

# Short-term wind speed predictions with machine learning techniques

M. A. Ghorbani<sup>1</sup> · R. Khatibi<sup>2</sup> · M. H. FazeliFard<sup>1</sup> · L. Naghipour<sup>3</sup> · O. Makarynsky<sup>4</sup>

Received: 22 September 2013 / Accepted: 3 August 2015 / Published online: 14 September 2015  
© Springer-Verlag Wien 2015

**Abstract** Hourly wind speed forecasting is presented by a modeling study with possible applications to practical problems including farming wind energy, aircraft safety and airport operations. Modeling techniques employed in this paper for such short-term predictions are based on the machine learning techniques of artificial neural networks (ANNs) and genetic expression programming (GEP). Recorded values of wind speed were used, which comprised 8 years of collected data at the Kersey site, Colorado, USA. The January data over the first 7 years (2005–2011) were used for model training; and the January data for 2012 were used for model testing. A number of model structures were investigated for the validation of the robustness of these two techniques. The prediction results were compared with those of a multiple linear regression (MLR) method and

with the Persistence method developed for the data. The model performances were evaluated using the correlation coefficient, root mean square error, Nash–Sutcliffe efficiency coefficient and Akaike information criterion. The results indicate that forecasting wind speed is feasible using past records of wind speed alone, but the maximum lead time for the data was found to be 14 h. The results show that different techniques would lead to different results, where the choice between them is not easy. Thus, decision making has to be informed of these modeling results and decisions should be arrived at on the basis of an understanding of inherent uncertainties. The results show that both GEP and ANN are equally credible selections and even MLR should not be dismissed, as it has its uses.

---

Responsible Editor: C. Simmer.

---

✉ M. A. Ghorbani  
Ghorbani@tabrizu.ac.ir; m.ali\_ghorbani@ymail.com

R. Khatibi  
rahman.khatibi@gmail.com

L. Naghipour  
Naghipour.L@tabrizu.ac.ir

O. Makarynsky  
makarynsky@live.com

<sup>1</sup> Department of Water Engineering, Faculty of Agriculture, University of Tabriz, Tabriz, Iran

<sup>2</sup> GTEV-ReX - Research and Mathematical Modelling, Swindon, UK

<sup>3</sup> Department of Water Engineering, Faculty of Civil Engineering, University of Tabriz, Tabriz, Iran

<sup>4</sup> METOcean Dynamic Solutions, 19 Pelion Street, Bardon 4065, Australia

## 1 Introduction

Autoregressive types of regression models are used to formulate forecasting models by using the information contained within their recorded values. The study employs wind speed time series at the hourly interval and investigates four techniques: two machine learning techniques (artificial neural networks and genetic expression programming) and two techniques for comparison: a multiple linear regression and the simple Persistence methods. The study has potential applications to energy sources from the wind and, as stated by Burton et al. (2001) and Li and Shi (2010), this source of energy is particularly attractive for being clean, renewable, economically competitive and environmentally friendly. The wind energy systems depend on wind speed and a host of factors as discussed by Tandjaoui et al. (2013), including (i) low production capacity when deployed in a sheltered area and (ii)

generated noise. The focus of this paper is on studying the forecasting problems of wind speed.

Machine learning techniques have been used extensively for wind speed predictions over the years. Some recent examples include applications of artificial neural networks (Bilgili and Sahin 2010; Li and Shi 2010; Khatibi et al. 2014), Fuzzy Logic (Barbounis and Theocharis 2007; Damousis et al. 2004; Kariniotakis et al. 1996a; Wang et al. 2004), Genetic Programming (Ghorbani et al. 2010; Guven et al. 2008; Kalra and Deo 2007; Kalra et al. 2008; Khatibi et al. 2011; Ustoorikar and Deo 2008), as well as radial basis function (Beyer et al. 1994), recurrent neural networks (Kariniotakis et al. 1996b; More and Deo 2003) and support vector machines (Ji et al. 2007; Mohandes et al. 2004). Notably, artificial neural networks and genetic programming techniques are among the most frequently used; they will be described in more detail here.

Artificial neural networks (ANNs) are parallel information processing system and emulate the working processes in the brain. ANNs consist of a set of neurons or nodes arranged in layers and, in the case that weighted inputs are used, these nodes provide suitable inputs by conversion functions (Kisi 2005). Each neuron in a layer is connected to all the neurons of the next layer, but without any interconnection among the neurons in the same layer. Neural networks can learn from past data, recognize hidden patterns or relationships in historical observations and use them to forecast future values and thus system behavior.

The genetic programming (GP) methods, first proposed by Koza (1992), are wide ranging and similar to genetic algorithms (GA) (Goldberg 1989). GP techniques are robust applications of optimization algorithms and represent one way of mimicking natural selection. These techniques derive a set of mathematical expressions to describe the relationship between the predictant and dependent variables using such operators as mutation, recombination (or crossover) and evolution. These are operated in a population evolving in generations through a definition of fitness and selection criteria. Applications of GP suit a wide range of problems and in particular to cases where: (i) the interrelationships among the relevant variables are poorly understood or suspected to be wrong; (ii) conventional mathematical analyses are constrained by restrictive assumptions, but approximate solutions are acceptable (Banzhaf et al. 1998). The genetic expression programming (GEP), an extension of genetic programming (GP), is used as an alternative approach to prediction of wind speed time series in this study. The fundamental difference between the GEP and GP algorithms resides in the nature of the individuals. In GP, the individuals are non-linear entities of different sizes and shapes (parse trees). In GEP, the individuals are encoded as linear strings of fixed length (the genome or chromosomes), which are afterward expressed

as non-linear entities of different sizes and shapes (simple diagrammatic representations or expression trees). Also, GEP is faster than GP, by two to four orders of magnitude. Due to its simple and efficient features, GEP is also a popular method for evolution modeling and widely used (Ferreira 2006).

Statistical regression models are simple and straightforward and usually the first choice to establish the baseline; conversely, their applications often go back many decades. In addition to reporting multiple linear regression (MLR) method, the Persistence method is also reported here as a possible alternative, which has no mathematical sophistication.

The present work describes the development and training of ANNs and GEP models for the purpose of estimating hourly wind speed, and the results are compared with the MLR and the Persistence Method. The model performances have been estimated by using the correlation coefficient (CC), Nash–Sutcliffe efficiency coefficient ( $E$ ), root mean square error (RMSE) and Akaike information criterion (AIC). The study employs the time series of wind speed values obtained at Kersey in Colorado, USA.

## 2 Materials and methods

### 2.1 Artificial neural networks (ANNs)

Neural networks are inspired by the studies of the brain and nervous systems in biological organisms. They have the capability for self-learning and automatic abstraction. Their developments go back to McCullon and Pitts (1943) as the designers of the first artificial neural networks (ANNs). Over the years, ANNs have been recognized to provide an important alternative to the traditional methods of data analysis and modeling.

The fundamental processing element of a neural network is a neuron. Each neuron computes a weighted sum of its  $p$  input signals,  $x_i$ , for  $i = 0, 1, 2, \dots, P$  hidden layers,  $w_{ij}$ , and then applies a nonlinear activation function to produce an output signals  $u_j$ . The model of a neuron is

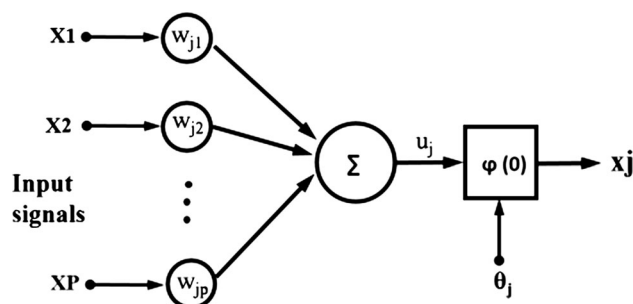


Fig. 1 Nonlinear model of a neuron

shown in Fig. 1. A neuron  $j$  may be mathematically described with the following pair of equations:

$$u_j = \sum_{i=0}^p w_{ij}x_i \tag{1}$$

and

$$x_j = \phi(u_j - \theta_j). \tag{2}$$

The use of threshold  $\theta_j$  has the effect of applying an affine transformation to the output of the linear combiner in the model of Fig. 1. A threshold transfer function is sometimes used to quantify the output of a neuron in the output layer. In particular, depending on whether the threshold  $\theta_j$  is positive or negative, the relationship between the effective internal activity level or activation potential  $v_j = (u_j - \theta_j)$  of neuron  $j$  and the linear combiner output  $u_j$  is modified in the manner illustrated in Fig. 2. Affine transformation method is applied for the coordinate transformation between two reference systems. A coordinate transformation model can be optimized so that it is easy to perform and gives the highest accuracy. If coordinates from two coordinate systems are available for some common points, those transformation parameters can be estimated (Haykin 1999; Melesse and Hanley 2005).

The sigmoid logistic nonlinear function is described by the following equation (Bilgili et al. 2007):

$$\phi(x) = \frac{1}{1 + e^{-x}}. \tag{3}$$

The type of ANN used in this study is a feedforward multilayer perceptron (MLP) which is the most commonly used ANN in hydro-meteorological applications. A set of neurons or nodes may be arranged as layers. The structure

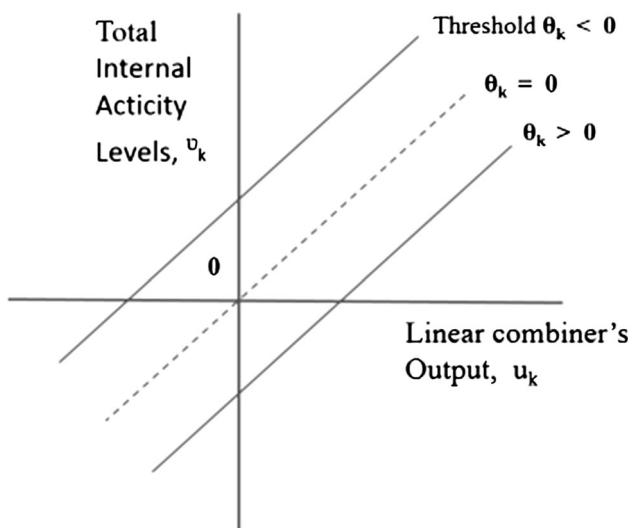


Fig. 2 Transformation produced by the presence of a threshold

of a three-layer MLP is shown in Fig. 3. It consists of three layers: an input layer, a hidden layer and an output layer. The number of neurons in the input and output layers is defined based on the number of input and output variables of the system under investigation, respectively. However, the number of neurons in the hidden layer(s) is usually determined via a trial-and-error procedure. As seen from the figure, the neurons of each layer are connected to the neurons of the next layer by weights. To obtain optimal values of these connection weights, ANNs must be trained.

The neural networks technique for wind speed prediction was first investigated by Mohandes et al. (1998) who compared its performance with an autoregressive model. This included investigations of the statistical characteristics of mean monthly and daily wind speed in Jeddah, Saudi Arabia. The autocorrelation coefficients and correlogram were employed to investigate the real diurnal variation of mean wind speed. They showed that for prediction of wind speed time series, feedforward back propagation (FFBP) artificial neural networks had higher accuracy than the used regression model, and the best network based on the RMSE over the training data was one with 24 hidden units.

The study by Cadenas and Rivera (2009) investigated hourly wind speed time series during about 1 month representative of three sites in Mexico, and diverse configurations of ANN were generated and compared through error measures. First, a model with three layers and seven neurons was chosen, according to the recommendations of various authors. The results were not sufficiently satisfactory, so other three models were developed, consisting of three layers and six neurons, two layers and four neurons, and two layers and three neurons. The simplest model of two layers, with two input neurons and one output neuron, performed the best in the short-term wind speed forecasting and showed a good accuracy to be used for the energy supply.

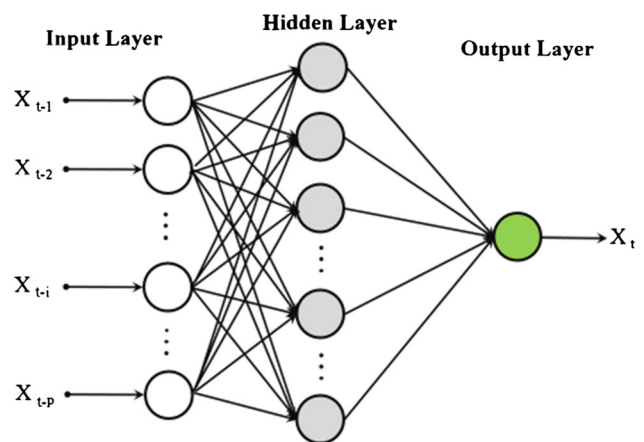


Fig. 3 Simple configuration of a multilayer perceptron neural network

The authors indicated that the efficiency of the FFBP may be increased by using the conjugate gradient learning.

A comprehensive comparison study on the application of different ANN, namely, adaptive linear element, back propagation and radial basis function in hourly wind speed forecasts was presented by Li and Shi (2010). The results of that study showed that no single neural network outperformed others when overall evaluation metrics were considered.

## 2.2 Genetic expression programming

Genetic expression programming (GEP) proposed by Candida Ferreira in 1999 is an evolutionary technique, which combines features of its predecessors, genetic algorithm (GA) and genetic programming (GP). The main difference between the three algorithms resides in coding of individuals, as defined below. In GA, individuals are represented by symbolic strings of a fixed length, the so-called chromosomes. Individuals in GP are nonlinear entities of different sizes and shapes, the so-called parse trees. GEP individuals are encoded as symbolic strings of a fixed length (chromosomes), which are then expressed as nonlinear entities of different size and shape, the so-called expression trees (ET).

GA and GP suffer from two main limitations. They may be easy to manipulate genetically but they lose in functional complexity. On the other hand, when their functional complexity is high, it is extremely difficult to reproduce without some modification. Furthermore, use of genetic operators in GP is very limited and operators act directly on parse trees. GEP resolves those constraints leading to possible improvements in speed and accuracy (Ferreira 2002). The application of GEP to practical problems is diversifying and in recent years and these include: Guven et al. 2008; Ustoorikar and Deo 2008; Kisi and Guven 2010; Zakaria et al. 2010; Khatibi et al. 2014. GEP automatically generates equations that describe cause and effect relationships in the data; it is significantly slower in developing models and generates relatively simple equations describing the relationships that can be interpreted directly.

In GEP, the chromosome consists of a linear, symbolic string of a fixed length. One chromosome can contain one or more genes, each encoding a sub-expression tree. Despite the fact that the length of chromosomes is fixed, it is still possible to code expression trees of different sizes and shapes. Structural organization of genes in a head and a tail always guarantees production of valid programs (Ferreira 2001a). GEP identifies an appropriate relationship for any given time series by two components: (i) a set of functions and their parameters (referred to as the terminal

set), which emulates the role of proteins or chromosomes in biological systems; and (ii) a parse tree, which is a functional set of basic operators such as  $\{+, -, *, /, \wedge, \sqrt{\quad}, \log, a \log, \sin, a \sin, \exp, \dots\}$ .

The process of GEP begins with the random generation of the chromosomes of the initial population. Then the chromosomes are expressed and the fitness of each individual is evaluated. The individuals are then selected according to fitness to reproduce with modifications, leaving progeny with new traits. The individuals of this new generation are, in their turn, subjected to the same developmental process: expression of the genomes, confrontation of the selection environment and reproduction with modification. The process is repeated for a certain number of generations, or until a solution has been found (Ferreira 2001b). The fundamental steps of the GEP are schematically represented in Fig. 4.

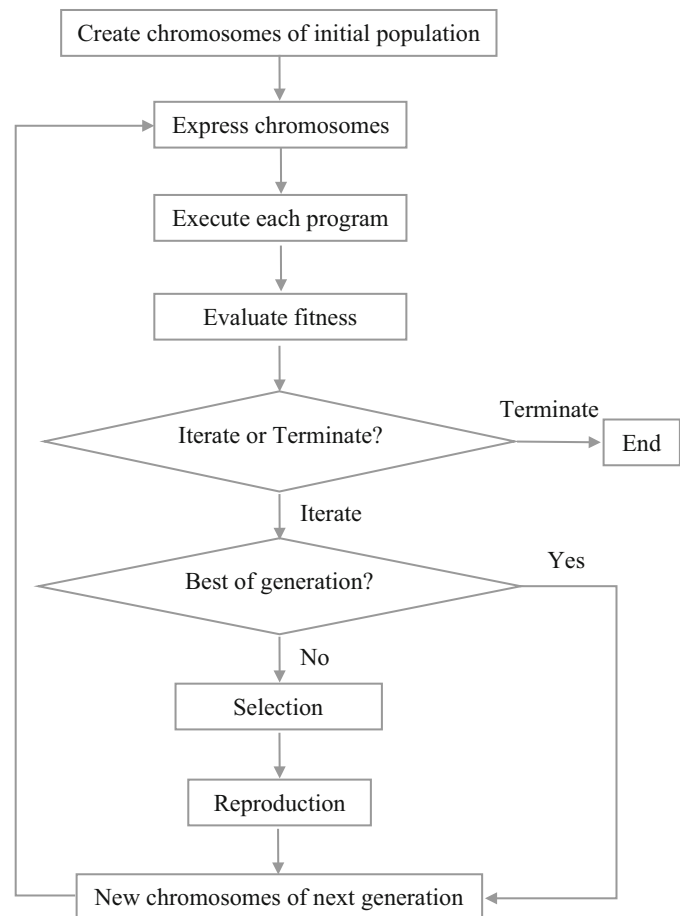
Flores et al. (2005) explored the applicability of genetic programming to wind speed prediction. Unlike other artificial intelligent techniques, GP provided closed-form models for the time series under analysis. They used a form of GEP for the implementation of the modeling tools, and the techniques were applied to the time series formed by monthly averages of the wind speed in the Isthmus of Tehuantepec, Mexico. GP has shown to be a good alternative to provide models for wind speed prediction.

The method of least squares support vector machine (LS-SVM) for short-term wind speed prediction was put forward by Xiaojuan et al. (2009), and the influence of parameters selection of LS-SVM on prediction accuracy was analyzed. The genetic algorithm was adopted to realize parameters optimization of LS-SVM and establish short-term wind speed prediction model of LS-SVM based on genetic algorithm. It was shown that the method proposed in this paper can quickly and effectively carry out short-term wind speed prediction by simulation example.

Huang et al. (2011) used a real-valued genetic algorithm (RGA)-based least-squared support vector machine (LS-SVM) to precisely predict the short-term regional wind speed. A dataset including the time, temperature, humidity and the average regional wind speed being measured in a randomly selected date from a wind farm being located in Penghu, Taiwan, was selected for verifying the forecast efficiency of the proposed RGA-based LS-SVM.

Numerical weather prediction models were used by Arellano et al. (2012) to produce wind speed forecasts at a high spatial resolution. The integration of the weather research and forecasting-advanced research WRF (WRF-ARW) mesoscale model with four different downscaling approaches was presented. WRFARW forecasts and observations at three different sites of the state of Illinois in the USA were analyzed before and after applying the

**Fig. 4** The flowchart of a genetic expression algorithm



downscaling techniques. Three of the proposed methods needed a predefined model to be applied. The fourth approach, based on genetic programming (GP), implicitly found the optimal model to downscale WRF forecasts, so no previous assumptions about the model had to be made. The results obtained demonstrated that GP was able to successfully downscale the wind speed predictions, reducing significantly the inherent error of the considered numerical models.

### 2.3 Multiple linear regression (MLR)

The multiple linear regression analysis is a widely used technique for expressing the dependence of a response variable on several predictants. It fits a linear combination of the components of a multiple signals  $x_i$  to a single output signal  $y$ , as defined by (4):

$$y = a_0 + \sum_{i=1}^n a_i x_i, \quad (4)$$

where  $a_i$  values are called regression coefficients, which are estimated by using the least square or any other similar method. In this study, the coefficients of regression were

determined using the least square method. The stepwise regression is applied as a robust method for the selection of best subset models and is based on adding or deleting the variable/variables with the greatest impact on the residual sum of squares (Ghorbani et al. 2012).

### 2.4 Persistence method

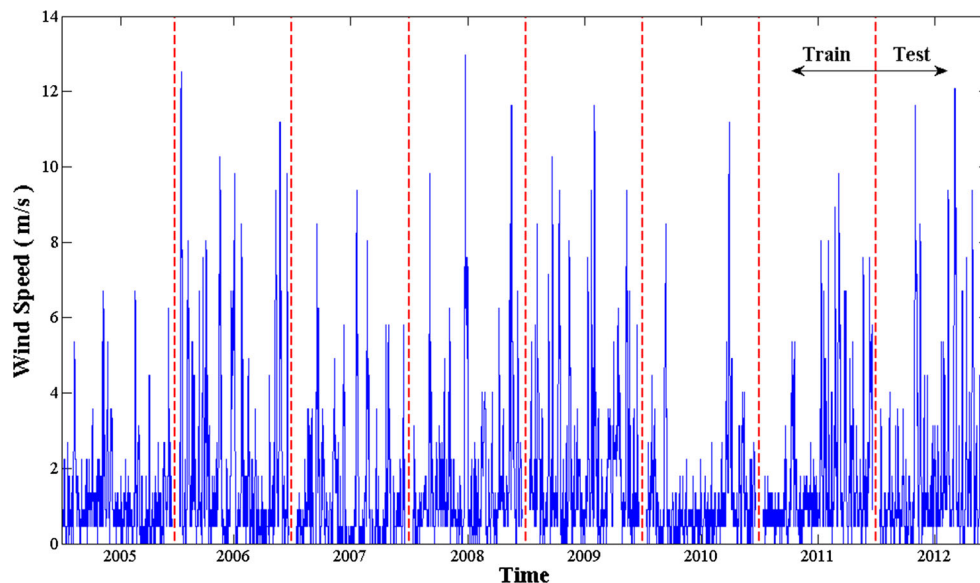
The Persistence method is the simplest way for the short-term forecasting of wind speed and assumes that the future wind speed is the same as the current one or  $\hat{x}_{t+k} = x_t$ . This

**Table 1** Summary of wind speed data ( $x_t$ , m/s) used in the study

Data set	Training	Testing
Number of data points	5206	744
Mean	1.56	1.78
Maximum value	12.96	12.07
Minimum value	0	0
Standard deviation	1.85	2.14
Coefficient of variation	1.20	1.20
Skewness	2.37	2.29



**Fig. 5** Time series plots of hourly wind speeds at the Kersey site for the period of January of each year



method is used as reference for evaluating the performance of advanced forecasting methods (Zhu and Genton 2012).

## 2.5 Used data and performance criteria

In this study, hourly wind speed time series were used from the Kersey site in Colorado, USA (latitude  $40^{\circ}22'36''$  north and longitude  $104^{\circ}31'55''$  west; altitude 1409.7 m above sea level).

The dataset used in this study corresponds to the months of January 2005 through to January 2012. The wind speed characteristics are presented in Table 1 for the site and their time series are displayed in Fig. 5 at the hourly interval.

Hourly wind speed time series for the site was downloaded from the Colorado Climate Center (<http://ccc.atmos.colostate.edu/>). As shown in Fig. 5, the data for the first 7 years (2005–2011) were used for training and that for 2012 for model testing. Four performance criteria are employed, comprising: correlation coefficient (CC), root mean square error (RMSE), Nash–Sutcliffe efficiency coefficient ( $E$ ) and Akaike information criterion (AIC). These are used to assess the goodness of fit for the selected models. The latter two performance criteria are expressed as:

$$E = 1 - \frac{\sum_{i=1}^N (x_{\text{obs}} - x_{\text{comp}})^2}{\sum_{i=1}^N (x_{\text{obs}} - \bar{x}_{\text{obs}})^2}, \quad (5)$$

$$\text{AIC} = N \cdot \text{Ln}(\text{MSE}) + 2k. \quad (6)$$

In the expressions,  $i$  is an integer varying from 1 to  $N$ ,  $x_{\text{obs}}$  and  $x_{\text{comp}}$  are the observed and computed wind speed,

respectively, the average value of the associated variable is represented with a bar above it,  $N$  is the total number of records and  $k$  is the number of model coefficients or parameters, and MSE is mean square errors.

The CC, which may range from  $-1$  to  $1$ , is a statistical measure of how well the regression line fits the observed data; a coefficient value of one indicates that the regression line perfectly fits the observed data. The CC also indicates whether or not the two variables move in the same or opposite directions and the degree of linear association.

The RMSE can provide a balanced evaluation of the goodness of fit of the model; the best RMSE would be zero, or close to it.

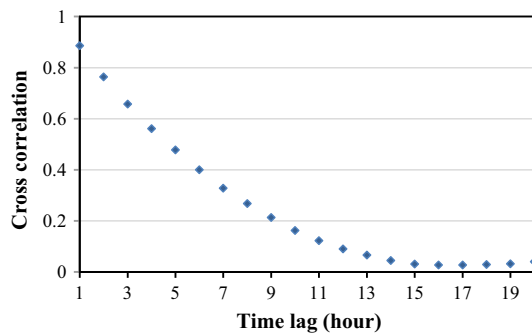
The range of  $E$  lies between  $1.0$  (perfect fit) and  $-\infty$ . A lower than zero efficiency indicates that the mean value of the observed time series would have been a better predictor than the model being tested.

The Akaike information criterion (AIC) indicates by how much the dependent variable changes with the change of the predictant (Kisi and Guven 2010). In the selection process, the model with a lower AIC value is preferred.

## 3 Results

### 3.1 Selection of inputs

Inclusion of multiple inputs into any predictive system is often at the expense of increasing system complexity, but with diminishing returns. Therefore, selection of relevant input variables is an important problem when developing such systems. Overview of past studies indicated that modeling/predicting of wind speeds based on machine



**Fig. 6** Cross correlations between time-lagged wind speed (m/s) values at the Kersey site

learning techniques usually involved using several previous wind records (see e.g., Mohandes et al. (2004); Monfared et al. (2009); Cadenas and Rivera (2010); Li and Shi (2010); Sheela and Deepa (2013)).

In this study, input variables were determined by the cross-correlation between the wind speed at the present time  $x(t)$  and time-lagged wind speed  $x(t - 1), x(t - 2), \dots, x(t - d)$ . Figure 6 shows the variation of the cross-correlation against the lag time for the wind speed at the Kersey site. It is seen clearly that the cross-correlation values are higher than zero for time lags of up to 14 h, and approximately equal to zero after a time lag of 15 h. Therefore, any value of wind speed to be forecasted is regressed on a maximum number of its antecedent 14 values. The 14 input structures are shown in Table 2 and these models were used to train and test the developed ANNs, GEP, Persistence, and MLR models.

**Table 2** Model structures for hourly wind speed prediction

Model	Input	Output
M1	$x_{t-1}$	$x_t$
M2	$x_{t-1}, x_{t-2}$	$x_t$
M3	$x_{t-1}, x_{t-2}, x_{t-3}$	$x_t$
M4	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}$	$x_t$
M5	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}$	$x_t$
M6	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}$	$x_t$
M7	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}$	$x_t$
M8	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}$	$x_t$
M9	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}$	$x_t$
M10	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}$	$x_t$
M11	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}, x_{t-11}$	$x_t$
M12	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}, x_{t-11}, x_{t-12}$	$x_t$
M13	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}, x_{t-11}, x_{t-12}, x_{t-13}$	$x_t$
M14	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}, x_{t-11}, x_{t-12}, x_{t-13}, x_{t-14}$	$x_t$

### 3.2 ANN training

The training of the ANN model used the inputs as represented in Fig. 3 and the outputs obtained from the network are compared with the target output values to estimate errors. The computed errors are back-propagated through the network and the connection weights are updated, until reaching a desirable level of performance, if at all. The logarithmic sigmoid transfer function was used in the hidden layer and the linear transfer function was employed from the hidden layer to the output layer, because the linear function is known to be robust for a continuous output variable.

Selection of the number of neurons for the hidden layer intended achieving the best network architecture. A three-layer network that achieved the minimum value of RMSE was selected.

The M1 to M14 model structures with different input structures were trained and tested. The optimum number of neurons in the hidden layer was identified for  $I, 2I,$  and  $2I + 1$ , where  $I$  is the number of inputs, which was successfully implemented by, e.g., Makarynsky (2004); Mishra and Desai (2006); Makarynsky and Makarynska (2007). The values applied in the input and output layers were normalized in the range from 0 to 1. The effect of changing the number of hidden neurons on the CC, RMSE,  $E$ , and AIC for each model is presented in Table 3.

In the training phase, the M8 model, where the number of hidden neurons is equal to  $I$ , demonstrates the best CC, RMSE and  $E$  statistic values of 0.892, 0.839 m/s and 0.796, respectively. This is selected as for the comparison

**Table 3** Comparison of ANN structures for the Kersey Site

Model	Hidden Layer	Training				Testing			
		CC	RMSE (m/s)	<i>E</i>	AIC	CC	RMSE (m/s)	<i>E</i>	AIC
M1	I	0.881	0.878	0.777	-1350.6	0.909	0.896	0.826	-162.1
	2I	0.882	0.875	0.778	-1384.5	0.909	0.895	0.826	-162.7
	2I + 1	0.882	0.875	0.778	-1384.7	0.909	0.895	0.826	-163.1
M2	I	0.884	0.868	0.782	-1470.6	0.913	0.877	0.833	-192.0
	2I	0.886	0.862	0.785	-1537.4	0.913	0.878	0.832	-190.1
	2I + 1	0.886	0.861	0.785	-1547.7	0.914	0.876	0.833	-193.3
M3	I	0.884	0.869	0.781	-1455.8	0.914	0.874	0.834	-193.8
	<b>2I<sup>a</sup></b>	<b>0.881</b>	<b>0.885</b>	<b>0.773</b>	<b>-1265.5</b>	0.909	0.900	0.824	-151.5
	2I + 1	0.881	0.879	0.776	-1339.5	0.913	0.876	0.833	-191.4
M4	I	0.886	0.863	0.784	-1524.9	0.912	0.882	0.831	-179.0
	2I	0.887	0.858	0.787	-1584.1	0.913	0.876	0.833	-189.8
	2I + 1	0.888	0.854	0.789	-1630.3	0.912	0.886	0.829	-171.7
M5	I	0.885	0.866	0.783	-1491.9	0.911	0.887	0.829	-167.9
	2I	0.888	0.856	0.788	-1610.4	0.911	0.889	0.828	-165.7
	2I + 1	0.886	0.864	0.784	-1509.7	0.912	0.886	0.829	-169.7
M6	I	0.886	0.864	0.784	-1505.1	0.911	0.888	0.828	-164.1
	2I	0.887	0.860	0.786	-1551.7	0.909	0.899	0.824	-146.3
	2I + 1	0.884	0.870	0.781	-1440.6	0.911	0.889	0.828	-162.3
M7	I	0.887	0.858	0.787	-1578.8	0.909	0.899	0.824	-144.2
	2I	0.887	0.857	0.787	-1586.3	0.909	0.900	0.824	-142.3
	2I + 1	0.891	0.843	0.794	-1759.7	0.912	0.883	0.830	-170.5
M8	I	0.891	0.843	0.794	-1761.0	0.909	0.894	0.826	-150.1
	2I	0.891	0.851	0.790	-1658.1	0.903	0.933	0.811	-87.7
	2I + 1 <sup>b</sup>	0.888	0.856	0.788	-1598.8	<i>0.914</i>	<i>0.874</i>	<i>0.834</i>	<i>-184.8</i>
M9	<b>I<sup>c</sup></b>	<b>0.892</b>	<b>0.839</b>	<b>0.796</b>	<b>-1803.2</b>	<b>0.908</b>	<b>0.899</b>	<b>0.824</b>	<b>-140.2</b>
	2I	0.886	0.862	0.785	-1527.2	0.909	0.903	0.823	-133.8
	2I + 1	0.881	0.879	0.776	-1319.6	0.900	0.937	0.809	-79.2
M10	I	0.887	0.864	0.784	-1497.4	0.911	0.899	0.824	-138.1
	2I	0.883	0.882	0.775	-1285.0	0.904	0.941	0.808	-70.8
	2I + 1	0.886	0.866	0.783	-1475.4	0.909	0.906	0.821	-126.1
M11	I	0.886	0.862	0.785	-1526.5	0.911	0.892	0.827	-147.6
	2I	0.889	0.853	0.789	-1629.0	0.905	0.919	0.816	-103.9
	2I + 1	0.888	0.859	0.787	-1562.0	0.905	0.921	0.816	-100.6
M12	I	0.885	0.868	0.782	-1449.0	0.913	0.880	0.831	-165.5
	2I	0.884	0.879	0.777	-1321.2	0.899	0.959	0.800	-38.2
	2I + 1	0.886	0.867	0.783	-1458.5	0.913	0.884	0.830	-158.9
M13	I	0.889	0.853	0.789	-1622.0	0.913	0.881	0.831	-162.6
	2I	0.889	0.851	0.790	-1645.0	0.910	0.890	0.828	-146.8
	2I + 1	0.887	0.874	0.779	-1366.9	0.911	0.902	0.823	-126.6
M14	I	0.890	0.848	0.792	-1689.7	0.910	0.893	0.826	-139.1
	2I	0.885	0.866	0.783	-1469.5	0.907	0.906	0.821	-118.1
	2I + 1	0.885	0.865	0.784	-1483.4	0.905	0.916	0.817	-101.6

<sup>a</sup> The model structure in bold italics suggests the model selection on the basis of the AIC value

<sup>b</sup> The results in italics above identify the best values for the testing periods, which do not correspond to the selected model

<sup>c</sup> The results in bold show the selected model specified as: M8(8,8,1)



**Table 4** Defined operators for GEP modeling

Function set	Operator
F1	{+, −, *, /}
F2	{+, −, *, /, $x^2$ }
F3	{+, −, *, /, Exp( $x$ )}
F4	{+, −, *, /, $x^2$ , $x^3$ , $x^{1/3}$ , exp, Ln, $\sqrt{x}$ , Sin( $x$ ), Cos( $x$ ), Arctang( $x$ )}

**Table 5** Initial setting for implementing the GEP models

General setting		Genetic operators	
Number of chromosomes	30	Mutation rate	0.044
Head size	8	Inversion rate	0.1
Number of genes	4	IS transposition rate	0.1
Number of generation	1000	RIS transposition rate	0.1
Linking function	Addition	One-point recombination rate	0.3
Fitness function		Two-point recombination rate	0.3
Error type	RMSE	Gene recombination rate	0.1
		Gene transposition rate	0.1

purpose, although the best performance for the validation phase is the M8 model with  $2I + 1$  numbers of hidden neurons (the best CC, RMSE and  $E$  statistic values of 0.914, 0.874 m/s and 0.834, respectively). Hence, in accordance with the performance indices, ANN (8, 8, 1) has been selected as the most effective and appropriate ANN model. Notably, the performance parameters are not overly sensitive to model structures and there is a conflict between the performance parameters, such that the above selected model structure based on CC, RMSE and  $E$  does not produce the lowest AIC value. Therefore, according to the AIC values, the lowest value is  $-1266$  as produced by the model structure of M3 with  $2I$ -number of hidden layers.

### 3.3 GEP training

In the preliminary investigations for the GEP model, the four sets of operators presented in Table 4 were tested.

The GEP structure was developed based on the authors' previous studies, literature review and trial-and-error procedure. The chromosomes of GEP are generally composed of more than one gene. Each gene codes for a sub-program or sub-expression tree. Then the sub-program can interact with one another in different ways, forming a more complex program. In the GEP model, the sub-expression tree must be linked through the linking function. Both the addition and multiplication operators were tested in the model, and it was found that the addition function provided a better fitness value. This study employed both: (i) the RMSE fitness function (based on the absolute error), and (ii) the various chromosomes linked together into algebraic subtrees by adding them, as opposed to multiplying them.

The default parameters and the architecture used for GEP modeling are given in Table 5. In this table, insertion sequence elements or IS elements are short fragments of the genome with a function or terminal in the first position that transposes to the head of genes except the root. The IS transposition operator randomly chooses the chromosome, the start and termination points of the IS element and the target site. Root insertion sequence (RIS) elements are short fragments with a function in the first position that transpose to the start position of genes. The RIS transposition operator randomly chooses the chromosome, the gene to be modified and the start and termination points of the RIS element.

The set of GEP model structures, defined in Table 4, were investigated. Their performances were evaluated for each model combination in Table 2 and the results of this investigation are presented in Table 6. The results show that model structures are almost insensitive to the values of CC, RMSE and  $E$ , but there is more discrimination suggested by the AIC values. Based on CC, RMSE and  $E$  values, the M2 model with the function type F3 suggests the best performance for the training phase, which produces the statistic values of 0.889, 0.869 m/s and 0.79 (and AIC value of  $-1615.9$ , which is not the best), respectively, and this is selected as the representative model. Notably, the same M2 model with the function type F3 performs the best in terms of the CC, RMSE and  $E$  statistic values of 0.884, 0.909 m/s and 0.781, respectively, for the testing phase. However, the quality of these performance parameters drops in the testing phase, but the performance parameters for the testing phase should not be used for decision making and is discussed in more detail in the next section. Notably, based on the AIC values, the M14 model with the function type F2 suggests the best performance for

**Table 6** The results of GEP models for the training and testing period

Model	Function Set	Training				Testing			
		CC	RMSE ( <i>m/s</i> )	<i>E</i>	AIC	CC	RMSE ( <i>m/s</i> )	<i>E</i>	AIC
M1	F1	0.886	0.879	0.785	-1494.9	0.879	0.925	0.773	-24.04
	F2	0.886	0.879	0.785	-1487.2	0.879	0.926	0.772	-23.65
	F3	0.886	0.877	0.786	-1509.2	0.881	0.921	0.775	-25.72
	F4	0.886	0.879	0.785	-1487.6	0.880	0.925	0.773	-24.09
M2	F1	0.887	0.875	0.787	-1536.3	0.882	0.918	0.776	-24.83
	F2	0.886	0.890	0.780	-1343.0	0.879	0.941	0.765	-16.41
	F3 <sup>a</sup>	<b>0.889</b>	<b>0.869</b>	<b>0.790</b>	<b>-1615.9</b>	<b>0.884</b>	<b>0.909</b>	<b>0.781</b>	<b>-28.15</b>
	F4	0.887	0.876	0.786	-1523.4	0.882	0.916	0.777	-25.32
M3	F1	0.887	0.876	0.786	-1518.4	0.881	0.920	0.775	-21.77
	F2	0.888	0.872	0.789	-1580.0	0.884	0.910	0.780	-25.52
	F3	0.887	0.875	0.787	-1534.6	0.881	0.919	0.776	-22.19
	F4	0.886	0.879	0.785	-1480.5	0.881	0.921	0.775	-21.50
M4	F1	0.887	0.882	0.784	-1444.9	0.881	0.925	0.773	-18.05
	F2	0.886	0.896	0.777	-1264.1	0.879	0.945	0.763	-10.70
	F3	0.888	0.873	0.788	-1558.8	0.883	0.916	0.777	-21.24
	F4	0.886	0.879	0.785	-1488.9	0.878	0.930	0.771	-16.30
M5	F1	0.886	0.890	0.780	-1334.1	0.879	0.940	0.765	-10.33
	F2	0.887	0.888	0.781	-1364.6	0.881	0.935	0.768	-12.35
	F3	0.887	0.879	0.785	-1480.7	0.880	0.925	0.773	-16.00
	F4	0.886	0.880	0.785	-1466.7	0.880	0.921	0.775	-17.31
M6	F1	0.887	0.889	0.780	-1352.7	0.880	0.937	0.767	-9.51
	F2	0.887	0.878	0.786	-1492.8	0.880	0.927	0.772	-12.89
	F3	0.887	0.879	0.785	-1479.5	0.881	0.917	0.776	-16.42
	F4	0.887	0.877	0.786	-1505.4	0.881	0.921	0.775	-15.18
M7	F1	0.887	0.876	0.787	-1515.2	0.881	0.921	0.775	-13.11
	F2	0.887	0.876	0.786	-1510.4	0.881	0.920	0.775	-13.37
	F3	0.887	0.898	0.776	-1229.3	0.880	0.943	0.764	-5.04
	F4	0.887	0.875	0.787	-1530.4	0.881	0.919	0.776	-13.70
M8	F1	0.886	0.906	0.772	-1126.4	0.879	0.955	0.758	1.50
	F2	0.886	0.906	0.772	-1126.4	0.879	0.955	0.758	1.50
	F3	0.888	0.872	0.789	-1564.6	0.883	0.913	0.778	-13.55
	F4	0.888	0.873	0.788	-1546.6	0.882	0.919	0.776	-11.46
M9	F1	0.886	0.883	0.783	-1424.1	0.879	0.929	0.771	-5.55
	F2	0.886	0.906	0.772	-1123.4	0.879	0.955	0.758	3.74
	F3	0.887	0.876	0.787	-1508.5	0.881	0.918	0.776	-9.68
	F4	0.887	0.876	0.787	-1513.2	0.881	0.918	0.776	-9.55
M10	F1	0.886	0.906	0.772	-1122.4	0.879	0.955	0.758	6.00
	F2	0.887	0.876	0.787	-1513.7	0.881	0.920	0.775	-6.73
	F3	0.887	0.875	0.787	-1517.7	0.881	0.918	0.776	-7.41
	F4	0.886	0.879	0.785	-1465.3	0.881	0.919	0.776	-7.13
M11	F1	0.886	0.906	0.772	-1119.2	0.879	0.955	0.758	8.29
	F2	0.887	0.876	0.787	-1506.6	0.883	0.912	0.779	-7.24
	F3	0.887	0.888	0.781	-1347.8	0.881	0.936	0.767	1.47
	F4	0.887	0.876	0.787	-1510.8	0.882	0.916	0.777	-5.75
M12	F1	0.888	0.875	0.787	-1520.4	0.882	0.917	0.777	-3.14
	F2	0.887	0.888	0.781	-1348.4	0.880	0.937	0.767	4.14
	F3	0.886	0.906	0.772	-1117.0	0.879	0.955	0.758	10.61
	F4	0.886	0.886	0.782	-1368.1	0.880	0.932	0.769	2.42

**Table 6** continued

Model	Function Set	Training				Testing			
		CC	RMSE (m/s)	E	AIC	CC	RMSE (m/s)	E	AIC
M13	F1	0.887	0.899	0.776	-1203.8	0.880	0.944	0.763	9.06
	F2	0.887	0.885	0.783	-1389.4	0.881	0.930	0.770	4.16
	F3	0.886	0.906	0.772	-1114.7	0.879	0.955	0.758	12.96
	F4	0.887	0.876	0.787	-1503.3	0.881	0.919	0.775	0.16
M14	F1	0.886	0.906	0.772	-1111.5	0.879	0.955	0.758	15.34
	<b>F2<sup>b</sup></b>	<b>0.886</b>	<b>0.906</b>	<b>0.772</b>	<b>-1110.1</b>	<b>0.879</b>	<b>0.955</b>	<b>0.758</b>	<b>15.36</b>
	F3	0.887	0.879	0.785	-1455.1	0.883	0.919	0.776	2.35
	F4	0.886	0.884	0.783	-1400.7	0.880	0.925	0.773	4.59

<sup>a</sup> The results in bold show the selected model specified as: M2-F3

<sup>b</sup> The model structure in bold italics suggests the model selection on the basis of the AIC value

the training phase, which produces the lowest AIC value of -1110.

The expression trees obtained for the GEP (M2, F3) model are as shown in Fig. 7 and the simplified form of equation derived for the expression trees is as follows:

$$x_t = -0.039 + x_{t-1} + \exp[-\exp(x_{t-1})] + 0.0136(x_{t-2})^2. \tag{7}$$

### 3.4 MLR fitting

The standard form of the MLR model based on Eq. (4) was used for the wind speed prediction. Table 7 presents the performance of the MLR model, according to which the model performance is almost insensitive to the model structure. Based on CC, RMSE and E values, the M13 model produces the best CC, RMSE and E statistics of 0.888, 0.874 m/s and 0.788 (note that the AIC value of -1527.0 is not optimum for this model structure), respectively. This is selected as the representative model. Notably, for the testing phase, the quality of these performance parameters drops and it seems that the M6 model has best performance in terms of CC, RMSE and E with statistic values of 0.882, 0.914 m/s and 0.778, respectively. However, this is not selected as discussed further in Sect. 4. The table further shows that on the basis of the AIC statistic value, M1 should be the model structure of choice with the AIC value of -1487.4.

$$x_t = 0.212 + 0.972 \times x_{t-1} - 0.104 \times x_{t-2} + 0.026 \times x_{t-3} - 0.021 \times x_{t-4} + 0.018 \times x_{t-5} + 0.08 \times x_{t-6} - 0.043 \times x_{t-7} + 0.024 \times x_{t-8} + 0.011 \times x_{t-9} - 0.039 \times x_{t-10} + 0.014 \times x_{t-11} - 0.011 \times x_{t-12} + 0.010 \times x_{t-13}. \tag{8}$$

### 3.5 The results by the Persistence method

This model requires no training due to its underlying simple structure, i.e., it is only implemented as M1 with no parameters to identify. Table 7 also presents the performance of the Persistence method, which produces the CC, RMSE, E and AIC statistic values of 0.886, 0.905 m/s, 0.772 and -1148.1, respectively, and these values for the testing periods are 0.879, 0.958 m/s, 0.757, and -12.3, respectively.

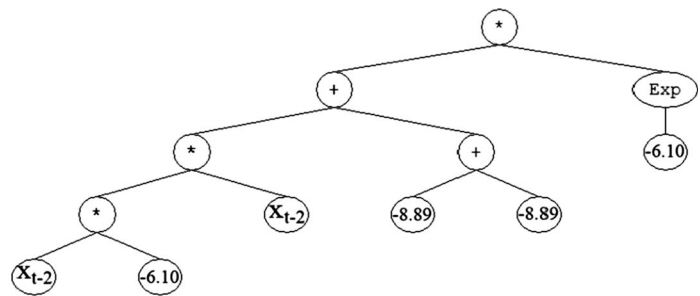
## 4 Model inter-comparisons

To further investigate the fitness of the developed models, the performance criteria of the optimum ANN, GEP and MLR models and the Persistence method were compared with one another. The overall performances of the models for training and testing datasets are summarized in Table 8 and show a mixed fortune for the models. In the first place, the CC, RMSE and E statistic values are quite insensitive to model structure. In a black-and-white world defining the actual statistic values as the basis of decision making, the ANN model performs better than the GEP and MLR models and the Persistence method. However, the Persistence method produces the lowest AIC value. In this black-and-white world, this conflict indicates a disastrous outcome with no hint to resolve the conflict. The results provide anecdotal evidence for a more detailed scrutiny of the results, as discussed below.

The predicted values were also compared with that of the observed values for the testing period and the results are presented in Fig. 8.

**Fig. 7** The expression trees for the selected GEP model in this study

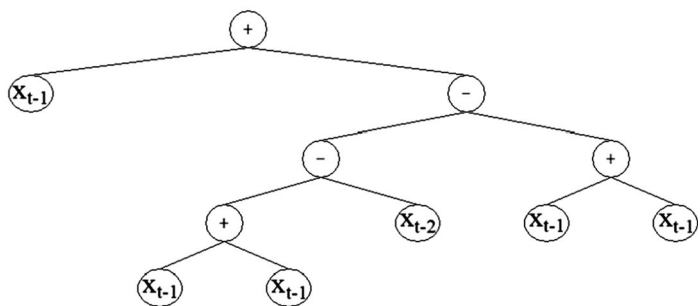
Sub-ET 1



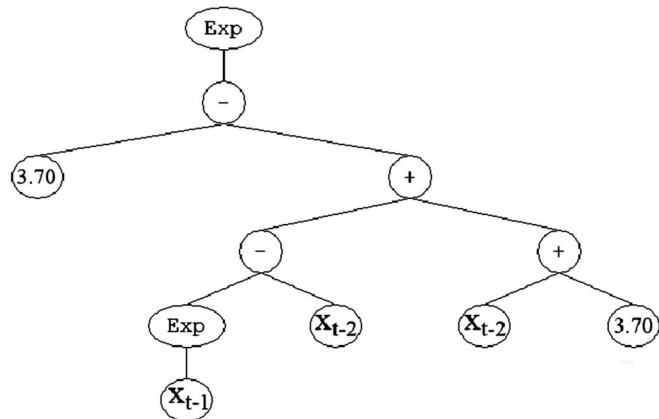
Sub-ET 2



Sub-ET 3



Sub-ET 4



## 5 Discussions

The inter-comparison of the results above suggest that the Persistence method is the “best” in terms of the AIC statistic, yet this is a simple method that does not warrant being even called a model. In fact, this method may be labeled as a model with “zero information content” i.e., low grade information. If this is the best model, does it not expose a flaw in the world of mathematical modeling that mathematical sophistication serves little? There is a very straightforward answer to this, which is: “no, the zero-order Persistence model has very little to offer.” While in a

pluralistic culture of modeling, any modeling technique has merits to be considered, the selection criteria should not be based on the nominal value of the performance parameter.

In the first place, the concept of “lead time” is a very important concept in forecasting practices, which refers to real time when forecasting is required for the future time. The longer the lead time in a forecasting task, the greater is the utility of the forecasting activities. However, too far into the future is associated with increasing inherent uncertainties. Figure 6 shows that the bottom line for the forecasting lead time is 14 h, but it also indicates the autocorrelation at 14 h lead time to be very low (high

**Table 7** The results of the MLR model for the training and testing period

Model	Training					Testing			
	Comb.	CC	RMSE (m/s)	<i>E</i>	AIC	CC	RMSE (m/s)	<i>E</i>	AIC
MLR	M1 <sup>a</sup>	<b>0.886</b>	<b>0.879</b>	<b>0.785</b>	<b>-1487.4</b>	0.879	0.926	0.772	-23.7
	M2	0.887	0.875	0.787	-1538.0	0.881	0.919	0.776	-24.3
	M3	0.887	0.875	0.787	-1534.9	0.881	0.919	0.776	-22.2
	M4	0.887	0.875	0.787	-1534.5	0.881	0.918	0.776	-20.4
	M5	0.887	0.875	0.787	-1531.6	0.881	0.918	0.776	-18.2
	M6 <sup>b</sup>	0.887	0.875	0.787	-1531.7	<i>0.882</i>	<i>0.914</i>	<i>0.778</i>	<i>-17.6</i>
	M7	0.887	0.875	0.787	-1534.1	0.882	0.915	0.778	-15.1
	M8	0.887	0.875	0.787	-1531.8	0.882	0.916	0.777	-12.6
	M9	0.887	0.874	0.787	-1530.5	0.882	0.915	0.778	-10.8
	M10	0.887	0.874	0.788	-1533.5	0.882	0.916	0.777	-8.2
	M11	0.888	0.874	0.788	-1530.9	0.882	0.916	0.777	-5.9
	M12	0.888	0.874	0.788	-1528.6	0.882	0.916	0.777	-3.6
	M13 <sup>c</sup>	<b>0.888</b>	<b>0.874</b>	<b>0.788</b>	<b>-1527.0</b>	<b>0.882</b>	<b>0.916</b>	<b>0.777</b>	<b>-1.3</b>
	M14	0.888	0.874	0.788	-1524.6	0.882	0.916	0.777	1.1
Persistence	-	0.886	0.905	0.772	-1148.1	0.879	0.958	0.757	-12.3

<sup>a</sup> The results in bold italics show a possible model selection based on the AIC value

<sup>b</sup> The results in italic show the model with best values for the testing periods specified as M6—this was not selected

<sup>c</sup> The results in bold show the selected model specified by Eq. (8)

**Table 8** Comparison of the performances of Persistence method, and MLR, ANN and GEP models for the Kersey site

Model	Training				Testing			
	CC	RMSE (m/s)	<i>E</i>	AIC	CC	RMSE (m/s)	<i>E</i>	AIC
ANN	0.892	0.839	0.796	-1803.2	0.908	0.899	0.824	-140.2
GEP	0.889	0.869	0.790	-1615.9	0.884	0.909	0.781	-28.15
MLR	0.888	0.874	0.788	-1527.0	0.882	0.916	0.777	-1.3
Persistence	0.886	0.905	0.772	-1148.1	0.879	0.958	0.757	-12.3

uncertainties). On this basis, the various model structures defined in Table 2 may be viewed as follows: (i) any model (ANN, GEP or MLR) using the M1 model structure extracts a limited amount of information from the recorded data; (ii) these ANN, GEP or MLR models using the M14 model structure extract a maximum amount of information from the recorded data; (iii) the model structures between M1 and M14 operate in between. The extraction of information is in terms of estimating the values of regression parameters in Eqs. (1) and (2) or (4) or expression trees.

Notwithstanding the above, these regression models are also required to be parsimonious, i.e., the number of independent variables should not be too few and not too many. It indicates that too many independent variables add to the complexity of the mathematical problem without a corresponding return (improvement) in the accuracy of the model; whereas too few of them would undermine the accuracy. The parsimony of a model is not simple either.

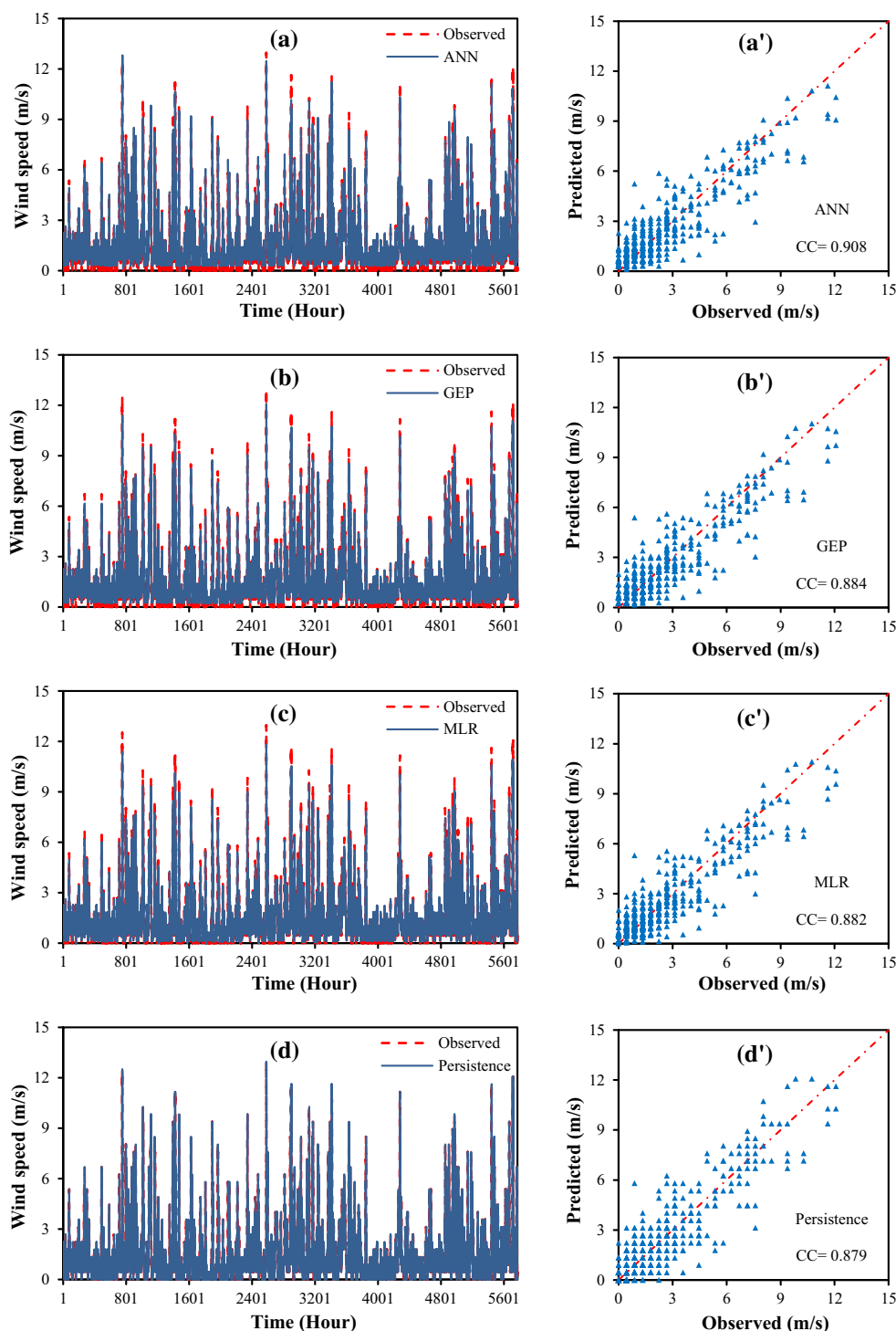
The Persistence method using the bare information within the 1 h antecedent observed value has a very limited

utility, although it is still better than nothing. Notably, it is widely known that assuming the weather today would be the same as that of yesterday can still be useful, but such a forecast can be completely wrong as well. Another use of the Persistence method is that it helps to understand the baseline of the forecasting problem better. This method is not recommended for use.

While the selected model structure of M13 for MLR (based on CC) extracts information from 13 h of auto-correlated data, that of M1 based on AIC extracts information from 1 h of antecedent record and this should also not be a preferred option. However, the M13 model structure associated with MLR has fluctuating results with a greater tendency to underestimate (or even overestimate). Hence, this model is associated with problems, though it is easy to implement.

The model structure for GEP (based on CC statistic value) is M2, but that based on AIC is M14. However, the CC values are insensitive to the model structure and therefore the choice of model structure should be based on

**Fig. 8** Comparison of the testing results of ANN, GEP, Persistence and MLR models for the Kersey site



AIC. This conflict in the implication of performance criteria is reversed by ANN, in which the model structure selection based on CC suggests M8,  $I$ , which has 8 h of information extraction base, whereas that based on AIC suggest M3,  $2I$ , which has 3 h information extraction base.

The above conflicts are not the only ones. For instance, it is expected that the best model should have the lowest

drop in the quality of the information expressed by performance criteria by applying the trained model for the forecasting at the testing period. However, the results presented above indicate that this expectation does not prevail generally. When an overall view is taken, it is clear that it is futile to seek the single best model, but different models have to be used and the individual models have to



be studied in a greater detail to gain an insight into their performance. On this basis, both GEP and ANN are equally credible selections and even MLR should not be dismissed, as it has its uses. The better performance of GEP and ANN over MLR has been underlined by other researchers, e.g., Beyer et al. (1994); Mohandes et al. (1998); Sfetsos (2000); Li and Shi (2010); Cadenas and Rivera (2009); Bilgili and Sahin (2010); Upadhyay et al. (2011).

Short-term wind speed forecasting, as studied in this paper, is crucial for the production of wind energy, aircraft and airport safety, as well as in other applications including the account given by Lei et al. (2009), Luickx et al. (2008), Monfared et al. (2009), Pinson et al. (2009) and Sfetsos (2000) on estimating the efficiency, the integration and scheduling of a wind power-generating system. Also, an insight into wind speed behavior is used for positioning the systems in a sheltered area through predictive estimations of wind energy dependent on wind (Burton et al. 2001). Modeling may be the key for cost-efficient predictions of wind speed. In this paper, a comparison of the performance of different modeling strategies shows that forecasting wind speed is feasible, but different techniques would lead to different results, where the choice between them is not easy. Thus, decision making has to be informed of these modeling results, and decisions should be arrived at on the basis of understanding inherent modeling uncertainties.

## 6 Concluding remarks

The objective of this study was to predict wind speed by using wind speed time series recorded at an hourly interval. The recorded data from the Kersey site in Colorado, USA, were used to investigate the performance of two modeling strategies: artificial neural networks (ANNs) and genetic expression programming (GEP). The obtained results were compared to multiple linear regression (MLR) model and the Persistence method.

Autocorrelation and partial autocorrelation functions for various wind speed data lags were used to find out the number of past observations to provide effective inputs to the models. The model performances were estimated and their results were compared with one another. This inter-comparisons indicated that the ANN, GEP, MLR models and the Persistence method can be successfully applied to the tasks of forecasting short-term wind speed. In this paper, a comparison of the performance of different modeling strategies shows that forecasting wind speed is feasible, but different techniques would lead to different results, where the choice between them is not easy. Thus, decision making has to be informed of these modeling results and decisions should be arrived at on the basis of understanding inherent uncertainties. The results show that it is futile to seek for the single best

model, but different models have to be used and the individual models have to be studied in greater detail to gain an insight into their performance. On this basis, both GEP and ANN are equally credible selections and even MLR should not be dismissed, as it has its uses.

**Acknowledgments** The authors would like to acknowledge and thank the Colorado Climate Center for providing the data used in this study. The authors would also like to thank the two anonymous reviewers for their valuable suggestions, which resulted in a more technically sound and complete presentation of the conducted work.

## References

- Arellano GM, Nolle L, Bland J (2012) Improving WRF-ARW wind speed predictions using genetic programming. *Res Dev Intell Syst XXIX*. doi:[10.1007/978-1-4471-4739-8\\_27](https://doi.org/10.1007/978-1-4471-4739-8_27)
- Banzhaf W, Nordin P, Keller RE, Francone FD (1998) Genetic programming-introduction: on the automatic evolution of computer programs and its applications. Morgan Kaufmann, San Francisco, CA, USA
- Barbounis TG, Theocharis JB (2007) A locally recurrent fuzzy neural network with application to the wind speed prediction using spatial correlation. *Neurocomputing* 70(7–9):1525–1542
- Beyer HG, Degner T, Hausmann J, Homan M, Rujan P (1994) Short term forecast of wind speed and power output of a wind turbine with neural networks. In: *Proceeding the 2nd European congress on intelligent techniques and soft computing*, Aachen (Germany)
- Bilgili M, Sahin B (2010) Comparative analysis of regression and artificial neural network models for wind speed prediction. *Meteorol Atmos Phys* 109(1–2):61–72
- Bilgili M, Sahin B, Yasar A (2007) Application of artificial neural networks for the wind speed prediction of target station using reference stations data. *Renew Energy* 32(14):2350–2360
- Burton T, Sharpe D, Jenkins N, Bossanyi E (2001) *Wind energy handbook*. Wiley, New Jersey
- Cadenas E, Rivera W (2009) Short term wind speed forecasting in La Venta, Oaxaca, México, using artificial neural networks. *Renew Energy* 34:274–278
- Cadenas E, Rivera W (2010) Wind speed forecasting in three different regions of Mexico, using a hybrid ARIMA-ANN model. *Renew Energy* 35:2732–2738
- Damousis IG, Alexiadis MC, Theocharis JB, Dokopoulos PS (2004) A fuzzy model for wind speed prediction and power generation in wind parks using spatial correlation. *IEEE T Energy Convers* 19(2):352–361
- Ferreira C (2001a) Gene expression programming in problem solving. In: *6th online world conference on soft computing in industrial applications (invited tutorial)*
- Ferreira C (2001b) Gene expression programming: a new adaptive algorithm for solving problems. *Compl Syst* 13(2):87–129
- Ferreira C (2002) Combinatorial optimization by gene expression programming: inversion revisited. In: *Proceedings of the Argentine Symposium on Artificial Intelligence*, pp 160–174, Santa Fe, Argentina
- Ferreira C (2006) *Gene expression programming mathematical modeling by an artificial intelligence*. Springer, Heidelberg
- Flores JJ, Graff M, Cardenas E (2005) Wind prediction using genetic programming and gene expression programming. *Techniques and methodologies for modeling and simulation of systems*. AMSE, International Association for Advanced of Modeling and Simulation, Lyon France-Mexico, pp 34–40

- Ghorbani MA, Khatibi R, Aytok A, Makarynsky O (2010) Sea water level forecasting using genetic programming and comparing the performance with artificial neural networks. *Comput Geosci* 36:620–627
- Ghorbani MA, Khatibi R, Asadi H, Yousefi P (2012) Inter-comparison of an evolutionary programming model of suspended sediment time-series with other local models. In: Soto SV (ed) *Genetic programming*, ISBN 980-953-307-215-4, InTech Open Access Publisher
- Goldberg DE (1989) *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley Longman Publishing Co., Inc., Boston
- Güven A, Aytok A, Yüce MI, Aksoy H (2008) Genetic programming-based empirical model for daily reference evapotranspiration estimation. *CLEAN Soil Air Water* 36(10–11):905–912
- Haykin S (1999) *Neural networks: a comprehensive foundation*, 2nd edn. Prentice Hall, New Jersey
- Huang CY, Chiang BY, Chang SY, Tzeng GH, Tseng CC (2011) Predicting of the short term wind speed by using a real valued genetic algorithm based least squared support vector machine. *Intell Decis Technol* 10:567–575
- Ji GR, Pu H, Yong-Jie Z (2007) Wind speed forecasting based on support vector machine with forecasting error estimation. In: *Proceedings of the 6th international conference on machine learning and cybernetics*, pp 2735–2739
- Kalra R, Deo MC (2007) Genetic programming to retrieve missing information in wave records along the west coast of India. *Appl Ocean Res* 29(3):99–111
- Kalra R, Deo MC, Kumar R, Agarwal VK (2008) Genetic programming to estimate coastal waves from deep water measurements. *Int J Ecol Dev* 10(8):67–76
- Kariniotakis G, Stavrakakis GS, Nogaret EF (1996a) A fuzzy logic and neural network based wind power model. In: *Proceeding the European wind energy conference*. Goteborg (Sweden), pp 596–599
- Kariniotakis G, Stavrakakis GS, Nogaret EF (1996b) Wind power forecasting using advanced neural network models. *IEEE T Energy Conver* 11(4):762–767
- Khatibi R, Ghorbani MA, Kashani MH, Kisi O (2011) Comparison of three artificial intelligence techniques for discharge routing. *J Hydrol* 403(3–4):201–212
- Khatibi R, Ghorbani MA, Naghipour L, Jothiprakash V, Fathima TA, FazeliFard MH (2014) Inter-comparison of time series models of lake levels predicted by several modeling strategies. *J Hydrol* 511:530–545
- Kisi O (2005) Daily river flow forecasting using artificial neural networks and auto-regressive models. *J Eng Env Sci* 29:9–20
- Kisi O, Güven A (2010) A machine code-based genetic program for suspended sediment concentration estimation. *Adv Eng Softw* 41(7–8):939–945
- Koza JR (1992) *Genetic Programming: On the programming of computers by means of Natural Selection*. The MIT Press, Bradford Book, Cambridge
- Lei M, Shiyan L, Chuanwen J, Hongling L, Yan Z (2009) A review on the forecasting of wind speed and generated power. *Renew Sustain Energy Rev* 13(4):915–920
- Li G, Shi J (2010) On comparing three artificial neural networks for wind speed forecasting. *Appl Energy* 87(7):2313–2320
- Luickx PJ, Delarue ED, D'haeseleer WD (2008) Considerations on the backup of wind power: operational backup. *Appl Energy* 85(9):787–799
- Makarynsky O (2004) Improving wave predictions with artificial neural networks. *Ocean Eng* 31(5–6):709–724. doi:10.1016/j.oceaneng.2003.05.003
- Makarynsky O, Makarynska D (2007) Wave prediction and data supplementation with artificial neural networks. *J Coastal Res* 23(4):951–960
- McCullon WS, Pitts W (1943) A logical calculus of the ideas immanent in neurons activity. *Bull Math Biophys* 5(4):115–133
- Melesse AM, Hanley RS (2005) Artificial neural network application for multi ecosystem carbon flux simulation. *Ecol Model* 189(3–4):305–314. doi:10.1016/j.ecolmodel.2005.03.014
- Mishra AK, Desai VR (2006) Drought forecasting using feed-forward recursive neural network. *Ecol Model* 198:127–138
- Mohandes MA, Rehman S, Halawani TO (1998) A neural networks approach for wind speed prediction. *Renew Energy* 13(3):345–354
- Mohandes MA, Halawani TO, Rehman S, Hussain A (2004) Support vector machines for wind speed prediction. *Renew Energy* 29(6):939–947
- Monfared M, Rastegar H, Kojabadi H (2009) A new strategy for wind speed forecasting using artificial intelligent methods. *Renew Energy* 34(3):845–848
- More A, Deo MC (2003) Forecasting wind with neural networks. *Mar Struct* 16(1):35–49
- Pinson P, Nielsen HA, Madsen H, Kariniotakis G (2009) Skill forecasting from ensemble predictions of wind power. *Appl Energy* 86(7–8):1326–1334
- Sfetsos A (2000) A comparison of various forecasting techniques applied to mean hourly wind speed time series. *Renew Energy* 21(1):23–35
- Sheela KG, Deepa SN (2013) Neural network based hybrid computing model for wind speed prediction. *Neurocomputing* 122:425–429
- Tandjaoui MN, Benachaiba C, Abdelkhalek O, Dennai B, Mouloudi Y (2013) The impact of wind power implantation in transmission systems. *Energy Procedia* 36:260–267
- Upadhyay KG, Choudhary AK, Tripathi MM (2011) Short-term wind speed forecasting using feed-forward back-propagation neural network. *Int J Sci Technol* 3(5):107–112
- Ustoorikar K, Deo MC (2008) Filling up gaps in wave data with genetic programming. *Mar Struct* 21(2–3):177–195
- Wang X, Sideratos G, Hatzigiorgianni N, Tsoukalas LH (2004) Wind speed forecasting for power system operational planning. In: *proceedings of the international conference on probabilistic methods applied to power systems*, pp 470–474
- Xiaojuan H, Xilin Z, Bo G (2009) Short-term wind speed prediction model of LS-SVM based on genetic algorithm advanced technology. In: *proceedings of the 3rd international conference on teaching and computational science (WTCS)*
- Zakaria NA, Azamathulla HM, Chang CK, Ghani AA (2010) Gene expression programming for total bed material load estimation—a case study. *Sci Total Environ* 408(21):5078–5085
- Zhu X, Genton MG (2012) Short-term wind speed forecasting for power system operations. *Int Stat Rev* 80(1):2–23