10

Machine Learning Techniques for Environmental Data Estimation

Vassilios Petridis[1] and Vassilis Syrris[2]

[1] Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Greece petridis@eng.auth.gr
[2] Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Greece vsyrris@auth.gr

Summary. In this paper we investigate the issue of wind speed prediction at a particular location in the urban area of Thessaloniki, Greece, based on the historical data containing wind parameter values at two other different locations. We evaluate the performance of two significant machine learning methodologies, the Fuzzy Lattice Neurocomputing (FLN) and the Support Vector Regression (SVR). The results of the specific applications are compared with past work on the same data set, and a discussion upon the exhibited features is carried out.

10.1 Introduction

Wind prediction (short/long-term) is of great significance for the wind energy production which is the fastest growing type of renewable energy in Europe. The energy produced strongly depends on the actual wind speed at a certain location, so the power output cannot be guaranteed at all times. The objective for the energy market is the secure and economical management of a power system [13], thus there is an urgent need for the development of flexible computational techniques capable of controlling and governing efficiently the energy resources.

Several researchers studied wind parameters at a certain location [9, 17]. However the performance of such an approach was not particularly remarkable, at short-time prediction mainly, especially when compared with simple techniques (e.g. Persistence). Wind forecasting is a stochastic process with a high level of non-stationarity. This fact prompted researchers to take into account *spatial correlation* of wind parameters at different locations [3, 19, 21].

This article describes two computational representations the first originating from the Fuzzy Lattice Neurocomputing (FLN) framework introduced in [11, 12, 20] and the second from Support Vector Machines and the Kernel-based framework. FLN is a scheme which combines elements from the Adaptive Resonance Theory (ART) [6], the mathematical theory of lattices [4] and

V. Petridis and V. Syrris: *Machine Learning Techniques for Environmental Data Estimation*, Studies in Computational Intelligence (SCI) **67**, 195–214 (2007)
www.springerlink.com

the theory of fuzzy sets [25]. The FL-framework can handle families of intervals. Also it can cope with disparate types of data including vectors, Boolean data, symbols, images, text, graphs, etc. On the other hand, we have a promising regression technique, the Support Vector Regression (SVR) which has been introduced by V. Vapnik and his collaborators [16, 23, 24]. It is deemed as a universal learning machine founded on the principle of Structural Risk Minimization which derives from statistical theory.

The goal herein is to test and validate the prediction hypothesis resolved by both a FLN clustering/regression and a SVR approach. We examine the case of real-time learning in a significantly non-stationary problem and we demonstrate how the spatial correlation improves the final learning performance. The application domain is the forecasting of wind speed at a meteorological station based on values also measured by two other stations presenting spatial correlation with the one in discussion.

10.2 Case Description

The study area is around the Thermaikos gulf, in the city of Thessaloniki, Greece (Fig. 10.1); it is a region rather flat and almost at sea level. The type of wind we are interested in is medium to high-speed. The prevailing winds blow from the north-northwest. Therefore, two meteorological stations have been situated at locations S_1 and S_2 that together with our location of interest, the local station S_0, are along the axis N-NW where strong winds appear. The wind speed distribution is similar in all three stations. Their distances are: $(S_1S_2) = 27$ km, $(S_2S_0) = 12$ km and obviously $(S_1S_0) = 39$ km.

One minute measurements of wind speed and direction were collected at the aforementioned three locations for a year [1] and processed in order to remove non valid values and errors. The remaining set was averaged for every 15 minutes eliminating thus the measurement noise and the sudden wind variations. The resulting measurements set consists of 6 measurements at each instant $(\upsilon_{S_1}(t), \varphi_{S_1}(t), \upsilon_{S_2}(t), \varphi_{S_2}(t), \upsilon_{S_0}(t), \varphi_{S_0}(t))$ where υ and φ denote speed and direction respectively. There are 3260 time instants.

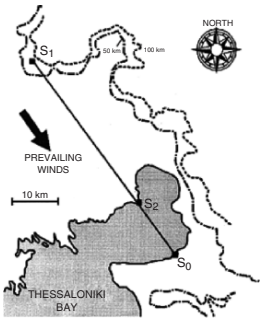

Fig. 10.1. The map shows the location of the three meteorological stations

We point out that the location of the three stations allows the exploitation of any *spatial correlations* among the locations involved. To this end we use neighboring areas measurements concerning wind events in order to achieve

higher precision in the predictions regarding the study area. Our aim is to predict future values of site S_0 based on past values of all three stations.

In time-series prediction, the prediction origin, denoted t_0, is the time from which the prediction is generated. The time between the prediction origin and the predicted data point is the prediction horizon h or the time lead of the time series, while the stack of data used to conduct the prediction is defined as batch history data, b_{hd}, or look back length.

10.3 The FLN Forecasting Method

The reasoning behind the wind prediction problem as we formulate it is to exploit possible similarities and correlations among the three stations by means of FL-clustering application onto their historical data concerning wind speed and wind direction. Thus, the subject matter is simplified and cast as a problem of clustering; that is detecting sets of values the average of which, with the suitable adjustment, can be employed as predictor. In case of new evidence, the algorithm classifies it to a cluster. In fact this method is a regression technique involving two phases: a clustering process in the first phase and a classification process in the second phase of the algorithm. The method is applicable in a fuzzy lattice data domain and in this paper, in space R^4.

10.3.1 The FL-framework

A *lattice* L is a partially ordered set of which any two elements have a greatest lower bound (*meet*) denoted by $x \wedge y$ and a least upper bound (*join*) denoted by $x \vee y$. A lattice L is called *complete* when each of its subsets has a least upper bound and a greatest lower bound in L. A non-void complete lattice has a least element (O) and a greatest element (I).

At this point we mention a considerable asset of lattice theory in knowledge representation. A lattice L can be the Cartesian product $L = L_1 \times \cdots \times L_N$ of N constituent lattices $L_1, \cdots, L_N$. A product lattice L involving disparate constituent lattices has the potential of dealing with disparate types of data such as vectors of real numbers, propositions, fuzzy sets, events in a probability space, symbols, graphs, etc.
A useful function in a lattice L is a valuation function $v : L \to R$ defined by

$$v(x) + v(y) = v(x \wedge y) + v(x \vee y), x, y \in L.$$

A valuation is called positive if and only if $x < y \Rightarrow v(x) < x(y)$. An *inclusion measure* function σ is defined on a complete lattice L as a map $\sigma : L \times L \to [0, 1]$ such that for $u, w, x \in L$ the following three axioms are satisfied:

(Axm1) $\sigma(x, O) = 0, x \neq O$
(Axm2) $\sigma(x, x) = 1, \forall x \in L$
(Axm3) $u \leq w \Rightarrow \sigma(x, u) \leq \sigma(x, w)$ *Consistency Property*

Given a positive valuation function $v(.)$ in a lattice L an inclusion measure can be defined by the ratio $\sigma(x,u) = \frac{v(u)}{v(x \vee u)}$ [11]. The complete lattice $\tau(L)$ of intervals of lattice elements has been analyzed in [11]. In that work it was shown that an inclusion measure in a lattice L implies a fuzzy lattice, which can be defined as follows: *A fuzzy lattice is a pair* $< L, \mu >$*, where* L *is a crisp lattice and* $(L \times L, \mu)$ *is a fuzzy set with membership function* $\mu : L \times L \rightarrow [0,1]$ *such that* $\mu(x,y) = 1$ *if and only if* $x \leq y$*. It turns out that* $< L, \sigma >$ *is a fuzzy lattice.*

Finally the interval size is defined given a positive valuation function $v(.)$ in a lattice L. The size of an interval $x = [a,b] \in \tau(L)$ is a function $Z : L \rightarrow R$ as expressed in $Z([a,b]) = v(b) - v(a)$.

10.3.2 Application of the method

In FL-data formulation we represent both parameters (speed and direction) of each station as a 2-dimensional vector: $x_{S_0}(t) = [\upsilon_{S_0}(t), \varphi_{S_0}(t)], x_{S_1}(t) = [\upsilon_{S_1}(t), \varphi_{S_1}(t)], x_{S_2}(t) = [\upsilon_{S_2}(t), \varphi_{S_2}(t)]$. The data set D_{FL} consists of $x_{S_0}(t)$, $x_{S_1}(t), x_{S_2}(t)$ for 3260 time instants. So the reference domain is considered to be the complete product lattice $L = R^2 \times R^2$ (called a rectangle), where a constituent complete lattice is the interval of real numbers. The prediction scheme takes the form:

$$\begin{aligned} \tilde{x}_{S_0}(t_0 + h) = f(x_{S_0}(t_0), ..., x_{S_0}(t_0 - (b_{hd} - 1)), x_{S_1}(t_0), ..., \\ x_{S_1}(t_0 - (b_{hd} - 1)), x_{S_2}(t_0), ..., x_{S_2}(t_0 - (b_{hd} - 1))) \end{aligned} \tag{10.1}$$

During the experiments we set up the variables h and b_{hd} to be equal.

A positive valuation function in R is given by $v(x) = x$. The *inclusion measure* $\sigma : R \times R \rightarrow [0,1]$ is defined as:

$$\sigma(x,y) = \frac{v(y)}{v(x \vee y)} = \frac{y}{\mathtt{max}(x,y)}, x, y \in R.$$

In order to determine the inclusion measure of a point $u = (x,y)$ into the rectangle $w = (x_1, y_1) \times (x_2, y_2)$, we represent the former by its respective degenerate rectangle $u = (x,y) \times (x,y)$ and then we introduce the following formula that calculates the inclusion measure of u in w:

$$\sigma(u,w) = \frac{1}{2}\left(\frac{mean(x_1,x_2)}{\mathtt{max}(x, mean(x_1,x_2))} + \frac{mean(y_1,y_2)}{\mathtt{max}(y, mean(y_1,y_2))}\right) \tag{10.2}$$

$$\text{where } mean(a,b) = \tfrac{a+b}{2}, a, b \in R.$$

It is worth mentioning that all the points of the constituent lattices are comparable as we can see in the following example: consider the points $u = (0,4)$ and $w = (3,2)$. Their degenerate rectangles are: $u = (0,4) \times (0,4)$ and $w = (3,2) \times (3,2)$. According to the relationship (10.2) their respective inclusion measures are: $\sigma(u,w) = 1.5/2 = 0.75$ and $\sigma(w,u) = 1/2 = 0.5$ and consequently, we infer that $u \leq w$.

10.3.3 Algorithm description

The prediction scheme of (10.1) is implemented by means of the algorithm described in this subsection. Before we proceed to the essence of the FL learning mechanism, we should mention two crucial points for the fine tuning of the model. A major parameter is the *vigilance parameter* $\rho \in R$ which is necessary for the clustering procedure in order to adjust the size of the clusters. The vigilance parameter specifies the algorithm sensitivity. As ρ increases, the calculated rectangles size increases too. Another basic factor is the *correction* of the function outcome (10.1). After many experiments we obtain the best results when we adjust the vigilance parameter as $\rho = s$ (standard deviation of the sample/time window), while the prediction improves if we take its average with the last measured value, i.e. if $x_{S_0}(t)$ is the last known value at station S_0 measured at time t and $\tilde{x}_{S_0}(t+h)$ is the estimation at time $t+h$ given by the (10.1), then the final prediction is given by:

$$p_{S_0}(t+h) = \frac{x_{S_0}(t) + \tilde{x}_{S_0}(t+h)}{2} \tag{10.3}$$

Moreover, there is no need for any normalization of the experimental values.

The algorithm consists of two phases, the clustering procedure and the prediction function:

The clustering phase as an iterative procedure:

C1. A data set of n vectors is given.

C2. At the initial pass of the algorithm, the first vector is presented and its degenerate rectangle constitutes the first cluster. As the algorithm proceeds a set of clusters CR is created $CR = \{CR_1, ..., CR_L\}$ as described in the sequel. A cluster is represented by a rectangle in $R^2 \times R^2$.

C3. At each iteration a new vector $u = [x, y]$ is fed to the algorithm.

C4. The inclusion measure of $u = [x, y]$ is calculated with respect to each rectangle $CR_1, ..., CR_L$.

C5. Rectangles $CR_1, ..., CR_L$ compete over input $u = [x, y]$. Winner is the rectangle CR_W which includes u the most, i.e. it has the largest inclusion measure.

C6. Winner rectangle CR_W is augmented tentatively so as to include the degenerate rectangle $u = [x, y] \times [x, y]$. For instance, if $CR_W = [x_1, y_1] \times [x_2, y_2]$ then the union rectangle CR_U is calculated as:

$$CR_U = [\mathtt{min}(x, x_1), \mathtt{min}(y, y_1)] \times [\mathtt{max}(x, x_2), \mathtt{max}(y, y_2)]$$

C7. If size of CR_U is smaller than or equals the vigilance parameter then the rectangle CR_W is replaced by CR_U. Otherwise, reset occurs (i.e. the size of CR_W is reverted to its previous state and it is excluded from the list of the candidate rectangles to be compared with u), and the next winner is selected among the remaining rectangles.

C8. If all the rectangles have been reset, then input $u = [x, y]$ is learned as a new rectangle, $CR_{new} = [x, y] \times [x, y]$.

The prediction phase as a categorization procedure:

1. We set both the time horizon to be h steps from the last known measurement at S_0 (at the moment $t = 0$) and the look back length to be b_{hd}. This means that we want to estimate $x_{S_0}(t_0 + h)$ based on b_{hd} vectors.
2. The three vectors $x_{S_0}(t = 0) = [\upsilon_{S_0}(t = 0), \varphi_{S_0}(t = 0)], x_{S_1}(t = 0) = [\upsilon_{S_1}(t = 0), \varphi_{S_1}(t = 0)]$ and $x_{S_2}(t = 0) = [\upsilon_{S_2}(t = 0), \varphi_{S_2}(t = 0)]$ of the data set D_{FL} are selected.
3. The standard variation s of the above vectors is calculated.
4. For the clustering procedure [C1–C8] we use as data set the vectors $x_{S_0}(t = 0), x_{S_1}(t = 0), x_{S_2}(t = 0)$. The result is the generation of the set of cluster/s referred to as CR.
5. An input vector W is presented to the algorithm. When $t = 0$ then $W = x_{S_0}(t = 0)$. The inclusion measure of W is calculated for each rectangle of step 4 resulting thus in the inclusion measure vector.
6. The values of the inclusion measure vector are normalized to give the sum of 1 producing the column vector $[NIM(CR_i)]$, where $NIM(CR_i)$ is the normalized inclusion measure of rectangle CR_i, $i = 1, 2, ..., L$.
7. The mean of a rectangle $CR_i = [x_{1i}, y_{1i}] \times [x_{2i}, y_{2i}]$ is a point $M_i = \left(\frac{x_{1i}+x_{2i}}{2}, \frac{y_{1i}+y_{2i}}{2}\right) i = 1, 2, ..., L$. The estimation of $x_{S_0}(t_0+h)$ is calculated by multiplying the mean of each rectangle with its respective normalized inclusion measure and then by summing each partial result, i.e. $\tilde{x}_{S_0}(t_0 + h) = [M_i] \times [NIM(CR_i)], i = 1, 2, ..., L$.
8. The estimate is corrected by averaging the produced result from step 7 with the last measurement (at the station S_0) as shown in (10.3). The outcome $p_{S_0}(t = 1)$ is the first estimation for one step ahead $h = 1$.
9. We check if we have reached the desired time horizon. If so, then the algorithm stops otherwise it continues to the next step.
10. A new data set is created using the vectors $x_{S_0}(t = 0), x_{S_1}(t = 0), x_{S_2}(t = 0)$, the estimate $p_{S_0}(t = 1)$ and the vectors for one step back (except the case where we have reached the look back length b_{hd}) $x_{S_0}(t = -1), x_{S_1}(t = -1), x_{S_2}(t = -1)$, i.e. $NewData = \{x_{S_0}(t_{-1}), x_{S_1}(t_{-1}), x_{S_2}(t_{-1}), x_{S_0}(t_0), x_{S_1}(t_0), x_{S_2}(t_0), \widetilde{x}_{S_0}(t_1)\}$.
11. The vigilance parameter is taken equal to the standard variation of the $NewData$ data set.
12. We apply the clustering procedure [C1-C8] to the data set defined in step 10. New rectangles are formed representing the knowledge generated by the contribution of new evidence.
13. The flow of the algorithm goes to stage 5 by using as new input the estimated value $p_{S_0}(t = 1)$ (in general, by using the last estimated value), i.e. $W = p_{S_0}(t = 1)$, and CR is the set of clusters generated in step 12.

10.4 The SVR Method

The theory of Support Vector Machines has been extended in order to cope with regression problems. Suppose that we have a data set $\{(x_i, y_i), i = 1, 2, ..., n\}$ of measurements where $x_i \in X$ and $y_i \in R$ (X denotes the space of the input patterns and R is the set of real numbers). We consider that the tuples (x_i, y_i) are taken from an unknown distribution $P(x, y)$.

10.4.1 Linear regression

The formulation of the linear regression task stated by Vapnik [23] is the following *convex optimization problem*:

$$\begin{aligned} &\text{minimize } \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \\ &\text{subject to } \begin{cases} y_i - \langle w, x_i\rangle - b \le \varepsilon + \xi_i \\ \langle w, x_i\rangle + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases} \end{aligned} \tag{10.4}$$

where $\langle \cdot, \cdot \rangle$ refers to the inner product in X. $C > 0$ is a pre-specified value that determines the trade-off between the flatness of regressor and the amount up to which deviations larger than ε are tolerated and ξ_i, ξ_i^* are slack variables interpreted as: ξ for exceeding the target value by more than ε and ξ^* for being more than ε below the target value (soft margin regression).

In addition to (10.4) a loss function is minimized. For our SV-models we choose the linear ε-insensitive loss function :

$$L_\varepsilon = \begin{cases} 0 \text{ for } |y_i - (\langle w, x_i\rangle - b)| \le \varepsilon \\ |y_i - (\langle w, x_i\rangle - b)| - \varepsilon \text{ otherwise} \end{cases}$$

This defines an ε tube in such a way that if the predicted value is within the tube then the loss is zero, whereas if the predicted point is outside the tube, the loss is the magnitude of the difference between the predicted value and the radius ε of the tube.

To assure that the training data appear as inner products among the vectors and to better handle the constraints, the problem is transformed into a Lagrangian formulation named the primal form:

$$\begin{aligned} L_P = {} & \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) - \sum_{i=1}^{n}(\eta_i\xi_i + \eta_i^*\xi_i^*) \\ & - \sum_{i=1}^{n} a_i\left(\varepsilon + \xi_i - (y_i - \langle w, x_i\rangle - b)\right) \\ & - \sum_{i=1}^{n} a_i^*\left(\varepsilon + \xi_i^* - (\langle w, x_i\rangle + b - y_i)\right) \end{aligned} \tag{10.5}$$

where $a_i, a_i^*, \eta_i, \eta_i^* \ge 0$ are the Lagrange multipliers.

Differentiating with respect to w, b, ξ_i, ξ_i^* gives:

$$\begin{aligned}
\frac{\partial L}{\partial w} &= w - \sum_{i=1}^{n} (a_i - a_i^*)\, x_i = 0 \\
\frac{\partial L}{\partial b} &= \sum_{i=1}^{n} (a_i^* - a_i) = 0 \\
\frac{\partial L}{\partial \xi_i} &= C - a_i - \eta_i = 0 \\
\frac{\partial L}{\partial \xi_i^*} &= C - a_i^* - \eta_i^* = 0
\end{aligned} \tag{10.6}$$

Substituting for w into (10.5) and using the relations (10.6) the dual optimization problem is obtained:

$$\begin{aligned}
L_D = &-\tfrac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (a_i - a_i^*) \left(a_j - a_j^*\right) \langle x_i, x_j \rangle \\
&- \varepsilon \sum_{i=1}^{n} (a_i + a_i^*) + \sum_{i=1}^{n} (a_i - a_i^*)\, y_i
\end{aligned} \tag{10.7}$$

This function is maximized subject to:

$$\sum_{i=1}^{n} (a_i - a_i^*) = 0 \text{ and } 0 \le a_i, a_i^* \le C \tag{10.8}$$

Solving the first equation of (10.6) for w gives: $w = \sum_{i=1}^{n} (a_i - a_i^*)\, x_i$ so the candidate linear function takes the form:

$$f(x) = wx + b = \sum_{i=1}^{n} (a_i - a_i^*) \langle x_i, x \rangle + b \tag{10.9}$$

The Karush-Kuhn-Tucker complementary conditions [14, 15] are:

$$\begin{aligned}
&a_i \left(\varepsilon + \xi_i - (y_i - \langle w, x_i \rangle - b)\right) = 0 \\
&a_i^* \left(\varepsilon + \xi_i^* - (\langle w, x_i \rangle + b - y_i)\right) = 0 \\
&\eta_i \xi_i = (C - a_i)\, \xi_i = 0 \\
&\eta_i^* \xi_i^* = (C - a_i^*)\, \xi_i^* = 0
\end{aligned} \tag{10.10}$$

The (10.10) imply that $a_i a_i^* = 0$ which means that the set of *dual variables* can never be nonzero at the same time. Those patterns x_i with $a_i > 0$ or $a_i^* > 0$ are support vectors. If $a_i \in (0, C)$ or $a_i^* \in (0, C)$ then (x_i, y_i) lies on the boundary of the tube surrounding the regression function at distance ε. Moreover, if $a_i = C$ or $a_i^* = C$ then the point lies outside the tube. Thus, the parameter b is computed as follows:

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

10.4.2 Nonlinear regression

In this case the nonlinear function has the form: $f(x) = \langle w, \phi(x) \rangle + b$ where $\phi(x)$ is the image of input vector x in a high dimensional space. Using the trick of kernel functions [7] the dual form (10.5) is replaced by the quadratic problem:

$$\begin{aligned} L_D = & -\tfrac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(a_i - a_i^*\right) \left(a_j - a_j^*\right) K\left(x_i, x_j\right) \\ & -\varepsilon \sum_{i=1}^{n} \left(a_i + a_i^*\right) + \sum_{i=1}^{n} \left(a_i - a_i^*\right) y_i \end{aligned} \tag{10.11}$$

where $K\left(x_i, x_j\right) = \langle \phi(x_i), \phi(x_j) \rangle$ is a *kernel* function satisfying Mercer's conditions [8]. The relationship (10.11) is maximized subject to the constraints (10.8). Finally, the regressor takes the form:

$$f(x) = \sum_{i=1}^{n} (a_i - a_i^*) K(x_i, x) + b \tag{10.12}$$

In that way, we manage to apply linear regression not in the low dimensional input space but in a high dimensional (feature) space via the kernel function which makes the mapping implicitly, i.e. without knowing $\phi(x)$.

10.4.3 The prediction of SV-models

As analyzed in previous section, the objective of SVR is to construct a hyperplane that lies "close" to as many of the data points as possible [22]. The solution is obtained as a set of support vectors that can be sparse. These lie on the boundary and as such summarize the information required to separate the data.

SVR has an advantage over other function estimation methods: it is capable of controlling the capacity of the hypothesis; the algorithm selects the subspace of the hypothesis space that is optimal in terms of some bound on the generalization of the hypothesis. This fact leads to the choice of the kernel function, the width of the ε-insensitive zone and the capacity control C for controlling the regression model. Since there is not a rigorous way to select the finest set of hyperparameters, i.e. kernel, ε and C, we use the leave-one-out cross-validation method by constructing successive time windows of size $b_{hd} + 1$, i.e. we utilize b_{hd} vectors in order to predict one value h steps ahead.

After a sufficient number of experiments, we obtained the best results when we used the RBF kernel $K\left(x_i, x_j\right) = \langle \phi(x_i), \phi(x_j) \rangle = e^{-\frac{\left(x_i - x_j\right)^2}{2\sigma^2}}$, where the width of radial basis functions equals to $\sigma = 0.08$ and the other parameters were tuned to the values: $\varepsilon = 0.05$, $C = 1$.

We developed two different models :

$M1$. Solve the quadratic optimization problem (10.11), calculate the Lagrange multipliers and the bias and finally use the regressor (10.12) to get the prediction.
$M2$. Solve the quadratic optimization problem (10.11) and express the final prediction as a linear dependence of the arisen support vectors.

Each of the above models has two alternative versions concerning the input data type and the output:

a. We define the sets $V_1 = \{x_i(t) = [\upsilon_{S_0}(t)\ \upsilon_{S_1}(t)\ \upsilon_{S_2}(t)]$ for $t = 1, 2, ..., 3260\}$ (it consists of 3260 3-dimensional vectors) and $V_2 = \{\upsilon_{S_0}(t), \upsilon_{S_1}(t), \upsilon_{S_2}(t)$ for $t = 1, 2, ..., 3260\}$ (it consists of 3×3260 scalars). Thus, the inputs of the models are either i) 3-dimensional vectors that is $x_i \in V_1$ or ii) 1-dimensional vectors that is $x_i \in V_2$.
b. The outcome of the models $\tilde{x}_{S_0}(t_0 + h)$ can either i) remain intact or ii) be averaged with the last measured value $x_{S_0}(t_0)$ at S_0, i.e.

$$pred_{S_0}(t_0 + h) = \frac{x_{S_0}(t_0) + \tilde{x}_{S_0}(t_0 + h)}{2} \tag{10.13}$$

For simplicity reasons, when we refer, for instance, to model $M1.a_{(i)}.b_{(ii)}$ we mean the option which uses the regressor (10.12), it takes 3-dimensional vectors as input and its output results from (10.13) (i.e. model $M1$, versions $a_{(i)}$ and $b_{(ii)}$). The same convention holds for all the other options.

10.4.4 Algorithms description

A sliding window is used containing $x(t), x(t-1), ..., x(t-b_{hd}+1)$. For example, we consider as look back length 3 steps and as future time horizon 10 steps. If we start at the moment $t_0 = 6$, we use three vectors at time instances $t_{-2} = 4$, $t_{-1} = 5$ and $t_0 = 6$ as historical data which lead to the prediction at $t_{10} = 16$. At next moment $t_0 = 7$ we use the vectors at time instances $t_{-2} = 5$, $t_{-1} = 6$ and $t_0 = 7$ to make a prediction at time $t_{10} = 17$ etc. The algorithms of the previous section are described more analytically below:

Algorithm $M1.a_{(i)}.b_{(i)}$

S1) Set future time horizon h, the look back length b_{hd} and the prediction origin t_0.
S2) Normalize data in [−1,1].
S3) For $i = t_0$ to 3260:
  S3.1) The vectors x_i in regressor (10.12) are the vectors in the interval $(t_0 - b_{hd} + 1, t_0)$: $[x_{S_1}(t_0 - k), x_{S_2}(t_0 - k), x_{S_0}(t_0 - k)]$ and $y_i = x_{S_0}(t_0 - k + h)$, where $k = m - 1$ and $m = (b_{hd} - 1), ..., 2, 1$.
  S3.2) Solve the quadratic problem.

S3.3) Use the regressor function (10.12) where
$x = [x_{S_1}(t_0+h), x_{S_2}(t_0+h), x_{S_0}(t_0+h)]$
S3.4) The regressor's output $\tilde{x}_{S_0}(t_0+2h)$ is the prediction for h steps ahead.

Algorithm $M2.a_{(i)}.b_{(i)}$

Until $S3.2$ the steps are the same with the previous algorithm.

— S3.3) We take into account the $x_{S_0}(t_0+h)$ component of the input vector x. Then we consider the $x_{S_0}(t+h)$ component of the support vector SV_i, $i = 1, ..., M$ (where M is the number of support vectors) denoted by $x_{S_0}^{SV_i}$. Finally, the column vector $D_{ist} = [|x_{S_0}^{SV_1} - x_{S_0}(t_0+h)|, ..., |x_{S_0}^{SV_M} - x_{S_0}(t_0+h)|]^T$ is computed. The D_{ist} is normalized as $ND_{ist} = D_{ist} / \sum_{i=1}^{M} |x_{S_0}^{SV_i} - x_{S_0}(t_0+h)|$.

— S3.4) The final prediction is: $x_{S_0}(t_0+2h) = [x_{S_0}^{SV_1}, ..., x_{S_0}^{SV_M}] \times ND_{ist}$.

In the cases $1.a_{(ii)}$ and $2.a_{(ii)}$ we modify step $S3.1$ as:

Select all vectors in the interval $(t_0 - b_{hd} + 1, t_0)$: $[x_{S_1}(t_0-k), x_{S_0}(t_0-k+h)], [x_{S_2}(t_0-k), x_{S_0}(t_0-k+h)], [x_{S_0}(t_0-k), x_{S_0}(t_0-k+h)]$, where $k = m-1$ and $m = (b_{hd}-1), ..., 2, 1$.
In the cases $1.b_{(ii)}$ and $2.b_{(ii)}$ we change the outcome of step $S3.4$ using (10.13).

10.5 Model Evaluation

In order to test the performance of the models we compare the degree of their prediction success with other approaches such as:

Persistence: The most commonly used reference model for short term forecasting of wind is the Persistence method which assumes that: "*the conditions that existed at the beginning of the forecast period will continue or persist through to the end of the period*"[5]:

$$\tilde{y}_{t+h} = y_t \text{ where } y_t \text{ is the last measured value}$$

This method is not only the simplest modeling approach but is also the most economical to implement, and surprisingly accurate for short term forecasting (1 to 5 hours).

Moving Average: It is a widely used forecasting method constituting a generalization of the Persistence model. This simple approach is based on the average value of the variable over a specific number of preceding periods. In this paper we define the Moving Average method as

$$\tilde{y}_{t+h} = \frac{1}{h} \sum_{j=0}^{h-1} y_{t-j}, h = 1, 2, ..., n, \text{ where } h = b_{hd}$$

As h goes to infinity the Moving Average tends to the global average: $\tilde{y}_{t+h} = \overline{y}_t$ where $\overline{y}_t$ is the average of all the available measurements until time t.

Recurrent Neural Networks: A Recurrent Neural Network employs feedback connections and has the potential to represent "certain computational structures in a more parsimonious fashion"[10]. RNNs address the temporal relationship of their inputs by maintaining an internal state. In the latest bibliography regarding wind forecasting we find the RNN algorithm in [2] and because of its having been applied on the same data set displaying good performance we compare it to the FL-model.

To measure the precision of the models (i.e. how the model output is close to the real value) we make use of three statistical types of errors:

- The Mean Absolute Error: $MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \tilde{y_i}|$
- The Normalized Mean Square Error: $NMSE = \frac{\frac{1}{N}\sum_{i=1}^{N}\left\{\left(y_i - \widetilde{y_i}\right)^2\right\}}{\frac{1}{N}\sum_{i=1}^{N}\left\{\left(y_i - \frac{1}{N}\sum_{i=1}^{N} y_i\right)^2\right\}}$
- The Root Mean Square Error: $RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\{\left(y_i - \tilde{y_i}\right)^2\right\}}$

where y_i is the real value, $\tilde{y_i}$ is the model output and N is the number of tested values. Smaller values of the aforesaid uncertainty statistics denote better model performance. The reason for using these criteria is that they operate independently of application and target value specification while they are not biased towards models that over or under predict.

10.6 Experimental Results

The FL-Approach: Tables 1.2, 1.3 regarding both parameters of wind (speed and direction) display the comparison analysis among the reference models and the proposed approach. We use symbol "-" when the value is not provided. Explanation of the table columns is given in Table 1.1.

For the validation of the result, we adopt the *sliding window* where $2h \times 3$ is the window size for all the 3260 vectors and h is the time horizon within which the variable to be predicted lies. When we try to make a prediction h steps ahead from the moment t we take as input values the time frame $h-1$ steps back from t, i.e. $t, t-1, ..., t-(h-1)$ (each step refers to a period of 15 minutes).

The SV-Approach: The accuracy of each forecasting method is determined by its specific architecture [8]. The SVR algorithms offer a sufficient number of degrees of freedom in customizing models to a particular forecasting task. As we mentioned before, in order to evaluate the candidate models and determine

TABLE 1.1: EXPLANATION OF TABLES 1.2 AND 1.3

Column	Comments
1	Time horizon ahead in minutes & hours.
2	Time horizon ahead in steps.
3	Time steps backwards. This column does not refer to the RNN approach since this model needs to be trained at a sufficient number of steps back in order to minimize the difference between real and estimated value.
4-5-6	NMSE, RMSE and MAE of the persistence method.
7-8-9	NMSE, RMSE and MAE of the moving average method.
10-11-12-13	NMSE of the moving average method upon the sample data used in [2], NMSE, percentage improvement compared to the NMSE moving average model [2] and MAE of the RNN algorithm [2].
14-15-16-17	NMSE, percentage improvement compared to the NMSE moving average model, RMSE and MAE of the FL-prediction approach based on information provided by the station S_0.
18-19-20-21	NMSE, percentage improvement compared to the NMSE moving average model, RMSE and MAE of the FL-prediction approach based on information provided by all stations S_0, S_1, S_2.

TABLE 1.2
COMPARISON OF THE FL-APPROACH WITH THE REFERENCE MODELS IN PREDICTING WIND SPEED

Minutes ahead	Steps ahead	Look back length	Persistence			Moving Average			RNN				FL-Clustering Approach							
													S0				S0, S1, S2			
			NMSE	RMSE	MAE	NMSE	RMSE	MAE	MA NMSE	NMSE	%	MAE	NMSE	%	RMSE	MAE	NMSE	%	RMSE	MAE
15	1	1	0.0327	0.7415	0.5226	0.0327	0.7415	0.5226	0.0498	0.0366	26.55	-	0.0327	0.00	0.7415	0.5226	0.0327	0.00	0.7415	0.5226
30	2	2	0.0569	1.0848	0.7680	0.0629	1.1405	0.7972	0.1008	0.0746	25.98	-	0.0569	9.54	1.0848	0.7680	0.0569	9.54	1.0848	0.7680
60-1h	4	4	0.0985	1.4601	1.0268	0.1099	1.5425	1.0905	0.1778	0.1337	24.79	-	0.0976	11.19	1.4537	1.0200	0.0874	20.47	1.3754	0.9818
90	6	6	0.1162	1.7118	1.2201	0.1352	1.8466	1.3106	0.2448	0.1860	24.01	-	0.1158	14.35	1.7091	1.2108	0.1026	24.11	1.6087	1.1454
120-2h	8	8	0.1868	1.9428	1.3856	0.2180	2.0988	1.4985	0.3239	0.2474	23.61	-	0.1852	15.05	1.9345	1.3724	0.1650	24.31	1.8259	1.2903
150	10	10	0.1938	2.1339	1.5332	0.2250	2.2993	1.6317	0.3830	0.2983	22.11	-	0.1910	15.11	2.1185	1.5066	0.1708	24.09	2.0036	1.4120
180-3h	12	12	0.2253	2.3085	1.6519	0.2588	2.4742	1.7503	0.4292	0.3392	20.98	-	0.2208	14.68	2.2851	1.6154	0.1993	22.99	2.1711	1.5157
360-6h	24	24	0.5807	3.0368	2.1845	0.7182	3.3771	2.4654	-	-	-	-	0.5855	18.48	3.0494	2.1909	0.5443	24.21	2.9399	2.0886
720-12h	48	48	0.9843	4.1798	3.1322	1.0909	4.4003	3.2207	-	-	-	-	-	-	-	-	0.8570	21.44	3.9002	2.8951

TABLE 1.3
COMPARISON OF THE FL-APPROACH WITH THE REFERENCE MODELS IN PREDICTING WIND DIRECTION

Minutes ahead	Steps ahead	Look back length	Persistence			Moving Average			RNN				FL-Clustering Approach							
													S0				S0, S1, S2			
			NMSE	RMSE	MAE	NMSE	RMSE	MAE	MA NMSE	NMSE	%	MAE	NMSE	%	RMSE	MAE	NMSE	%	RMSE	MAE
15	1	1	0.0317	0.2601	0.1288	0.0317	0.2601	0.1288	-	-	-	-	0.0317	0.00	0.2601	0.1288	0.0317	0.00	0.2601	0.1288
30	2	2	0.058	0.3687	0.1904	0.0618	0.3806	0.2018	-	-	-	-	0.0580	6.15	0.3687	0.1904	0.0580	6.15	0.3687	0.1904
60-1h	4	4	0.0964	0.4788	0.2632	0.0968	0.4800	0.2738	-	-	-	-	0.0901	6.92	0.4631	0.2609	0.0821	15.19	0.4421	0.2869
90	6	6	0.1721	0.5335	0.3014	0.1801	0.5458	0.3268	-	-	-	-	0.1602	11.05	0.5148	0.3020	0.1389	22.88	0.4793	0.3131
120-2h	8	8	0.1958	0.5874	0.3408	0.2099	0.6081	0.3792	-	-	-	-	0.1821	13.24	0.5664	0.3441	0.1518	27.68	0.5172	0.3422
150	10	10	0.2233	0.6338	0.3749	0.2444	0.6631	0.4290	-	-	-	-	0.2074	15.14	0.6108	0.3818	0.1692	30.77	0.5517	0.3693
180-3h	12	12	0.1997	0.6769	0.4117	0.2210	0.7120	0.4751	-	-	-	-	0.1850	16.29	0.6516	0.4206	0.1483	32.90	0.5833	0.3975
360-6h	24	24	0.8155	0.8783	0.5975	0.8650	0.9046	0.6678	-	-	-	-	0.7335	15.20	0.8330	0.5935	0.6003	30.60	0.7536	0.5492
720-12h	48	48	0.5219	1.0702	0.7925	0.3507	0.8772	0.6753	-	-	-	-	-	-	-	-	0.3249	7.36	0.8653	0.6770

the suitable architecture we adopt the leave-one-out cross-validation method; we select these ones with the lowest error on the validation data set.

The following Tables (2.1, 3.1, 4.1) display the performance of the proposed SV-models. We show these results which correspond to look back length $b_{hd} = 3$ and 6 and to time horizon $h = 1, 6, 20, 40, 80$ and 160 steps ahead. At each instant the first line of the tables refers to $\tilde{x}_{S_0}(t_0 + h)$ and the second line refers to $pred_{S_0}(t_0 + h)$.

TABLE 2.1
COMPARISON OF SV MODELS (1-DIMENSIONAL INPUT VECTORS) WITH THE REFERENCE MODELS

Minutes ahead	Steps ahead	Look back length	Persistence			Moving Average			SV Models											
									S0 svs			S0 with kernel			S0, S1, S2 svs			S0, S1, S2 with kernel		
			NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE
15	1	3	0.03	0.09	0.06	0.06	0.13	0.09	0.10	0.17	0.12	2.07	0.76	0.18	0.21	0.24	0.19	0.11	0.17	0.12
15	1	3	0.03	0.09	0.06	0.06	0.13	0.09	0.05	0.11	0.08	0.54	0.39	0.11	0.07	0.14	0.11	0.05	0.12	0.08
90-1.5h	6	3	0.13	0.21	0.15	0.14	0.22	0.16	0.17	0.24	0.17	1.78	0.77	0.23	0.21	0.26	0.20	0.18	0.25	0.18
90-1.5h	6	3	0.13	0.21	0.15	0.14	0.22	0.16	0.14	0.21	0.15	0.53	0.42	0.18	0.13	0.21	0.15	0.14	0.22	0.16
300-5h	20	3	0.33	0.35	0.25	0.34	0.35	0.25	0.36	0.36	0.26	1.85	0.81	0.31	0.34	0.35	0.25	0.35	0.35	0.26
300-5h	20	3	0.33	0.35	0.25	0.34	0.35	0.25	0.33	0.35	0.25	0.70	0.50	0.27	0.30	0.33	0.23	0.33	0.34	0.25
600-10h	40	3	0.93	0.47	0.35	0.94	0.47	0.35	0.97	0.48	0.36	3.11	0.87	0.39	0.87	0.46	0.33	0.92	0.47	0.36
600-10h	40	3	0.93	0.47	0.35	0.94	0.47	0.35	0.93	0.47	0.35	1.43	0.59	0.36	0.84	0.45	0.33	0.90	0.47	0.35
1200-20h	80	3	1.52	0.61	0.44	1.51	0.60	0.44	1.53	0.61	0.44	3.64	0.94	0.47	1.35	0.57	0.42	1.43	0.59	0.44
1200-20h	80	3	1.52	0.61	0.44	1.51	0.60	0.44	1.51	0.60	0.44	1.98	0.69	0.45	1.37	0.58	0.42	1.45	0.59	0.44
2400-40h	160	3	1.56	0.56	0.43	1.52	0.55	0.42	1.52	0.56	0.42	4.11	0.91	0.45	1.43	0.54	0.40	1.43	0.54	0.42
2400-40h	160	3	1.56	0.56	0.43	1.52	0.55	0.42	1.52	0.55	0.42	2.11	0.65	0.43	1.42	0.54	0.41	1.46	0.54	0.42

The algorithms S0 svs and S0 with kernel use as input data only the measurements of station S_0. The S0 svs and S0,S1,S2 svs are M2.$a_{(ii)}$ algorithms while the S0 with kernel and S0,S1,S2 with kernel are M1.$a_{(ii)}$ algorithms.

TABLE 2.2
IMPROVEMENT (%) OF SV MODELS (1-DIMENSIONAL INPUT VECTORS) AND PERSISTENCE RELATED TO MOVING AVERAGE

Minutes ahead	Steps ahead	Look back length	Persistence			SV Models											
						S0 svs			S0 with kernel			S0, S1, S2 svs			S0, S1, S2 with kernel		
			NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE
15	1	3	46.60	26.93	26.51	-76.26	-32.86	-35.70	-3548.56	-504.49	-107.44	-268.02	-91.98	-114.13	-95.44	-39.80	-38.01
15	1	3	46.60	26.93	26.51	17.53	9.12	7.78	-847.58	-208.06	-27.18	-21.31	-10.23	-20.50	13.44	6.96	4.86
90-1.5h	6	3	8.29	4.24	3.78	-20.02	-9.55	-10.53	-1136.02	-251.57	-50.36	-44.34	-20.14	-26.89	-25.05	-11.74	-16.57
90-1.5h	6	3	8.29	4.24	3.78	4.63	2.34	2.23	-270.65	-92.52	-14.66	10.92	5.62	3.64	3.93	2.06	-0.96
300-5h	20	3	1.45	0.73	0.21	-6.55	-3.30	-3.71	-447.12	-134.07	-23.86	-0.52	-0.33	-1.04	-4.02	-1.99	-6.18
300-5h	20	3	1.45	0.73	0.21	1.51	0.69	0.72	-105.6	-43.49	-6.94	11.91	6.08	6.83	3.62	1.83	-0.25
600-10h	40	3	0.79	0.40	0.77	-3.56	-1.84	-2.21	-231.99	-82.34	-11.17	7.44	3.72	5.44	1.67	0.84	-1.41
600-10h	40	3	0.79	0.40	0.77	0.86	0.36	0.40	-52.10	-23.42	-1.87	10.36	5.25	7.25	3.88	1.96	1.35
1200-20h	80	3	-0.68	-0.34	-1.01	-1.59	-0.87	-0.94	-141.07	-55.38	-6.80	10.86	5.52	4.94	5.59	2.83	0.07
1200-20h	80	3	-0.68	-0.34	-1.01	0.31	0.08	-0.04	-31.24	-14.64	-1.73	9.02	4.55	5.34	4.09	2.06	0.78
2400-40h	160	3	-2.91	-1.44	-1.77	-0.46	-0.26	-0.15	-171.38	-64.78	-6.85	5.88	2.96	3.67	5.74	2.91	1.13
2400-40h	160	3	-2.91	-1.44	-1.77	0.03	-0.01	-0.01	-38.98	-17.92	-1.89	6.14	3.09	3.15	3.38	1.71	0.84

TABLE 2.3
IMPROVEMENT (%) OF SV MODELS (1-DIMENSIONAL INPUT VECTORS) AND MOVING AVERAGE RELATED TO PERSISTENCE

Minutes ahead	Steps ahead	Look back length	Moving Average			SV Models											
						S0 svs			S0 with kernel			S0, S1, S2 svs			S0, S1, S2 with kernel		
			NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE
15	1	3	-87.27	-36.85	-36.07	-230.09	-81.82	-84.64	-6732.72	-727.23	-182.26	-589.20	-162.72	-191.36	-266.01	-91.31	-87.79
15	1	3	-87.27	-36.85	-36.07	-54.43	-24.37	-25.48	-1674.55	-321.57	-73.05	-127.19	-50.84	-63.96	-62.10	-27.32	-29.45
90-1.5h	6	3	-9.04	-4.42	-3.93	-30.87	-14.40	-14.88	-1247.78	-267.12	-56.27	-57.39	-25.46	-31.87	-36.35	-16.69	-21.15
90-1.5h	6	3	-9.04	-4.42	-3.93	-4.00	-1.98	-1.61	-304.16	-101.04	-19.17	2.87	1.45	-0.14	-4.75	-2.28	-4.92
300-5h	20	3	-1.47	-0.73	-0.21	-8.12	-4.06	-3.92	-455.18	-135.79	-24.12	-2.00	-1.07	-1.25	-5.55	-2.74	-6.40
300-5h	20	3	-1.47	-0.73	-0.21	0.06	-0.04	0.51	-108.62	-44.54	-7.17	10.61	5.39	6.63	2.21	1.11	-0.46
600-10h	40	3	-0.80	-0.40	-0.78	-4.39	-2.25	-3.01	-234.64	-83.07	-12.04	6.71	3.34	4.70	0.89	0.45	-2.20
600-10h	40	3	-0.80	-0.40	-0.78	0.07	-0.04	-0.38	-53.31	-23.91	-2.66	9.65	4.87	6.53	3.11	1.57	0.58
1200-20h	80	3	0.68	0.34	1.00	-0.90	-0.52	0.07	-139.43	-54.85	-5.73	11.47	5.84	5.88	6.23	3.17	1.07
1200-20h	80	3	0.68	0.34	1.00	0.98	0.42	0.95	-30.34	-14.25	-0.71	9.64	4.87	6.28	4.74	2.40	1.77
2400-40h	160	3	2.83	1.42	1.74	2.38	1.17	1.58	-163.71	-62.43	-4.99	8.54	4.34	5.34	8.40	4.29	2.85
2400-40h	160	3	2.83	1.42	1.74	2.86	1.41	1.73	-35.05	-16.24	-0.12	8.79	4.47	4.83	6.11	3.11	2.56

Tables 2.2, 2.3, 3.2, 3.3, 4.2 and 4.3 provide the improvement percentage with respect to both reference models, i.e. Persistence and Moving Average. For example, the improvement percentage of SV model $S_0, S_1, S_2 svs$ over the

TABLE 3.1
COMPARISON OF SV MODELS (3-DIMENSIONAL INPUT VECTORS) WITH THE REFERENCE MODELS

Minutes ahead	Steps ahead	Look back length	Persistence			Moving Average			SV Models					
									S0,S1,S2-3d svs			S0,S1,S2-3d kernel		
			NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE
15	1	3	0.03	0.09	0.06	0.06	0.13	0.09	0.30	0.29	0.22	0.48	0.36	0.30
15	1	3	0.03	0.09	0.06	0.06	0.13	0.09	0.09	0.16	0.12	0.14	0.19	0.16
90-1.5h	6	3	0.13	0.21	0.15	0.14	0.22	0.16	0.27	0.30	0.23	0.43	0.38	0.31
90-1.5h	6	3	0.13	0.21	0.15	0.14	0.22	0.16	0.14	0.21	0.16	0.18	0.25	0.20
300-5h	20	3	0.33	0.35	0.25	0.34	0.35	0.25	0.37	0.36	0.27	0.50	0.42	0.35
300-5h	20	3	0.33	0.35	0.25	0.34	0.35	0.25	0.29	0.32	0.23	0.33	0.34	0.27
600-10h	40	3	0.93	0.47	0.35	0.94	0.47	0.35	0.84	0.45	0.33	0.96	0.48	0.40
600-10h	40	3	0.93	0.47	0.35	0.94	0.47	0.35	0.80	0.44	0.32	0.81	0.44	0.35
1200-20h	80	3	1.52	0.61	0.44	1.51	0.60	0.44	1.23	0.55	0.40	1.23	0.54	0.45
1200-20h	80	3	1.52	0.61	0.44	1.51	0.60	0.44	1.29	0.56	0.40	1.24	0.55	0.43
2400-40h	160	3	1.56	0.56	0.43	1.52	0.55	0.42	1.39	0.53	0.39	1.34	0.52	0.44
2400-40h	160	3	1.56	0.56	0.43	1.52	0.55	0.42	1.37	0.53	0.40	1.28	0.51	0.40

The S0,S1,S2-3d svs is a M2.$a_{(i)}$ algorithm and the S0,S1,S2-3d kernel is a M1.$a_{(i)}$ algorithm

TABLE 3.2
IMPROVEMENT (%) OF SV MODELS (3-DIMENSIONAL INPUT VECTORS) AND PERSISTENCE RELATED TO MOVING AVERAGE

Minutes ahead	Steps ahead	Look back length	Persistence			SV Models					
						S0,S1,S2-3d svs			S0,S1,S2-3d kernel		
			NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE
15	1	3	46.60	26.93	26.51	-428.68	-130.10	-149.67	-743.65	-190.68	-238.73
15	1	3	46.60	26.93	26.51	-57.94	-25.77	-35.41	-140.94	-55.34	-80.86
90-1.5h	6	3	8.29	4.24	3.78	-84.84	-35.95	-44.66	-198.45	-72.76	-101.10
90-1.5h	6	3	8.29	4.24	3.78	4.80	2.43	-1.03	-27.70	-13.01	-28.47
300-5h	20	3	1.45	0.73	0.21	-8.84	-4.40	-7.21	-47.67	-21.61	-41.31
300-5h	20	3	1.45	0.73	0.21	13.38	6.86	6.49	3.61	1.75	-8.58
600-10h	40	3	0.79	0.40	0.77	10.28	5.21	6.20	-2.25	-1.19	-14.23
600-10h	40	3	0.79	0.40	0.77	14.81	7.63	9.32	13.70	7.03	1.29
1200-20h	80	3	-0.68	-0.34	-1.01	18.27	9.53	8.71	18.39	9.82	-3.17
1200-20h	80	3	-0.68	-0.34	-1.01	14.61	7.53	8.17	17.63	9.17	3.07
2400-40h	160	3	-2.91	-1.44	-1.77	8.41	4.28	6.44	11.80	6.06	-3.98
2400-40h	160	3	-2.91	-1.44	-1.77	9.59	4.89	4.83	15.26	7.92	3.74

TABLE 3.3
IMPROVEMENT (%) OF SV MODELS (3-DIMENSIONAL INPUT VECTORS) AND MOVING AVERAGE RELATED TO PERSISTENCE

Minutes ahead	Steps ahead	Look back length	Moving Average			SV Models					
						S0,S1,S2-3d svs			S0,S1,S2-3d kernel		
			NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE
15	1	3	-87.27	-36.85	-36.07	-890.08	-214.89	-239.71	-1479.93	-297.78	-360.89
15	1	3	-87.27	-36.85	-36.07	-195.78	-72.11	-84.24	-351.21	-112.58	-146.08
90-1.5h	6	3	-9.04	-4.42	-3.93	-101.55	-41.97	-50.34	-225.44	-80.40	-109.00
90-1.5h	6	3	-9.04	-4.42	-3.93	-3.81	-1.89	-5.00	-39.25	-18.00	-33.51
300-5h	20	3	-1.47	-0.73	-0.21	-10.45	-5.17	-7.43	-49.84	-22.50	-41.61
300-5h	20	3	-1.47	-0.73	-0.21	12.10	6.18	6.29	2.19	1.03	-8.80
600-10h	40	3	-0.80	-0.40	-0.78	9.56	4.83	5.47	-3.06	-1.59	-15.12
600-10h	40	3	-0.80	-0.40	-0.78	14.13	7.26	8.62	13.01	6.66	0.52
1200-20h	80	3	0.68	0.34	1.00	18.83	9.84	9.62	18.94	10.13	-2.15
1200-20h	80	3	0.68	0.34	1.00	15.19	7.84	9.08	18.19	9.48	4.04
2400-40h	160	3	2.83	1.42	1.74	11.00	5.64	8.07	14.30	7.40	-2.18
2400-40h	160	3	2.83	1.42	1.74	12.15	6.25	6.48	17.65	9.23	5.42

Persistence using the criterion NMSE at 1200 minutes ahead and $\tilde{x}_{S_0}(t_0+80)$ (Table 2.1) is $\frac{1.52-1.35}{1.52} \times 100\% = 11.47\%$ (Table 2.3).

However, it is common that the performance varies in the data set and, indeed, there is not an a priori reason to believe that accurate predictions

TABLE 4.1
COMPARISON OF SV MODELS (1-DIMENSIONAL & 3-DIMENSIONAL INPUT VECTORS) WITH THE REFERENCE MODELS

Minutes ahead	Steps ahead	Look back length	Persistence			Moving Average			SV Models											
									S0,S1,S2-1d svs			S0,S1,S2-1d kernel			S0,S1,S2-3d svs			S0,S1,S2-3d kernel		
			NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE
15	1	6	0.03	0.09	0.06	0.07	0.15	0.10	0.19	0.25	0.2	0.07	0.16	0.11	0.26	0.29	0.22	0.35	0.34	0.27
15	1	6	0.03	0.09	0.06	0.07	0.15	0.10	0.06	0.14	0.11	0.04	0.11	0.08	0.08	0.16	0.12	0.10	0.18	0.15
90-1.5h	6	6	0.12	0.21	0.15	0.14	0.23	0.16	0.20	0.28	0.21	0.15	0.24	0.17	0.25	0.31	0.23	0.34	0.36	0.29
90-1.5h	6	6	0.12	0.21	0.15	0.14	0.23	0.16	0.12	0.21	0.15	0.12	0.21	0.15	0.13	0.22	0.16	0.16	0.24	0.19
300-5h	20	6	0.32	0.35	0.25	0.33	0.35	0.25	0.34	0.36	0.26	0.33	0.36	0.25	0.36	0.37	0.27	0.45	0.41	0.34
300-5h	20	6	0.32	0.35	0.25	0.33	0.35	0.25	0.29	0.33	0.23	0.31	0.34	0.25	0.28	0.33	0.23	0.31	0.34	0.27
600-10h	40	6	1.00	0.47	0.35	1.02	0.48	0.35	0.95	0.46	0.33	1.02	0.48	0.36	0.91	0.45	0.33	1.01	0.48	0.39
600-10h	40	6	1.00	0.47	0.35	1.02	0.48	0.35	0.90	0.45	0.33	0.99	0.47	0.35	0.86	0.44	0.32	0.88	0.44	0.35
1200-20h	80	6	1.43	0.61	0.44	1.41	0.60	0.44	1.27	0.57	0.42	1.41	0.60	0.44	1.16	0.55	0.40	1.19	0.55	0.45
1200-20h	80	6	1.43	0.61	0.44	1.41	0.60	0.44	1.29	0.58	0.41	1.40	0.60	0.44	1.21	0.56	0.40	1.20	0.56	0.43
2400-40h	160	6	1.22	0.56	0.43	1.16	0.55	0.41	1.10	0.53	0.4	1.17	0.55	0.42	1.08	0.53	0.39	1.06	0.52	0.43
2400-40h	160	6	1.22	0.56	0.43	1.16	0.55	0.41	1.10	0.53	0.4	1.18	0.55	0.42	1.07	0.53	0.40	1.03	0.52	0.41

S0,S1,S2-1d svs: $M2.a_{(ii)}$, S0,S1,S2-1d kernel: $M1.a_{(ii)}$, S0,S1,S2-3d svs: $M2.a_{(i)}$, S0,S1,S2-3d kernel: $M1.a_{(i)}$

TABLE 4.2
IMPROVEMENT (%) OF SV MODELS (1-DIMENSIONAL & 3-DIMENSIONAL INPUT VECTORS) AND PERSISTENCE RELATED TO MOVING AVERAGE

Minutes ahead	Steps ahead	Look back length	Persistence			SV Models											
						S0,S1,S2-1d svs			S0,S1,S2-1d kernel			S0,S1,S2-3d svs			S0,S1,S2-3d kernel		
			NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE
15	1	6	61.43	37.90	37.93	-191.18	-70.64	-89.53	-12.82	-6.22	-5.01	-294.65	-98.66	-116.41	-432.77	-130.82	-161.50
15	1	6	61.43	37.90	37.93	6.15	3.12	-5.20	43.76	25.01	24.85	-18.08	-8.66	-17.19	-55.56	-24.72	-41.75
90-1.5h	6	6	16.86	8.82	8.52	-44.12	-20.05	-26.41	-4.17	-2.06	-2.71	-77.03	-33.05	-41.36	-143.01	-55.89	-78.34
90-1.5h	6	6	16.86	8.82	8.52	15.22	7.92	6.37	14.26	7.40	7.26	9.92	5.09	2.11	-10.05	-4.91	-17.61
300-5h	20	6	4.34	2.19	1.38	-1.61	-0.80	-1.84	-0.76	-0.38	-0.99	-7.68	-3.77	-6.94	-34.92	-16.15	-33.51
300-5h	20	6	4.34	2.19	1.38	13.67	7.09	7.72	5.16	2.61	2.27	15.42	8.03	7.63	7.28	3.71	-5.85
600-10h	40	6	2.09	1.05	1.81	7.23	3.68	5.72	-0.16	-0.08	-0.27	11.34	5.84	6.99	0.62	0.31	-11.03
600-10h	40	6	2.09	1.05	1.81	11.42	5.88	8.15	2.81	1.41	1.70	16.10	8.40	10.42	13.68	7.09	2.33
1200-20h	80	6	-1.15	-0.58	-1.73	10.30	5.29	4.50	0.11	0.05	-0.53	18.13	9.52	8.25	16.13	8.42	-3.74
1200-20h	80	6	-1.15	-0.58	-1.73	8.86	4.53	5.08	0.64	0.32	-0.36	14.44	7.50	7.77	15.22	7.92	2.22
2400-40h	160	6	-5.16	-2.55	-3.23	5.33	2.70	3.28	-0.70	-0.35	-0.86	7.24	3.69	5.68	8.84	4.52	-4.99
2400-40h	160	6	-5.16	-2.55	-3.23	5.26	2.67	2.46	-1.52	-0.75	-1.23	8.19	4.18	3.92	11.37	5.85	1.80

TABLE 4.3
IMPROVEMENT (%) OF SV MODELS (1-DIMENSIONAL & 3-DIMENSIONAL INPUT VECTORS) AND MOVING AVERAGE RELATED TO PERSISTENCE

Minutes ahead	Steps ahead	Look back length	Moving Average			SV Models											
						S0,S1,S2-1d svs			S0,S1,S2-1d kernel			S0,S1,S2-3d svs			S0,S1,S2-3d kernel		
			NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE	NMSE	RMSE	MAE
15	1	6	-159.30	-61.03	-61.10	-655.04	-174.78	-205.34	-192.55	-71.04	-69.17	-923.33	-219.90	-248.64	-1281.46	-271.68	-321.28
15	1	6	-159.30	-61.03	-61.10	-143.36	-56.00	-69.49	-45.82	-20.76	-21.08	-206.17	-74.98	-88.81	-303.36	-100.84	-128.37
90-1.5h	6	6	-20.28	-9.67	-9.31	-73.35	-31.66	-38.17	-25.29	-11.94	-12.27	-112.94	-45.92	-54.52	-192.29	-70.97	-94.94
90-1.5h	6	6	-20.28	-9.67	-9.31	-1.98	-0.98	-2.34	-3.13	-1.55	-1.37	-8.35	-4.09	-7.00	-32.37	-15.05	-28.56
300-5h	20	6	-4.53	-2.24	-1.40	-6.21	-3.06	-3.26	-5.33	-2.63	-2.40	-12.56	-6.10	-8.44	-41.03	-18.76	-35.39
300-5h	20	6	-4.53	-2.24	-1.40	9.76	5.01	6.43	0.86	0.43	0.90	11.59	5.97	6.33	3.08	1.55	-7.34
600-10h	40	6	-2.13	-1.06	-1.84	5.25	2.66	3.99	-2.29	-1.14	-2.12	9.44	4.84	5.27	-1.51	-0.75	-13.07
600-10h	40	6	-2.13	-1.06	-1.84	9.53	4.88	6.45	0.73	0.37	-0.12	14.31	7.43	8.76	11.84	6.11	0.52
1200-20h	80	6	1.14	0.57	1.70	11.33	5.83	6.12	1.25	0.63	1.18	19.06	10.03	9.81	17.09	8.94	-1.98
1200-20h	80	6	1.14	0.57	1.70	9.90	5.08	6.69	1.77	0.89	1.35	15.41	8.03	9.34	16.18	8.45	3.88
2400-40h	160	6	4.91	2.48	3.13	9.98	5.12	6.31	4.24	2.14	2.30	11.79	6.08	8.63	13.31	6.89	-1.71
2400-40h	160	6	4.91	2.48	3.13	9.91	5.08	5.51	3.46	1.75	1.94	12.70	6.56	6.92	15.71	8.19	4.87

can be made at every single time step; in a mainly non-linear system, there may be islands of predictability implanted in a sea of unpredictable or chaotic behavior [18].

10.7 Discussion

Some interesting points emerging from the comparison analysis are given in the following.

The FL-Approach:

1. The Moving Average method exhibits the poorest performance whereas the RNN and the FL_{S_0,S_1,S_2}-approach demonstrate significant improvement in comparison with it and the Persistence method. Up to a certain degree, this result is expectable since the RNN and FL_{S_0,S_1,S_2} exploit the correlation among the three meteorological stations.
2. Additionally, the FL_{S_0}-approach utilizing only the historical data of station S_0 is not better than the Persistence in predicting wind speed, while it behaves better in wind direction predictions. This can be explained by the fact that the wind direction displays noticeably less variation than wind speed.
3. In the first two steps (15 and 30 minutes) the FL-model cannot beat the Persistence method because a crucial factor in FL-approach is the formulation of rectangles and in this time horizon the created rectangles are few and contain no significant information in order to result in more precise predictions. The RNN model seems to perform better in this time window but we have to consider the fact that it exploits more information (steps back) in the training phase.
4. The percentage improvement of both FL_{S_0,S_1,S_2} and RNN algorithms compared with MA shows a slightly better performance of the former especially after 4 steps ahead. For instance, at $h = 8$ the FL_{S_0,S_1,S_2}-model presents a 24.31% improvement while the respective percentage of RNN is 23.61%. Similarly, at $h = 12$ the FL_{S_0,S_1,S_2}-model displays a 22.99% improvement while the respective percentage of RNN is 20.98%. However, the RNN model and the FL-approach are not totally comparable since they refer to slightly different samples. However, we have to stress the fact that the FL-approach does not need any training while it is easily tuned (the vigilance parameter is set equal to the standard variation of the data sample and the curve correction is eventuated by means of the last measured value) and presents both low computational and time cost.
5. Another advantage of the FL-model for the user is data compression and the effortless extraction of simple rules (symbolic knowledge which has meaning to humans) where the prediction of a new fact is calculated from the weighted average of the created rectangles and its average with the last measured value.

In Fig. 10.2, we indicate how the FL-model approximates the real curve.

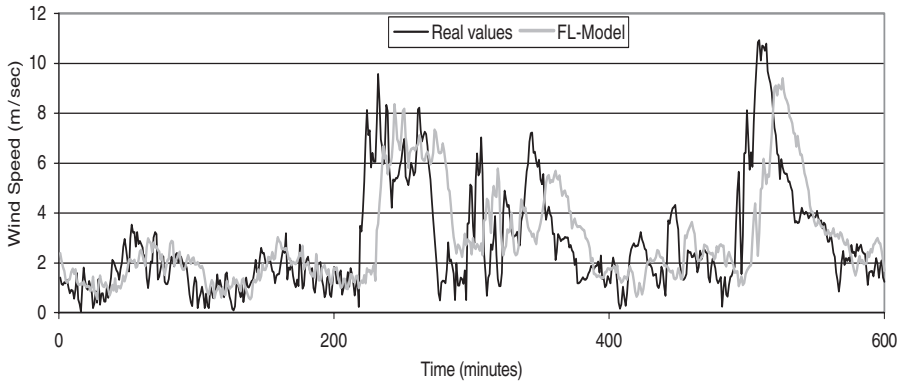

Fig. 10.2. Approximation of real curve by the FL-model

The SV-Approach: We test the SV-models only for wind parameter prediction.

1. The usage of small time frames (b_{hd}=1 or b_{hd}=6) limits the memory requirements for storage of the kernel matrix whilst the impact of past outliers is diminished. Moreover, after many tests we ascertained that greater values for b_{hd} do not give better results.
2. The two algorithms $S0$ *svs* and $S0$ *with kernel* cannot outperform any other model. This indicates the need for additional information.
3. In most cases the $pred_{S_0}(t_0 + h)$ gives better results than $\tilde{x}_{S_0}(t_0 + h)$. The latter seems to have improved after 80 steps.
4. The approach with the 3-dimensional input vectors exhibits better performance related to models with 1-dimensional input vectors.
5. The greater generalization ability is displayed by $S0, S1, S2$ *svs*, $S0, S1$, $S2-3d$ *svs* and $S0, S1, S2-3d$ *kernel* especially after 20 steps ahead. We note that these models take advantage of the spatial correlation among the three stations.

We point out that the performance of SV algorithms depends crucially on an appropriate choice of parameters. Although several different approaches exist for their selection, the issue of how to practically select a good set is still far from being resolved.

10.8 Conclusions

This paper applies comparatively two prediction methodologies (the Fuzzy Lattice Neurocomputing and the Support Vector Regression) to a real-world problem that of wind prediction which is a very demanding task. Short (0–6 hours) and long (>6 hours) time forecasting of wind variations is still an open problem.

FLN and SVR have sound mathematical foundations. The results produced depend on the selection of the valuation function in the FLN case and the kernel function in the SVR case. There are few parameters to adjust. In FLN we just tune the vigilance parameter while in SVR we adjust the capacity control and the kernel parameters (the width of the Gaussian kernel in RBF case). Additionally, they can be applied to different types of data such as real-valued vectors, symbols, images, text etc. Yet, only the FL technique manages to cope with different types of data simultaneously by considering the Cartesian product of multiple lattices. Both methodologies seek for a sparse representation of the final solution, i.e. FLN utilizes intervals whereas SVR uses the support vectors representing the bounds of the candidate solution regions standing for diverse classes. The problem of *missing* and *don't care* attribute values in the data has not been explicitly addressed within the SVR methodology whereas in the FL-framework *missing* values have been replaced by the least element and *don't care* values can be replaced by the greatest element of the corresponding lattice. Finally, the model selection problem is still an open research topic since both methodologies employ a quite expensive way (cross validation) to select the model parameters.

We have assessed the performance of the proposed models by comparing them with two well-known reference techniques employed widely as benchmarks to time-series analysis, the Persistence and the Moving Average method, and additionally to a recurrent neural network. In general the FL-approach demonstrates a satisfactory and constant performance. Future work involves the testing of the FL-approach in dealing with data typically containing high noise and significant non-stationarity. Finally, a synthesis of these two effective methodologies (FL and SVR) might result in superior algorithms.

References

1. Alexiadis MC (2002) Wind Speed Prediction at Aeolic Parks. PhD Thesis, Aristotle Univ of Thessaloniki, Greece
2. Barbounis A (2005) Optimized Real-Time Learning Algorithms for the Training of RNNs and Fuzzy RNNs: Application on Wind Speed and Wind Power Prediction at Aeolic Parks. PhD Thesis, Aristotle Univ of Thessaloniki, Greece
3. Beyer HG, Luther J, Steinberger-Willms R (1993) Power fluctuations in spatially dispersed wind turbine systems. Solar Energy 50:297–306
4. Birkhoff G (1967) Lattice Theory. American Math Society, Col Pub 25
5. Box J, Jenkins G (1976) Time Series Analysis, Forecasting and Control. HolDay
6. Carpenter G, Grossberg S (1987) A massively parallel architecture for self-organizing neural pattern recognition machine. Computer Vision, Graphics and Image Understanding 37:54–115
7. Cortes C, Vapnik V (1995) Support Vector Networks. Mach Learn 20:273–297
8. Cristianini N, Shawe-Taylor J (2000) An Introduction to Support Vector Machines (and other kernel-based learning methods). Cambridge Univ Press

9. Daniel AR, Chen AA (1991) Stochastic simulation and forecasting of hourly average wind speed sequences in Jamaica. Sol Energy 46:1–11
10. Elman JL (1991) Distributed representations, simple recurrent networks, and grammatical structure. Machine Learning 7(2/3):195–226
11. Kaburlasos VG, Petridis V (2000) Fuzzy lattice neurocomputing (FLN) models. Neural Networks 13(10):1145–1170
12. Kaburlasos VG, Petridis V (2002) Learning and decision-making in the framework of fuzzy lattices. In: Jain LC, Kacprzyk J (eds) New Learning Paradigms in Soft Computing, ser Studies in Fuzziness and Soft Computing 84:55–96. Physica-Verlag, Heidelberg, Germany
13. Kariniotakis G, Pinson P, Siebert N, Giebel G, Barthelmie R (2004) The state of the art in short-term prediction of wind power from an offshore perspective. In: Proc SeaTechWeek
14. Karush W (1939) Minima of Functions of Several Variables with Inequalities as Side Constraints. MS Thesis, Dept of Mathematics, Univ of Chicago
15. Kuhn HW, Tucker AW (1951) Nonlinear programming. In: Proc 2nd Berkeley Symp on Mathematical Statistics and Probabilistics pp 481–492. Univ of California Press
16. Müller KR, Smola AJ, Rätsch G, Schölkopf B, Kohlmorgen J, Vapnik V (1997) Prediction time series with support vector machines. In: Proc Int Conf on Artificial Neural Networks
17. Nielsen TS, Madsen H (1997) Statistical methods for predicting wind power. In: Proc European Wind Energy Association Conference (EWEC) pp 755–758
18. Packard NH (1990) A genetic learning algorithm for the analysis of complex data. Complex Systems 4(5):543–572
19. Palomino I, Martin F (1995) A simple method for spatial interpolation of the wind in complex terrain. J Appl Meteorology 34(7):1678–1693
20. Petridis V, Kaburlasos VG (1999) Learning in the framework of fuzzy lattices. IEEE Transactions on Fuzzy Systems 7(4):422–440
21. Schlueter RA, Park GL, Bouwmeester R, Shu L, Lotfalian M, Rastgoufard P, Shayanfar A (1984) Simulation and assessment of wind array power variations based on simultaneous wind speed measurements. IEEE Trans Power App Syst 103:1008–1016
22. Smola AJ and Schölkopf B (2004) A tutorial on support vector regression. Statistics and Computing 14:199–222
23. Vapnik V (1995) The Nature of Statistical Learning Theory. Springer, NY
24. Vapnik V, Golowich SE, Smola A (1997) Support vector method for function approximation, regression estimation, and signal processing. In: Adv in Neural Information Processing Systems 9:281–287. The MIT Press, Cambridge
25. Zadeh LA (1965) Fuzzy sets. Information and Control 8:338–353