

A NEURO-FUZZY MODEL FOR WIND SPEED PREDICTION BASED ON STATISTICAL LEARNING THEORY

Elizabeta Lazarevska

*Faculty of Electrical Engineering and Information Technologies,
"Ss. Cyril and Methodius" University in Skopje,
Rugjer Boshkovik bb, P.O. box 574, 1001 Skopje, Republic of Macedonia
elizabeta.lazarevska@feit.ukim.edu.mk*

Abstract: Wind is free, clean, and renewable source of energy and is fast becoming a desired alternative to conventional energy resources such as fossil fuels. That is why more and more countries are intensifying their efforts in wind energy research and harnessing. Among other wind characteristics, wind speed is crucial for planning, designing and operating wind energy systems. This is the reason for much research in the field of wind speed modelling and prediction. There are many research papers dealing with the problem of forecasting the wind speed, which requires special attention because of time-varying, stochastic and intermittent nature of wind. It has been shown in literature that among the many proposed models for wind speed prediction, the models based on soft computing techniques such as artificial neural networks, neuro-fuzzy inference systems and machine learning are superior in terms of approximation accuracy. While there are many neural models for wind speed prediction that deploy different learning methods, and there are many hybrid models based on fuzzy logic, neural networks and genetic algorithms etc., the research conducted in this work has shown that practically there are no neural models based on relevance vector machine and no neuro-fuzzy models that apply RVM learning mechanism, which is state of the art technique. This paper presents possibly for the first time in literature a neuro-fuzzy model for wind speed prediction based on Vapnik's statistical learning theory, Tipping's relevance vector machine and Kim's fuzzy inference system. The model is a fuzzy inference system of a Takagi-Sugeno type that relies on extended relevance vector machine for learning its parameters and fuzzy rules. The wind speed is modeled by means of available meteorological data such as total solar radiation, ambient temperature, humidity, atmospheric pressure, etc. The performance of the model is validated through its performance index and compared to other fuzzy and neural models for wind speed prediction. The simulation results show clearly that the model possesses excellent features and the best performance in terms of accuracy.

Key words: wind speed prediction; neuro-fuzzy modelling; extended relevance vector machine; kernel function; relevance vectors

НЕВРО-ФАЗИ МОДЕЛ ЗА ПРЕДВИДУВАЊЕ НА БРЗИНАТА НА ВЕТЕРОТ БАЗИРАН ВРЗ СТАТИСТИЧКА ТЕОРИЈА НА УЧЕЊЕ

Апстракт: Ветерот е бесплатен, чист и обновлив извор на енергија и брзо прераснува во посакувана алтернатива на конвенционалните извори на енергија како што се фосилните горива. Затоа сè повеќе земји ги интензивираат напорите во насока на истражување и искористување на енергијата на ветерот. Меѓу другите карактеристики на ветерот, неговата брзина е клучна при планирањето, дизајнирањето и искористувањето на ветроенергетските системи. Ова е причината за многубројните истражувања во областа на моделирање и предвидување на брзината на ветерот. Голем број научни трудови се занимаваат со проблематиката на предвидување на брзината на ветерот, која бара огромно внимание поради временски променливата, стохастичка и интермитентна природа на ветерот. Во литературата е покажано дека меѓу многуте предложени модели за предвидување на брзината на ветерот, моделите базирани врз меките компјутерски техники, како што се вештачките невронски мрежи, невро-фази-инферентните системи и машинското учење, се супериорни од аспект на точноста на предвидувањата. Иако постојат многу невронски модели за предвидување на брзината на ветерот кои применуваат различни методи на учење и многу хибридни модели базирани врз фази логика, невронски мрежи, генетски алгоритми итн., спроведеното истражување во овој труд покажува дека практично не постојат невронски модели базирани врз релевантноста на механизмот на векторите и невро-фази-моделите кои применуваат МВР механизми на учење, кои се сметаат за најсовремени техники. Во овој труд, можеби за првпат во литературата, е претставен невро-фази модел за предвидување на брзината на ветерот базиран врз статистичката теорија на учење на Вапник, механизмот на вектори на релевантност на

Типинг и фази-логичкиот систем на Ким. Моделот е фази-логички систем од типот Такаки-Сугено кој се потпира на проширен механизам на вектори на релевантност за учење на параметрите и фази-правилата. Брзината на ветерот е моделирана според достапните метеоролошки податоци: вкупно сончево зрачење, амбиентална температура, влажност, атмосферски притисок итн. Перформансите на моделот се валидирани преку неговиот индекс на перформанси и се споредени со други фази и невронски модели за предвидување на брзина на ветерот. Резултатите од симулацијата јасно покажуваат дека моделот се одликува со одлични карактеристики и нуди најдобри перформанси од аспект на точноста.

Клучни зборови: предвидување на брзина на ветерот; фази-невронско моделирање; проширен механизам на вектори на релевантност; кернел функција; вектори на релевантност

INTRODUCTION

The negative environmental impact of excessive exploitation of fossil fuels and their depletion have led to increased interest in renewable and clean energy sources [1]. Wind is one such source due to the fact that wind energy is free, environmental friendly, and inexhaustible [2]. In order to meet the highly increased energy demands caused by modern ways of living and reduce negative environmental issues such as the global warming, more and more countries are assigning a high priority to wind energy harnessing [3]. As a matter of fact, wind energy is now considered as the fastest growing source and this trend is expected to continue [4].

The bottleneck of wind energy utilization is the time-varying, stochastic, intermittent, and complex nature of wind speed. It is well-known that there is a non-linear cubic relationship between wind speed and the power output of wind turbines [5], which means that only a small deviation in wind speed will result in a large deviation in wind power output of the wind turbines. Therefore, it is of utmost importance for wind energy systems to accurately measure and estimate wind speed at a given site [6–8]. Normally, engineers deploy anemometers for measuring wind speed. However, measurement of wind speed is considered the most difficult among various climatological variables. For one, in a wind farm multiple anemometers must be used since the wind speed varies from one wind turbine to another and for other, the masts for mounting cup anemometers, which are the accepted standard for resource assessment, inevitably become much taller as wind turbines grow in size, thus making the application of wind anemometers much more expensive. The high cost of wind anemometers discourages their widespread application, which is why engineers replace wind anemometers with digital wind speed estimators for broad applications, such as in wind farms [9–10].

Many wind speed estimation methods are presented in the literature as of the present moment [11–17]. They can be classified according different

criteria. One such classification is according to the adopted prediction period. Different time scale horizons have been applied for wind speed prediction ranging from several minutes to several days, but all of them can be classified according to Table 1 [18].

Table 1

Different wind speed prediction horizons

Class	Time scale	Main application
Very short range forecasting	A few seconds to 30 min	Turbine active control
Short range forecasting	30 min to 6 hours	Power system management, energy trading
Medium range forecasting	6 hours to 1 day	Wind generator on/off decision, operational security, electric market purposes
Long range forecasting	1 day to 1 week	Unit commitment decision, maintenance scheduling

Another classification of wind speed prediction models can be performed based on the applied method, as is shown in Table 2.

The persistence is the simplest method for wind speed prediction that is based on the assumption of a strong correlation between present and future values of wind speed. In other words, it assumes that the future values of wind speed equal the present value. Despite its simplicity, the model is as good as any for short term predictions. Its accuracy decreases rapidly with increasing prediction time scale.

The numerical weather prediction (NWP) models use mathematical models of the atmosphere and oceans to predict the weather based on current weather conditions. The complex mathematical calculations involved in modern weather prediction require super powerful computers, and yet, the forecasting ability of NWP does not extend past several days, due to the errors caused by the chaotic nature of the partial differential equations governing the atmosphere.

Table 2
Classification of Wind Speed Prediction Models

P r e d i c t i o n w i n d s p e e d	Persistence	
	Numerical weather prediction	
	Regression analysis	Linear regression
		Least squares (LS)
	Statistical	Nonlinear regression
		Algebraic curve fitting
		Autoregressive moving average (ARMA)
		Autoregressive integrated moving average (ARIMA)
		Bayesian model averaging (BMA)
		Grey predictor (GP)
	Soft computing	Artificial neural networks (ANN)
		Support vector machines (SVM)
Hybrid soft computing techniques		

Present understanding is that this chaotic behavior limits accurate forecasts to about 14 days regardless how accurate the input data are and how precise the model is.

Statistical prediction methods include regression, and time-series models. Regression analysis is frequently used for prediction and forecasting, in which domain its application substantially overlaps with the field of machine learning. Some of the most familiar methods are linear regression and ordinary least squares regression, which are considered parametric, in that the regression function is defined in terms of a finite number of unknown parameters that are estimated from the data. The regression models define the relation between past values of wind speed, as well as past and forecast values of meteorological variables, and wind speed measurements.

Along with the traditional forecasting methods, soft computing methods can also be used for wind speed prediction. Recent research works have focused on artificial neural networks (ANN), and support vector machines (SVM), which generally produce superior approximation performance compared to other forecasting techniques. The wind

speed models based on artificial intelligence techniques, such as NN and SVM, belong to the class of black-box models.

Hybrid models for short-term wind speed prediction unite different modelling techniques and approaches, such as NN and genetic algorithms (GA), or NN and fuzzy inference systems (FIS). Much more insight into different techniques for wind speed prediction can be found in [19–21].

This paper presents a new approach to wind speed forecasting based on fuzzy logic and neural network techniques. The proposed neuro-fuzzy model for prediction of wind speed is a fuzzy inference system (FIS) with a learning mechanism based on statistical learning theory and extended relevance vector machines (RVM). The conducted research has shown that there are less than few papers presenting wind speed forecasting using RVM [22, 23], and none presenting a neuro-fuzzy model of wind speed based on extended RVM. Reference [22] proposes a neural model for a day ahead wind speed prediction that utilizes relevance vector learning machine. The algorithm combines Gaussian kernel functions and polynomial kernel functions in order to obtain mixed kernel functions for RVM. The obtained model is compared to other neural models based on back propagation learning algorithm (BP) and SVM and [22] claims that the simulation results have shown that the RVM model is more effective and robust and has better performance in terms of approximation accuracy, simulation and processing time, and model complexity than the applied BP and SVM models. Reference [23] proposes a RVM model based on empirical model decomposition (EMD) to predict wind speed. The EMD algorithm is used to decompose wind speed signal in order to lessen the influence of uncertainty and nonlinearity on the model. This decomposition process results in a series of intrinsic mode functions (IMF) and RVM algorithm is applied to each IMF to construct a partial prediction model. The final prediction is obtained by superposition of all partial prediction models obtained for the IMFs. The authors in [23] claim that this method gives better forecasting results in terms of approximation accuracy that BP and RVM models alone. The reasons for applying combined fuzzy logic and neural network techniques for wind speed prediction in this work are discussed below.

A NEURO-FUZZY MODEL FOR WIND SPEED PREDICTION

Very often the real-world problems are extremely non-linear, or time-varying, or too com-

plex all together to be described with precise mathematical means. In addition, they might be

unfamiliar, uncertain, imprecise and/or vague. In all those cases, the conventional modelling techniques do not perform well and other approaches are needed in modelling such systems. Fuzzy logic and artificial neural networks are two scientific fields that provide unconventional modelling techniques proven to be very important and successful tools for modelling this kind of systems. The fuzzy logic models alone, called fuzzy inference systems (FIS) are very effective tools for modelling of nonlinear dynamic systems, but they have limitations. Their main disadvantages are lack of adaptability to different structural or behavioral changes in the modelled system and difficulties with extracting the necessary and accurate knowledge for building the rule base. On the other hand, the artificial neural networks (ANN) possess an inherent ability to adapt their parameters to changing conditions within or around the modelled system and automatically acquire the necessary knowledge. However, despite many of their advantages, ANN still have a number of weak points such as lack of interpretability, difficulties in choosing the number of hidden units, the problem of over-fitting etc. In order to overcome the disadvantages of the two modelling techniques, and to emphasize their advantages, Jang has proposed a hybrid neuro-fuzzy model, which acquires its knowledge from a given input-output data [24]. These models have been actively investigated and applied since [24–27]. The integration of learning capabilities of ANN and transparency of FIS results in a hybrid intelligent system capable of human-like reasoning, which learns its fuzzy if-then rules by some kind of learning algorithm from the field of ANN. However, the established classic and reliable ANN training methods have number of weak points, the existence of local minima solutions being one. Thus, further research has been conducted and different learning approaches have been attempted. A major breakthrough has been achieved by the advanced learning method for classification and regression called support vector machine (SVM), developed within the area of statistical learning theory [28]. Because of its excellent performance

in various applications, SVM has been widely acknowledged as one of the leading machine learning techniques. The main advantages of SVM learning method are that it possesses a very efficient mechanism for avoiding the over-fitting problem of ANNs, has proven itself to be a very good approximation tool, and is known to produce fairly sparse models. Yet, despite its widespread success, SVM still has some limitations. The main disadvantages of the SVM learning mechanism are that the number of required SVs increases proportionally with the size of the training data set, which in turn increases its computational complexity, the employed kernel functions must satisfy the Mercer's condition, and the SVM makes point predictions rather than generating predictive distributions. In order to obtain the posterior probability distribution of the output estimates in SVM, additional processing is required. To surpass the mentioned disadvantages of the SVM, Tipping has formulated the relevance vector machine (RVM) [29], which can be described as a probabilistic model with functional form equivalent to SVM. Compared to SVM, RVM has the following advantages: provides a full predictive distribution of the output since it is fully based on the statistical learning theory, performs generalization as well as the SVM, the applied kernel functions do not have to satisfy the Mercer's condition, and it typically employs significantly less kernel functions than the SVM, i.e. acquires significantly greater sparseness. To sum up, the RVM does not suffer from the SVM limitations and disadvantages, and its generalization performance and accuracy are comparable to the ones of the SVM with the advantage of employing fewer kernel functions than the SVM.

The neuro-fuzzy model for wind speed prediction presented here employs Tipping's RVM learning mechanism and is based on several excellent papers [29–31]. As most modern neuro-fuzzy systems, it is presented as a special multilayer feedforward neural network. The obtained results show a very good generalization feature of the applied modelling technique.

A BRIEF DESCRIPTION OF THE APPLIED FIS

The modelling of wind speed is based on the available input-output data $\{x_k, y_k\}; k = 1, 2, \dots, N$. The FIS implemented for modelling these data, as presented in [31], has the same structure as a Takagi-Sugeno (TS) fuzzy model [32]. The fuzzy IF-

THEN rules of this system have the following form:

$$\begin{aligned} \mathcal{R}^i: & \text{ IF } x_1 \text{ is } A_1^i \text{ and } x_2 \text{ is } A_2^i \text{ and } \dots \text{ and } x_M \text{ is } A_M^i \\ & \text{ THEN } f_i = a_{i1}x_1 + \dots + a_{iM}x_M + a_{i0}; i = 1, \dots, n \end{aligned} \quad (1)$$

and they represent the relationships between the input \mathbf{x}_k and output y_k of the modeled system. The variable x_j in (1) defines the j -th feature of the k -th input variable \mathbf{x}_k , and at the same time it is the j input to the fuzzy rules \mathfrak{R}^i , A_j^i are appropriate fuzzy sets, a_{ij} are consequent parameters, f_i is the output of the i -th fuzzy rule, n is the number of fuzzy rules and M is the dimension of the input data vectors ($i = 1, 2, \dots, n$; $j = 1, 2, \dots, M$).

The fuzzy sets A_j^i ($i = 1, 2, \dots, n$; $j = 1, 2, \dots, M$) in ordinary TS fuzzy models are represented by their membership functions, which can belong to any of the widely accepted conventional forms (trigonal, trapezoidal, bell shaped, etc.). Here in this paper, the fuzzy sets A_j^i are represented by kernel functions, which have the following Gaussian form:

$$K(x_j, x_{ij}^*) = \exp\left[-\frac{(x_j - x_{ij}^*)^2}{2\theta_{ij}^2}\right]; \quad \begin{matrix} i = 1, 2, \dots, n \\ j = 1, 2, \dots, M \end{matrix} \quad (2)$$

The variable x_{ij}^* in (2) is the center and θ_{ij} is the variance of the Gaussian kernel function $K(x_j, x_{ij}^*)$; ($i = 1, 2, \dots, n$; $j = 1, 2, \dots, M$). In general, the kernel functions $K(x_j, x_{ij}^*)$ can be of many different types, but the Gaussian kernel function

(2) has the advantage of allowing exact computation of its parameters, the center and variance. The introduced kernel functions (2), in fact constitute the Gaussian membership functions of the input variables x_j in the applied FIS.

The fuzzy IF-THEN rules of the presented neuro-fuzzy model, which number is automatically determined by the employed extended relevance vector learning algorithm, have the following specific form:

$$\begin{aligned} \mathfrak{R}_i : & \text{IF } x_1 \text{ is } K(x_1, x_{i1}^*) \text{ and } \dots \text{ and } x_M \text{ is } K(x_M, x_{iM}^*) \\ & \text{THEN } f_i = a_{i1}x_1 + \dots + a_{iM}x_M + a_{i0}; i = 1, \dots, n \end{aligned} \quad (3)$$

where each kernel function $K(x_j, x_{ij}^*)$ corresponds to one fuzzy set, $A_j^i = K(x_j, x_{ij}^*)$. The function $K(x_j, x_{ij}^*)$ in (3) represents the grade of membership of x_j with respect to the fuzzy set A_j^i , x_j is the j -th input to the fuzzy rules \mathfrak{R}^i , the kernel parameters x_{ij}^* are relevance vectors (RV), and together with the kernel parameters θ_{ij} represent parameters of the antecedent part of the fuzzy rules, f_i is the output variable of the i -th fuzzy rule, a_{ij} are parameters of the consequent part of the fuzzy rules, the number of fuzzy rules n equals the number of RVs and ($i = 1, 2, \dots, n$; $j = 1, 2, \dots, M$).

A BRIEF OVERVIEW OF RVM

A brief review of the most important features of RVM is presented next, and it is based on several distinguished papers [29, 30]. The input-output relationship of a given data set of input-output pairs $D = \{\mathbf{x}_k, y_k\}$; $k = 1, 2, \dots, N$, can be modeled by RVM as a weighted sum of N appropriately chosen basis functions $\phi_i(\mathbf{x}) = K(\mathbf{x}, \mathbf{x}_i)$:

$$f(\mathbf{x}) = f(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^N w_i K(\mathbf{x}, \mathbf{x}_i). \quad (4)$$

Equation (4) establishes the relationship between the scalar outputs $\{y_k\}$; $k = 1, 2, \dots, N$ and the input vectors $\mathbf{x} = \{\mathbf{x}_k\}$; $k = 1, 2, \dots, N$; the model parameters w_i in (4) are called as weights, and the kernel functions $K(\mathbf{x}, \mathbf{x}_i)$ define one basis

function for each example in the training data set D . Learning the function $f(\mathbf{x})$ means learning its parameters, i.e. the weights w_i , and the fact that the model (4) is linear in the parameters makes the process of its learning easier.

The modelling process is done in a Bayesian probabilistic framework, because of the presence of noise and uncertainty in data in real world situations. So, it can be said that RVM is a Bayesian approach to efficient estimation of the model (4) parameters, i.e. the elements of the parameter vector $\mathbf{w} = [w_1 \ w_2 \ \dots \ w_N]^T$. It has the same functional form as SVM, but it is not to be concluded that RVM is a Bayesian version of SVM; it is rather an independent method of its own.

Assuming that the outputs of the modelled system are noisy which can be expressed as follows

$$y_k = f(\mathbf{x}_k, \mathbf{w}) + \varepsilon_k; k = 1, 2, \dots, N, \quad (5)$$

an explicit probabilistic model over the output noise component ε_k is defined firstly. This model is a Gaussian distribution with zero mean and variance σ^2 ,

$$p(\varepsilon_k | \sigma^2) = N(0, \sigma^2). \quad (6)$$

Then, it follows that the probability distribution of the output y_k over the input data is

$$p(y_k | \mathbf{x}_k, \mathbf{w}, \sigma^2) = N[f(\mathbf{x}_k, \mathbf{w}), \sigma^2]. \quad (7)$$

The mean and the variance of this Gaussian distribution are $f(\mathbf{x}_k, \mathbf{w})$ and σ^2 , respectively. For independently generated examples of the training data set, the likelihood of the complete data set is given by

$$\begin{aligned} p(\mathbf{y} | \mathbf{w}, \sigma^2) &= p(\mathbf{y} | \mathbf{x}, \mathbf{w}, \sigma^2) = \\ &= \frac{\exp\left(-\frac{1}{2\sigma^2} \|\mathbf{y} - \Phi\mathbf{w}\|^2\right)}{\sqrt{(2\pi\sigma^2)^N}}. \end{aligned} \quad (8)$$

The matrix Φ in (8) is a design matrix of dimension $N \times N$, and N is the number of training data; \mathbf{w} is parameter vector of dimension N , and its elements are chosen to maximize the likelihood (8). The number of model parameters in (4) equals the number of training examples N , which yields severe over-fitting. This can be avoided, and the number of parameters in (4) can be limited, by introducing an explicit prior probability distribution over \mathbf{w} of the following form:

$$p(\mathbf{w}, \mathbf{a}) = \prod_{i=1}^N N(w_i | 0, \alpha_i^{-1}) = \prod_{i=1}^N \left(\frac{\alpha_i}{2\pi}\right)^{\frac{1}{2}} \exp\left(-\frac{\alpha_i}{2} w_i^2\right). \quad (9)$$

The imposed constraint on model parameters (9) is a zero-mean Gaussian distribution with a variance $1/\mathbf{a}$, $\mathbf{a} = [\alpha_1 \ \alpha_2 \ \dots \ \alpha_N]^T$ being the appropriate vector of N new parameters. The parameters α_i are called as hyperparameters, and by adjoining independently each hyperparameter with one of the weights, they moderate the strength of the prior by controlling the inverse variance of the associated weight. The adopted Gaussian distribu-

tion of the prior in (9) indicates that smaller weights are a priori more probable, and leads to smoother and less complex models, thus encoding the preference for smoother (simpler) models into the learning algorithm.

Although the model (4) already possesses too many parameters, the addition of N new parameters does not impose additional problem, since during the learning process based completely on the Bayesian probabilistic framework, many of the hyperparameters α_i become extremely large, which in turn leads to very small values for the appropriate weights w_i . As a consequence, the corresponding terms of the sum in (4) are eliminated as irrelevant. In this way, the number of parameters of the model (4) is drastically reduced, which leads to the desired sparseness. The RVM learning procedure is extremely effective in selecting only the relevant basis functions leading to good generalization.

In addition to the prior distribution over \mathbf{w} , hyperpriors over the hyperparameters \mathbf{a} and the noise variance σ^2 must be defined as well, which is done by assuming a Gamma distribution as follows.

$$\begin{aligned} p(\mathbf{a}) &= \prod_{i=1}^N \text{Gamma}(\alpha_i | a, b) \\ p(\beta) &= \text{Gamma}(\beta | c, d); \quad \beta = \sigma^{-2} \end{aligned} \quad (10)$$

The introduction of hierarchical priors over the parameters of the model \mathbf{w} , the hyperparameters \mathbf{a} , and the noise variance σ^2 , is a crucial feature of the relevance vector machine which ultimately results in desired sparse models. By fixing the parameters a, b, c , and d in (10) equal to zero, the hyperpriors become uniform over a logarithmic scale, which has a very pleasing consequence that the model output does not depend on the measurement units of the training output data.

Having introduced all the priors, including the priors over \mathbf{w} and the hyperpriors over \mathbf{a} and σ^2 , and following the Bayesian framework, for each given input data \mathbf{x}_k , a correct prediction of the corresponding output y_k is performed:

$$p(\hat{y}_k | \mathbf{y}) = \int p(\hat{y}_k | \mathbf{w}, \mathbf{a}, \sigma^2) \cdot p(\mathbf{w}, \mathbf{a}, \sigma^2 | \mathbf{y}) d\mathbf{w} d\mathbf{a} d\sigma^2, \quad (11)$$

where $p(\mathbf{w}, \mathbf{a}, \sigma^2 | \mathbf{y})$ is the posterior probability distribution over all unknown model parameters \mathbf{w} ,

$$p(\mathbf{w}, \mathbf{a}, \sigma^2 | \mathbf{y}) = p(\mathbf{w} | \mathbf{y}, \mathbf{a}, \sigma^2) p(\mathbf{a}, \sigma^2 | \mathbf{y}). \quad (12)$$

The weight posterior distribution

$$p(\mathbf{w} | \mathbf{y}, \mathbf{a}, \sigma^2)$$

can be estimated as

$$\begin{aligned} p(\mathbf{w} | \mathbf{y}, \mathbf{a}, \sigma^2) &= N(\mathbf{w} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \\ &= (2\pi)^{-\frac{N+1}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp \left[-\frac{(\mathbf{w} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{w} - \boldsymbol{\mu})}{2} \right], \end{aligned} \quad (13)$$

$$\boldsymbol{\Sigma} = (\sigma^{-2} \boldsymbol{\Phi}^T \boldsymbol{\Phi} + \mathbf{A})^{-1}, \mathbf{A} = \text{diag}(\alpha_1, \alpha_2, \dots, \alpha_N), \quad (14)$$

$$\boldsymbol{\mu} = \sigma^{-2} \boldsymbol{\Sigma} \boldsymbol{\Phi}^T \mathbf{y}, \quad (15)$$

where $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$ are the covariance and mean of this weight posterior distribution, respectively. It should be noted that $\mu_i \rightarrow 0$ whenever $\alpha_i \rightarrow \infty$.

Unfortunately, the posterior distribution

$$p(\mathbf{a}, \sigma^2 | \mathbf{y})$$

of the hyper parameters cannot be computed analytically and must be approximated, which leads to maximization of

$$p(\mathbf{a}, \sigma^2 | \mathbf{y}) \propto p(\mathbf{y} | \mathbf{a}, \sigma^2) p(\mathbf{a}) p(\sigma^2) \quad (16)$$

with respect to \mathbf{a} and σ^2 . In case of uniform hyperpriors only the term $p(\mathbf{y} | \mathbf{a}, \sigma^2)$ with covariance \mathbf{C} , known as marginal likelihood,

$$\begin{aligned} p(\mathbf{y} | \mathbf{a}, \sigma^2) &= (2\pi)^{-\frac{N}{2}} |\mathbf{C}|^{-\frac{1}{2}} \exp \left(-\frac{1}{2} \mathbf{y}^T \mathbf{C}^{-1} \mathbf{y} \right) \quad (17) \\ \mathbf{C} &= \sigma^2 \mathbf{I} + \boldsymbol{\Phi} \mathbf{A}^{-1} \boldsymbol{\Phi}^T \end{aligned}$$

needs to be maximized, which is named as type-II maximum likelihood method [33].

The optimal values of α_i and σ^2 that maximize the marginal likelihood (17), i.e. its logarithm with respect to the hyperparameters α_i , cannot be obtained in a closed form. Instead, they are computed iteratively [34]

$$\alpha_{i, \text{new}} = \frac{\gamma_i}{\mu_i^2}; \quad \gamma_i = 1 - \alpha_i \Sigma_{ii}, \quad \sigma_{\text{new}}^2 = \frac{\|\mathbf{y} - \boldsymbol{\Phi} \boldsymbol{\mu}\|^2}{N - \sum_i \gamma_i}. \quad (18)$$

The parameter μ_i in (18) is the i -th posterior mean weight from (15), Σ_{ii} is the i -th diagonal element of the posterior weight covariance in (14), computed with the current values of parameters \mathbf{a} and σ^2 . Each parameter $\gamma_i \in [0, 1]$ can be considered a measure of how well its corresponding parameter w_i is determined by the given training data. For large values of α_i , $\Sigma_{ii} \approx \alpha_i^{-1}$ and it follows that $\gamma_i \approx 0$. On the other hand, when α_i is small, $\gamma_i \approx 1$ and w_i fits the data. It should be noted that N in the denominator of (18) refers to the number of data examples and not the number of basis functions.

The RVM learning algorithm is of an iterative type. The parameters α_i and σ^2 are repeatedly estimated every iteration step until the desired convergence criterion is achieved. At the same time, at each step of iteration the newly calculated values for α_i and σ^2 are used for updating the posterior statistics $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$. During this iteration process, many of the hyperparameters become very large approaching infinity, which indicates that the appropriate posterior distributions $p(w_i | \mathbf{y}, \mathbf{a}, \sigma^2)$ become very large at zero. This further means that the adjacent weights w_i are zero with a posteriori certainty. The zero valued weights result in pruning the corresponding basis functions in (4), thus reducing considerably the number of model parameters. The vectors from the training data set associated with the remaining nonzero weights are called the relevance vectors (RV).

THE STRUCTURE OF THE FIS WITH EXTENDED RVM

The structure of the neuro-fuzzy system is shown in Figure 1. It represents a neural network with six different layers. The first layer is called the input layer. It consists of nodes that represent

the input variables to the model. There is one node in this layer for each input variable. Thus, the input layer has a total of M nodes, M being the number of elements in the training input vector

$\mathbf{x}_k = (x_{k1}, x_{k2}, \dots, x_{kM})$. This layer has the sole role of transmitting the upcoming input data to the second layer. It does not perform any operations over the training input data.

The second layer is the fuzzification layer, since it performs fuzzification over the training input data by projecting the input space into a high-dimension feature space. This nonlinear projection is defined by the chosen kernel functions. Each node in this layer has exactly M inputs, M being the dimension of the input vector $\mathbf{x}_k = (x_{k1}, x_{k2}, \dots, x_{kM})$, i.e. the number of nodes in the input layer. The second layer consists of n nodes that represent the adequate kernel functions. These kernel functions are not required to satisfy the Mercer's condition as in the case of SVM learning mechanism. Therefore they can have a different shape: triangular, trapezoidal, bell, Gaussian, polynomial, Fourier series etc. The neuro-fuzzy model in Figure 1 uses the Gaussian kernel functions defined by (2). The choice of Gaussian kernel functions is very convenient because their parameters can be learned easily and computed precisely.

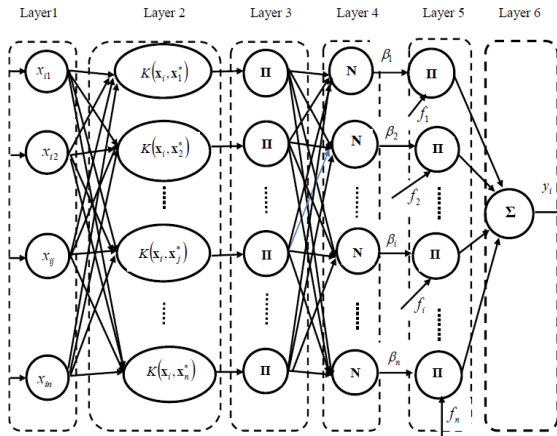


Fig. 1. The structure of the neuro-fuzzy model used for modelling the wind speed

From the fuzzy modelling perspective, the terms in (2) can be interpreted as follows: x_j is the j -th input to the fuzzy model, $K(x_j, x_{ij}^*)$ is the membership function $\mu_{ij} = \mu_{A_j^i}(x_j)$ of the j -th fuzzy input x_j with respect to the i -th fuzzy rule, the parameters x_{ij}^* and θ_{ij} of $K(x_j, x_{ij}^*)$ are the premise parameters of the corresponding fuzzy

rule, M is the number of fuzzy inputs to the neuro-fuzzy model, and the number n of kernel functions $K(x_j, x_{ij}^*)$ is the number of fuzzy rules, i.e. the number of nodes in the second layer. Because of the Gaussian shape of the selected kernel functions, the membership functions of the antecedent part of the fuzzy rules are Gaussian membership functions. From the RVM prospective, the center x_{ij}^* of $K(x_j, x_{ij}^*)$ is a relevance vector, the variance θ_{ij} , i.e. the width of the Gaussian kernel, is a kernel parameter, and n is the number of relevance vectors.

The third layer can be called as the rule layer, since a node in this layer generates the IF part of each fuzzy rule. The nodes in this layer can be called as the rule nodes, accordingly. This layer has n nodes, one for each fuzzy rule, and they compute the firing strength of the associated fuzzy rules. For each node the following T-norm operator is used:

$$K(\mathbf{x}, \mathbf{x}_i^*) = \prod_{j=1}^M K(x_j, x_{ij}^*); \quad i = 1, 2, \dots, n. \quad (19)$$

The vector $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iM})$ in (19) represents the i -th input vector to the model of dimension M , and $\mathbf{x}_i^* = (x_{i1}^*, x_{i2}^*, \dots, x_{iM}^*)$ is the RV of the i -th input vector. Instead of the product of membership functions in (19), any other T-norm operator could be used to perform the fuzzy AND operation. The first three layers in Figure 1 belong to the antecedent part of the FIS. The next layers belong to the consequent part of the FIS.

The fourth layer is the normalization layer. It consists of n nodes and each node performs normalization of the firing strength of the associated fuzzy rule. This normalization is done with respect to the sum of the firing strengths of all the fuzzy rules, and the output of each node in this layer is the normalized firing strength, i.e. weight β_i of the corresponding fuzzy rule, computed as the following ratio:

$$\beta_i = \frac{K(\mathbf{x}, \mathbf{x}_i^*)}{\sum_{j=1}^n K(\mathbf{x}, \mathbf{x}_j^*)}; \quad i = 1, 2, \dots, n. \quad (20)$$

Each node i in the fifth layer calculates the product of the normalized weight β_i for the i -th

rule and the local output variable f_i of the fuzzy system. The output variables v_i of the nodes in this layer are:

$$v_i = \beta_i f_i = \beta_i (a_{i1}x_{i1} + \dots + a_{iM}x_{iM} + a_{i0}). \quad (21)$$

The parameters $(a_{i0}, a_{i1}, \dots, a_{iM})$ in (21) can be called as consequent parameters, since they represent the parameters of the consequent part of the correspondent fuzzy rules.

The sixth and the last layer is the output layer. The single node in this layer computes the overall output $f(\mathbf{x})$ of the neuro-fuzzy model as the sum of all incoming signals,

$$f(\mathbf{x}) = \sum_{i=1}^n \beta_i f_i. \quad (22)$$

MODELLING AND SIMULATION RESULTS

The neuro-fuzzy model for wind speed prediction was built upon the available meteorological data for Mauna Loa (MOA), Hawaii, US, for year 2015, available at [35]. The measured input-output data points were randomly permuted and divided into two sets – training and evaluation data. The proposed neuro-fuzzy model for wind speed prediction was built and evaluated on 10 such randomizations between training and test data containing $\frac{3}{4}$ and $\frac{1}{4}$ data points, accordingly. After training with the training data set, 48 relevance vectors were generated, thus yielding a neuro-fuzzy model of wind speed with 48 fuzzy rules. The learning algorithm of the FIS with extended RVM according to [31] is shown in Figure. 2.

The output of the neuro-fuzzy model for wind speed forecasting with the obtained relevance vectors is shown in Figure. 3, compared to the actual measured output. It can be easily seen that the model output practically coincides with the measured output.

In order to evaluate the performance of the built neuro-fuzzy model for wind speed prediction, the root mean square error (RMSE) of the model output y_{model} is calculated

$$PI = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_{\text{real}}^i - y_{\text{model}}^i)^2}. \quad (23)$$

The structure of this adaptive neural network is not unique. For example, last two layers can be easily combined to form one defuzzification layer, which computes the overall output $f(\mathbf{x})$ of the neuro-fuzzy model using the center of gravity defuzzification method. Similarly, the third and the fourth layers can be combined together as one layer, to obtain an equivalent five layers neural network, as in [31].

The system in Figure 1 performs system optimization and generalization simultaneously. The number of fuzzy rules and the parameters of the membership functions are generated automatically by the extended relevance vector learning machine algorithm [31]. The parameters of the kernel functions are adjusted by the gradient ascent method (GAM) [31]. The coefficients in the consequent part of the fuzzy rules are determined by the least square method (LSE) [31].

It is compared to several unconventional models based on Sugeno-Yasukawa fuzzy identification [36], and extreme learning machine ELM [37], and the comparison results are shown in Table 3.

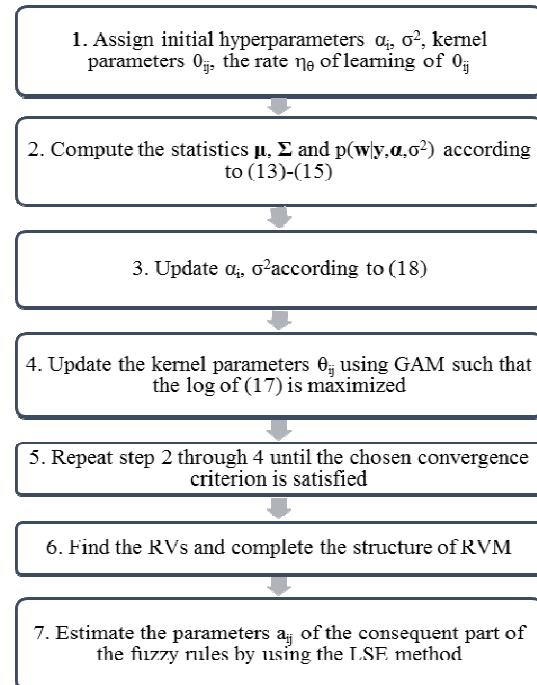


Fig. 2. The learning algorithm of the neuro-fuzzy model with extended RVM

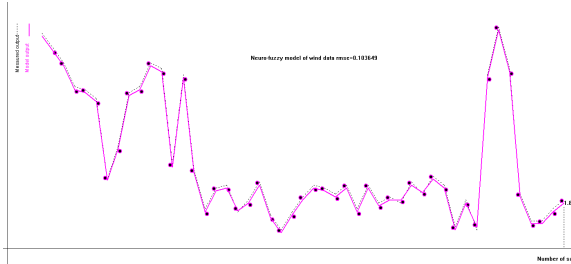


Fig. 3. The output of the neuro-fuzzy model for wind speed prediction with relevance vectors, compared to the actual measured output.

Table 3

Comparison of wind speed prediction models obtained by different modelling techniques

Model	RMSE
Position type fuzzy model	0.40438
Position-gradient type fuzzy model	0.33537
Neuro-fuzzy model based on extended RVM	0.103649
NN model based on ELM	0.162942

CONCLUSION

The simulation results proved the built FIS very effective in modelling the wind speed. The main advantages of the RVM learning algorithm are: the ability to provide accurate prediction model with fewer basis functions, automatic estimation of “nuisance” parameters, and the facility to utilize arbitrary basis functions.

Acknowledgment; The meteorological data used in this research are part of the meteorology measurements from NOAA/ESRL/GMD, where NOAA stands for National Oceanic and Atmospheric Administration, ESRL stands for Earth System Research Laboratory and GMD stands for Global monitoring division. They can be accessed at National Renewable Energy Laboratory NREL web-site [35], which is operated for the US Department of Energy by the Alliance for Sustainable Energy LLC.

REFERENCES

- [1] R. Ehrlich: *Renewable Energy: A first course*, 1st Edition, CRC Press, 2013.
- [2] R. P. Walker, A. Swift: *Wind Energy Essentials: Societal, Economic, and Environmental Impacts*, 1st Edition, Wiley, 2015.
- [3] L. R. Brown, E. Adams, J. Larsen, J. M. Roney: *The Great Transition: Shifting from Fossil Fuels to Solar and Wind Energy*, 1st Edition, W. W. Norton & Company, 2015.
- [4] Global Wind Energy Council: *Global wind statistics 2015*, GWEC, 2015. Available at: http://www.gwec.net/wp-content/uploads/vip/GWEC-PRstats-2015_LR_corrected.pdf.
- [5] R. Gasch, J. Twele: *Wind Power Plants: Fundamentals, Design, Construction and Operation*, 2nd edition, Springer, 2012.
- [6] Y. Tamura, K. Suda, A. Sasaki, Y. Iwatani, K. Fujii, R. Ishibashi, K. Hibi: Simultaneous measurements of wind speed profiles at two sites using Doppler sodars, *Journal of Wind Engineering and Industrial Aerodynamics*, **89**, 3–4, 325–335 (March 2001).
- [7] S. Soisuvann, Z. Jelenak, P. S. S. O. Cheng, Q. Zhu: CMOD5.H.Q. – A high wind geophysical model function for c-band vertically polarized satellite scatterometer measurements, *IEEE Transactions on Geoscience and Remotesensing*, **51**, 6, 3741–3760 (June 2013).
- [8] S. Yang, E. McKeogh: LIDAR and SODAR measurements of wind speed and direction in upland terrain for wind energy purposes, *Remote Sensing*, **3**, 9, 1871–1901 (2011).
- [9] A. Kusiak, W. Li: Estimation of wind speed: a data-driven approach, *Journal of Wind Engineering and Industrial Aerodynamics*, **98**, 10–11, 559–567 (Oct.–Nov. 2010).
- [10] M. Mohandes, S. Rehman, S. M. Rahman: Estimation of wind speed profile using adaptive neuro-fuzzy inference system (ANFIS), *Applied Energy*, **88**, 11, 4024–4032 (Nov. 2011).
- [11] J. L. Torres, A. García, M. De Blas, A. De Francisco: Forecast of hourly average wind speed with ARMA models in Navarre (Spain), *Solar Energy*, **79**, 1, 65–77 (July 2005).
- [12] R. G. Kavasseri, K. Seetharaman: Day-ahead wind speed forecasting using f-ARIMA models, *IEEE Transactions on Renewable Energy*, **34**, 5, 1388–1393 (May 2009).
- [13] T. M. H. El-Fouly, E. F. El-Saadany: Grey predictor for hourly wind speed and power forecasting, *IEEE Transactions on Power Systems*, **21**, 3, 1450–1452 (August 2006).
- [14] I. G. Damousis, P. Dokopoulos: A fuzzy expert system for the forecasting of wind speed and power generation in wind farms, In: *Wind Energy Conversion Systems: Technology and Trends*, S. M. Mayeen ed., London: Springer, 2012, pp. 197–226.
- [15] T. G. Barbounis, J. B. Theocharis, M. C. Alexadis, P. S. Dokopoulos: Long term wind speed and powercasting using local recurrent neural network models, *IEEE Transactions on Energy Conversion*, **21**, 1, 273–284 (March 2006).
- [16] M. A. Mohandes, T. O. Halawani, S. Rehman, A. A. Hussain: Support vector machine for wind speed prediction, *Renewable Energy*, **29**, 6, 939–947 (May 2004).
- [17] A. U. Haque, P. Mandal, J. Meng, M. E. Kaye, L. Chang: A new strategy for wind speed forecasting using hybrid

- intelligent models, 25th *IEEE Canadian Conference on Electrical and Computer Engineering, CCECE*, 2012, pp. 1–4.
- [18] S. S. Soman, H. Zareipour, O. Malik, P. Mandal: A review of wind power and wind speed forecasting methods with different time horizons, *Nort American Power*, **2**, 5, 8–16 (2010).
- [19] M. Bhaskar, A. Jain, N. V. Srinath: Wind speed forecasting: Present Status, *International Conference on Power System Technology (POWERCON)*, 24–28 Oct. 2010, pp. 1–6.
- [20] S. M. Lawan, W. A. W. Z. Abidin, W. Y. Chai, A. Baharun, T. Maasri: Different models of wind speed prediction: A comprehensive review, *International Journal of Scientific and Engineering Research*, **5**, 1, 1760–1768 (2014).
- [21] M. Lei, L. Shiyan, J. Chuanwen, L. Hongling, Z. Yan: A review on the forecasting of wind speed and generated power, *Renewable and Sustainable Energy Reviews*, **13**, 4, 915–920 (May 2009).
- [22] G. Sun, Y. Chen, Z. Wei, X. Li, K. W. Cheung: Day-ahead wind speed forecasting using relevance vector machine, *Journal of Applied Mathematics*, Volume **2014**, Article ID 437592, available on-line at: <http://dx.doi.org/10.1155/2014/437592>.
- [23] G. Yang, Z. Hu, X. Liu: A novel strategy for wind speed prediction in Wind farm, *Telkomnika*, **11**, 12, 7007–7013 (Dec. 2013). (2004).
- [24] J.-S. R. Jang: ANFIS: Adaptive-network-based fuzzy inference systems, *IEEE Trans. Sys. Man. Cybern.*, **23**, pp. 665–685 (1993).
- [25] C. T. Lin: A Neural fuzzy control systems with structure and parameter learning, *Fuzzy Sets and Systems*, **70**, pp. 183–212 (1995).
- [26] C. T. Lin, C. S. G. Lee,: *Neural fuzzy systems: A neural fuzzy synergism to intelligent systems*, Prentice-Hall, Englewood Cliffs, 1996.
- [27] C. F. Yang, C. T. Lin: An on-line self-constructing neural fuzzy inference network and its applications, *IEEE Trans. Fuzzy Syst.*, **6**, pp. 12–32 (1998).
- [28] V. N. Vapnik: *Statistical Learning Theory*, Wiley, N. Y., 1998.
- [29] M. E. Tipping: The relevance vector machine. In: S. A. Sola, T. K. Leen and K.-R. Muller, editors, *Advances in Neural Processing Systems* **12**, MIT Press, pp. 652–658 (2000).
- [30] M. E. Tipping: Sparse Bayesian learning and the relevance vector machine, *J. Mach. Learn. Res.*, **1**, pp. 211–244 (2001).
- [31] J. Kim, Y. Suga, S. Won: A new approach to fuzzy modelling of nonlinear dynamic systems with noise: relevance vector learning mechanism, *IEEE Trans. on Fuzzy Systems*, **14**, pp. 222–231 (2006).
- [32] T. Takagi, M. Sugeno: Fuzzy identification of systems and its applications to modelling and control, *IEEE Trans. Syst. Man. Cybern.* **15**, pp. 116–132 (1985).
- [33] J. O. Berger: *Statistical Decision Theory and Bayesian Analysis*, Springer, 2nd edition, 1985.
- [34] D. J. C. MacKey: Bayesian interpolation, *Neural Computation*, **4**, pp. 415–445 (1992).
- [35] http://www.nrel.gov/gis/data_wind.html
- [36] M. Sugeno, T. Yasukawa: A fuzzy-logic-based approach to qualitative modelling, *IEEE Trans. on Fuzzy Syst.*, **1**, pp. 7–33 (1993).
- [37] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: A new learning scheme of feedforward neural networks, *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN2004)*, 25–29 July, Budapest, Hungary, 2004, pp. 985–990.

