.33948
Imp_MAE	5.360 %	5.758 %	
Imp_RMSE	8.522 %	9.224 %	
Imp_SDE	10.552 %	11.847 %	

5. Conclusions

The use of SVR method has been proposed in this paper to predict wind speed time series, and then, have been experimentally compared with the obtained data by the reference models described above.

This study has used a standard protocol to evaluate the performance of forecasting procedures that some authors have introduced. We have compared the results according this protocol. We have shown that SVR method perform better evaluation criteria than other reference models.

Moreover, we have shown that SVR exhibit a good performance, beating Persistence, AR, and ARMA Predictors for different time horizons. It is also a flexible method able to be trained for obtaining predictions adapted to different time horizons ahead. This property would be useful in building a forecasting system fitted to final user needs, i.e. able to predict for different time horizon as shown in figure 3, tuning by learning the parameters of different SVR predictors for different time horizons.

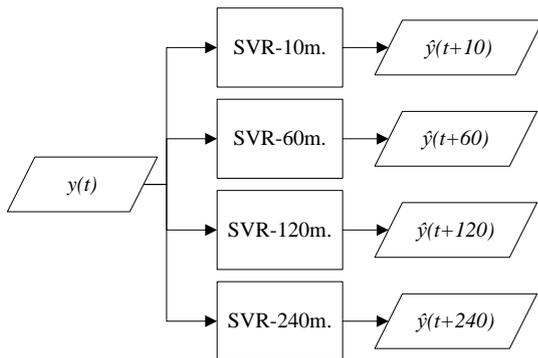


Figure 3. Forecasting system by different fitted SVR methods.

Acknowledgement

The authors wish to acknowledge to the Instituto Tecnológico de Canarias (ITC) for providing all the wind speed series used in this paper, acquired in its facilities in Pozo Izquierdo, Gran Canaria Island.

This work has been partially supported by a University of Las Palmas de Gran Canaria predoctoral fellowship to Ibrahim Espino, and also by the Comunidad Autónoma de Canarias research project (SolSubC200801000137).

References

[1] S. Soman, H. Zareipour, O. Malik, and P. Mandal, "A review of wind power and wind speed forecasting methods with different time horizons", in Proc. 42nd North American Power Symposium (NAPS) in Arlington, Texas, USA, 2010.

[2] I. Kuikka, "Wind nowcasting: optimizing runway in use", Technical report, Systems Analysis Laboratory, Helsinki University of Technology, 2009.

[3] M. Milligan and M. Schwartz, "Statistical wind power forecasting models: results for U.S. wind farms", in Proc. Windpower, Austin, Texas, USA, 2003.

[4] C. Potter and M. Negnevitsky, "Very short-term wind forecasting for Tasmanian power generation", in Proc. IEEE Trans. Power Syst. 2006, Vol. 21, No. 2, pp. 965-972.

[5] M. Lange and U. Focken, "New developments in wind energy forecasting", in Proc. IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century 2008, pp. 1-8.

[6] H. Madsen, "EC project ANEMOS: A protocol for standardizing the performance evaluation of short-term wind power prediction models", Technical report, Technical University of Denmark, 2004.

[7] G. Yule, "On a method of investigating periodicities in disturbed series", in Proc. Philosophical Transactions of the Royal Society of London 1927, Vol. 226, pp. 267-298.

[8] G. Box, G. Jenkins and G. Reinsel, Time series analysis: forecasting and control, Prentice Hall 3ed, 1994.

[9] H. Drucker, C. Burges, L. Kaufman, A. Smola and V. Vapnik, "Support vector regression machines", in Proc. Advances in Neural Information Processing Systems 1997, Vol. 9, pp. 155-161.

[10] V. Vapnik and A. Chervonenkis, "A note on one class of perceptrons", in Proc. Automation and Remote Control 1964, Vol. 25.

[11] V. Vapnik, Estimation of dependences based on empirical data, Springer, Berlin (1982).

[12] V. Vapnik, The nature of statistical learning theory, Springer, New York (1995).

[13] A. Smola and B. Schölkopf, "A tutorial on support vector regression", in Proc. Statistics and Computing 2004, Vol. 14, No. 3, pp. 199-222.

[14] K. Müller, A. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen and V. Vapnik, "Predicting time series with support vector machines", in Proc. Artificial Neural Networks ICANN 1997, Vol. 1327, pp. 999-1004.

[15] M. Stitson, A. Gammerman, V. Vapnik, V. Vovk, C. Watkins and J. Weston, "Support vector regression with ANOVA decomposition kernels", in Proc. Advances in Kernel Methods-Support Vector Learning 1999, Vol. 1, pp. 285-292.

[16] D. Matterna and S. Haykin, "Support vector machines for dynamic reconstruction of a chaotic system", in Proc. Advances in Kernel Methods-Support Vector Learning 1999, Vol. 1, pp. 211-242.

[17] J. Mercer, "Functions of positive and negative type and their connection with the theory of integral equations", in Proc. Philosophical Transactions of the Royal Society of London 1909, Vol. 209, pp. 415-446.

[18] C. Burges, "A tutorial on support vector machines for pattern recognition", in Proc. Knowledge Discovery and Data Mining 1998, Vol. 2, pp. 121-167.

[19] A. Smola, B. Schölkopf and K. Müller, "The connection between regularization operators and support vector kernels", in Proc. Neural Networks 1998, Vol. 11, pp. 637-649.

[20] LibSVM. <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>