Applied Energy 224 (2018) 550–566

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

# Time-series prediction of wind speed using machine learning algorithms: A case study Osorio wind farm, Brazil

A. Khosravi*, L. Machado, R.O. Nunes

*Graduate Program in Mechanical Engineering, Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil*

## HIGHLIGHTS

- Machine learning algorithms are developed to predict the time-series wind speed data.
- The developed models are MLFFNN, SVR, FIS, ANFIS, ANFIS-PSO, ANFIS-GA and GMDH.
- The employed models were examined on 5-min, 10-min, 15-min and 30-min intervals.
- GMDH model for all time intervals can successfully predict the target.
- PSO and GA algorithms can increase the prediction accuracy of the ANFIS model.

## ARTICLE INFO

*Keywords:*
Wind speed prediction
Neural networks
Adaptive neuro-fuzzy inference system
Group method of data handling
Particle swarm optimization
Genetic algorithm

## ABSTRACT

Machine learning algorithms (MLAs) are applied to predict wind speed data for Osorio wind farm that is located in the south of Brazil, near the Osorio city. Forecasting wind speed in wind farm regions is valuable in order to obtain an intelligent management of the generated power and to promote the utilization of wind energy in grid-connected and isolated power systems. In this study, multilayer feed-forward neural network (MLFFNN), support vector regression (SVR), fuzzy inference system (FIS), adaptive neuro-fuzzy inference system (ANFIS), group method of data handling (GMDH) type neural network, ANFIS optimized with particle swarm optimization algorithm (ANFIS-PSO) and ANFIS optimized with genetic algorithm (ANFIS-GA) are developed to predict the time-series wind speed data. The Time-series prediction describes a model that predicts the future values of the system only using the past values. Past data is entered as input and future data to be used for represents MLA output. The developed models are examined on 5-min, 10-min, 15-min and 30-min intervals of wind speed data. The results demonstrated that the GMDH model for all time intervals can successfully predict the time-series wind speed data with a high accuracy. Also, the combination of ANFIS models with PSO and GA algorithms can increase the prediction accuracy of the ANFIS model for all time intervals.

## 1. Introduction

The depletion of fossil fuel energy resources, increasing concern of air pollution, global warming and energy crisis have encouraged the research for clean and pollution-free sources of energy [1]. Wind energy is one of the most common of clean energies that has been developed significantly in the world [2]. This type of energy is more accessible, inexhaustible, fairly cheaper, renewable and sustainable, and environmentally friendly [3]. Wind energy is a free source of energy that has served the mankind for many countries for driving wind turbines, pumping water, ships, etc. Developing wind energy systems can improve the idea of electricity generation without pollution in the future [4]. However, the integration of wind farms into the power networks has become an important problem for commitment and control of power plants, connecting and disconnecting the power to the grid and management of the power [5]. The produced power by wind turbine is related to the wind speed. Wind is considered one of the weather variables which more difficult to be estimated [6]. Wind is intermittent in nature, so it is not possible to predict exact wind speed because of the continuously changing climate conditions.

Therefore, the advent of alternative energy sources, particularly wind power, and the need to manage energy resources necessitate the use of advanced tools for prediction of short-term wind speed (or other types of renewable energies) [7]. The contribution of wind speed prediction for a safe and economic operation of the network will organize by independent power producer, system operator distribution and

* Corresponding author.
*E-mail address:* Alikhosraviis86@ufmg.br (A. Khosravi).

https://doi.org/10.1016/j.apenergy.2018.05.043
Received 7 December 2017; Received in revised form 19 April 2018; Accepted 7 May 2018

electrical companies. In recent years many investigations have proposed machine learning algorithms (MLAs) (or Artificial Intelligence methods) to predict the variation of meteorological data such as wind speed, solar irradiance, relative humidity, and air temperature [8–15]. Artificial Intelligence methods are defined as an extensive scientific discipline which enable computer programs to solve the complex nonlinear problems. In this study, seven types of MLAs are implemented to predict the time-series wind speed data (a time series prediction describes a model that predict the future values of the system only using the past values). It is exposed the procedure to achieve the possible models which better explain the time-series wind speed behavior. These methods are described as follows:

The first method is multilayer feed forward neural network (MLFFNN) that is defined as a new method of programming computers. This method is employed to analyze the problems that are very difficult to solve using conventional techniques [16]. The second method is group method of data handling (GMDH) type neural network. This method for the first time has been introduced by Ivakhnenko [17] as a multivariate analysis approach for complicated systems modeling and recognition. GMDH with the algebraic approach of the progression avoids the complexity of obtaining former information [18]. Therefore, the complicated systems without having particular information of the systems will simulate by applying this method.

The third method is support vector regression (SVR) that optimize its structure based on the input data. This method has been introduced by Vapnik [19], which works based on the classification and regression technique [20]. In recent years, SVR was successfully employed on classification tasks in very different areas of application [21–24]. The fourth method is fuzzy inference system (FIS) that is defined as the process of mapping a set of input data into a set output data, using an approach based on fuzzy logic. For this model, membership functions (MFs), fuzzy logic operators and if-then rules are defined as the main parts of the fuzzy inference system [25]. The fifth method is adaptive neuro-fuzzy inference system (ANFIS). An ANFIS model is defined as a combination of an ANN (generally, radial basis function neural network) into a fuzzy inference system. This combination is carried out to obtain the knowledge of the human expert to adjust the fuzzy parameters [26]. The sixth method is the combination of particle swarm optimization (PSO) algorithm with ANFIS model (ANFIS-PSO). The PSO algorithm is employed to increase the performance of the ANFIS model, tuning the membership functions required to achieve a lower error. This optimization algorithm that was proposed by Kennedy and Eberhart in 1995 is a heuristic approach [27]. The seventh method is based on the interconnection of genetic algorithm (GA) and ANFIS model (ANFIS-GA). Many studies for prediction of wind speed have been published that a short summary of them is presented in the following.

Chang et al. [28] developed a radial basis function neural network (RBFNN) for short-term wind speed and power forecast. The proposed network was trained with 24 h of observation period with historical data. Also, their method was compared with other methods of neural networks. They obtained the mean absolute percentage error (MAPE) for backpropagation neural network (BPNN), RBFNN and ANFIS as 27%, 24% and 3.87%, respectively. Ramasamy et al. [29] proposed an artificial neural network (ANN) to estimate wind speed in the mountains region of India. Temperature, solar radiation, air pressure and altitude were selected as inputs of the model and mean daily wind speed was selected as the target. It was reported a MAPE and correlation coefficient around 4.55% and 0.98, respectively. Doucoure et al. [30] implemented wavelet neural network and multi-resolution analysis to forecast wind speed.

Liu et al. [31] developed a modified Taylor Kriging method in order to estimate wind speed time-series data. One-year wind speed data were divided into 10 samples and the proposed model was applied to each sample. The proposed method was compared with an autoregressive integrated moving average method (ARIMA) and the results illustrated that the proposed method outperformed the ARIMA method by 18.6% and 15.2% in term of mean absolute error (MAE) and root mean square error (RMSE), respectively. Noorollahi et al. [32] used ANNs for estimating temporal and spatial wind speed in Iran. They implemented BPNN, RBFNN and ANFIS models to predict the targets. The BPNN and ANFIS models yielded similar results. The RBFNN method illustrated larger errors in all cases. Schicker et al. [33] developed an interval-artificial neural network for short-term wind speed prediction.

Kumar et al. [34] proposed generalized regression neural network (GRNN) and multilayer perceptron neural network (MLPNN) in order to predict wind speed in Western Region of India. It was found that the GRNN model has given the better result than MLPNN in term of mean square error (MSE). Sheela et al. [35] proposed neural network based on hybrid computing model for wind speed prediction. The proposed model was compared with an MLPNN and the results demonstrated the hybrid computing model performed better in terms of minimization of errors. Petkovic [36] implemented an ANFIS method for estimation of wind speed distribution. It was achieved an improvement in predictive accuracy and capability of generalization with the ANFIS approach.

Bonfil et al. [37] developed a model based on SVR to predict wind speed for wind farms. Wavelet, extreme learning machine and outlier correction algorithm were developed in order to predict wind speed by Mi et al. [38]. Also, Liu et al. [14] applied secondary decomposition algorithm and Elman neural networks for forecasting wind speed data. It was achieved a satisfactory performance for the proposed method in multi-step wind speed prediction. Li et al. [39] developed a study based on three models of ANN to predict wind speed data. These methods were adaptive linear element, backpropagation and radial basis function neural network. It was observed that the three developed models were capable at wind speed prediction. Koo et al. [40] employed ANN to predict wind speed data based on geological and distance variables (a case study in South Korea).

Lodge and Yu [41] implemented an ANN to predict the short-term wind speed data. This model was constructed based on the current weather condition and historical wind speed data. The values of RMSE for training and testing datasets were determined as 0.5781 (m/s) and 0.8895 (m/s), respectively. Mohandes et al. [42] developed a support vector machine (SVM) for forecasting the wind speed data and compared to an MLP neural network. For testing datasets, the value of MSE for SVM and MLP models were obtained 0.0078 and 0.0090, respectively. Bilgili et al. [43] used MLP neural network in order to forecast wind speed data. To train the network, resilient propagation (RP) learning algorithm was employed. Also, *logsig* and *purelin* transfer functions were selected for the hidden and output layers, respectively. The MAPE for the developed model was obtained 14.13%.

Salcedo-Sanz et al. [44] developed evolutionary support vector regression algorithms to estimate short-term wind speed prediction. An Evolutionary Programming algorithm (EP) and particle swarm optimization algorithm (PSO) were applied to tackle the hyper-parameters estimation problem in regression SVM. Their results have shown that the new model based on SVM with EP algorithm outperformed the MLP neural network. Liu et al. [45] proposed two models of ANN to predict the wind speed data. The first model was SVM optimized with genetic algorithm (SVM-GA) and the second one was a combination of wavelet transform with SVM-GA model (W-SVM-GA). It was obtained that the W-SVM-GA performed better than the SVM-GA in terms of MAPE and RMSE.

### 1.1. The knowledge gap and novelty of this study

The wind speed forecast in the wind energy sector is valuable for wind power system planning, unit commitment decision, load balancing decision, maintenance arrangement and energy storage capacity optimization. Wind speed is period arrangement information measured at various interims of time. Wind speed data are always nonlinear and non-stationary therefore it is difficult to accurately estimate them. Hence, regarding the literature review, many investigations were developed in order to predict wind speed data. These investigations have developed intelligent methods by various input variables such as meteorological and geographical data (ambient temperature and pressure, relative humidity, solar irradiance, longitude, and latitude) for wind speed data estimation. For this target, several facilities should be employed to measure the meteorological data. In this study, the wind speed data are predicted only based on known values of measured wind speed data. For the first time, the idea of GMDH type neural network is developed to predict the time-series wind speed data. This algorithm is a self-organizing method by which gradually complicated models are generated based on the evaluation of their performances on a set of multi-input single-output data pairs. Based on the literature, although many investigations have proposed ANFIS model to estimate wind speed data, the proposed ANFIS model is based on fuzzy c-means (ANFIS-FCM). In this investigation, to obtain the best performance, two other models of ANFIS are developed and compared to ANFIS-FCM. The first one uses subtractive clustering to generate a fuzzy inference system (ANFIS-SC); the second one is ANFIS-GP that creates a fuzzy system using grid partitioning on data. In order to improve the prediction accuracy and generalization ability of the ANFIS model, two optimization algorithms are applied that are PSO and GA algorithms. Also, MLFFNN, SVR and FIS models are developed to predict the time-series wind speed data in the case study region. It is noteworthy to mention that the comparison of the intelligent models for forecasting the wind speed data will be possible by considering the same dataset. Based on the literature, there are limited articles concerning on the performance comparison of the MLFFNN, FIS, SVR, ANFIS-FCM, ANFIS-SC, ANFIS-GP, GMDH, ANFIS-PSO and ANFIS-GA models for prediction of the time-series wind speed data, which is the main contribution of this study.

## 2. Case study

Brazil is one of the countries in the industrial world with largest share of clean energy. In 2021 the installed wind capacity in Brazil will be about 15,500 MW (8.5% of the total energy, compared to 1.2% current). The case study is Osorio wind farm (Fig. 1) that is located in the south of Brazil, near the Osorio city, state of Rio Grande do Sul (latitude: −29.92° and longitude: −50.31°). This place is recognized for having a large wind potential capacity associated with good infrastructure conditions and electrical grid connection, so it was chosen for wind turbines settlement. The wind farm uses 148 wind turbines with rotor diameter of 70 m, rotor-swept area of 3960 $m^2$ and it is installed power of 317.9 MW. The area of the wind farm is about 10,400 ha, which 320 ha are occupied by wind farm infrastructures. The wind farm is composed of the wind parks Sangradouro, Osorio and Indios. Due to intermittent nature of wind, prediction of accurate wind speed data is necessary for it. Forecasting methods based on artificial intelligence can successfully predict wind speed in wind farm regions. It can relieve or avoid the disadvantageous impact to the electric grid.

**Fig. 1.** Osorio wind farm, Brazil.

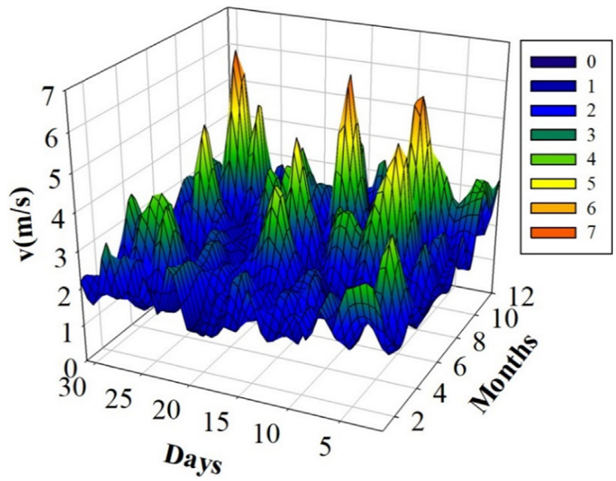


**Fig. 2.** Variation of wind speed in Osorio wind farm for an annual period.

Fig. 2 demonstrates the variation of the wind speed (m/s) for Osorio wind farm region. Due to intermittent nature of wind, it has various patterns in different area, height and length of time. The timescale is recognized as an important significance for investigation of produced wind power, and wind power studies under various timescales have different effects on the power system decision-making.

## 3. Material and methods

In this study, the intelligent methods are developed based on the past values of measured wind speed data. A possible linear or non-linear autoregressive process describes this type of prediction that is obtained by the following equation:

$$x(k + h) = f[x(k),x(k-1),x(k-2),...,x(k-n-1)] \tag{1}$$

In which the past value of the $x$ with the present value is described by the function $f$. Two ways can be applied to multi-step ahead prediction (consists of predicting the $h$ next values of the time-series) that are: iterative method consists of repeating one-step-ahead predictions to the desired horizon (this strategy is shown in Fig. 3) and independent value prediction (training the direct model to forecast $x(k + h)$).

An MLA in its most general form can be described as a function $f(x)$ which takes an input vector $x$ and generates an output vector $y$ [46]. It is an application of artificial intelligence that concentrates on the development of computer programs. MLAs are divided into two sections that are supervised learning (trains a model on known input and output data in order to predict the future outputs) and unsupervised learning

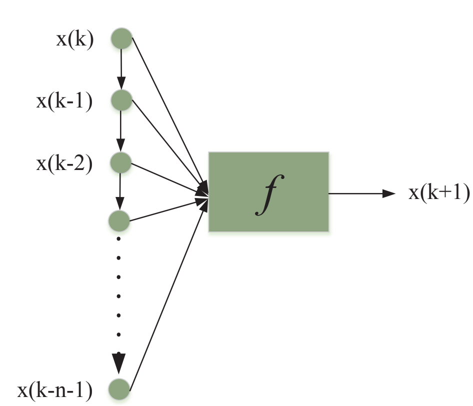


**Fig. 3.** Time-series prediction model.

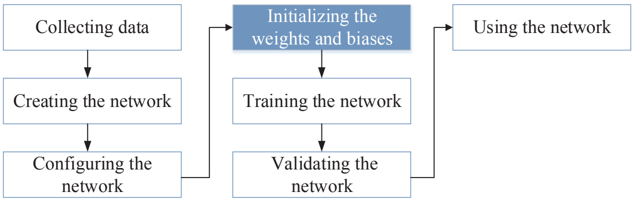


**Fig. 4.** Seven steps for designing an ANN.

(finds the hidden patterns in input data). Supervised learning uses regression technique (such as neural networks, adaptive neuro-fuzzy learning and SVR) and classification technique (such as support vector machine, boosted and neural networks) in order to find forecasting model. Also, unsupervised learning uses clustering technique that the common algorithms for implementing clustering are fuzzy k-means, neural networks and Gaussian mixture. Fig. 4 demonstrates seven steps of designing an ANN model to predict the target.

In this study, seven types of MLAs are developed in order to predict the wind speed in the case study zone, which are described in the following:

### 3.1. Multilayer feed-forward neural network (MLFFNN)

The function of a neural network is to map the relationship between the input(s) and the output(s). An ANN consists of three sections that are input layer, hidden layer and the output layer. Input vector (X) is multiplied by weight vector (W) and this multiplication vector (WX) is added to a bias (b) that led to the input vector (n). Then, this vector enters to a neuron. The neuron applies a transfer function (*f*) to this input vector that the output vector *a* is obtained. Fig. 5 shows a single neuron network that was explained in this section. Eq. (2) shows the relation between input and output,

$$n = W_{1,1}X_1 + W_{1,2}X_2 + \cdots + W_{1,r}X_r + b \tag{2}$$

Also, during training phase of the network, Eqs. (3) and (4) are applied to update the amount of the weight(s) and bias(s).

$$W(k+1) = W(k) + 2\alpha e(k)X^T(k) \tag{3}$$

$$\mathrm{b}(k+1) = \mathrm{b}(k) + 2\alpha e(k) \tag{4}$$

In which $\alpha$ and $e$ are the learning rate and error, respectively.

Multilayer neural networks are used to implement the complex problems. Fig. 6 demonstrates a multilayer neural network consists of an input layer, hidden layer and output layer. The output of a multilayer neural network is obtained by:

$$a^1 = f^1(IW^{1,1}X + b^1) \tag{5}$$

$$a^2 = f^2(LW^{2,1}a^1 + b^2) \tag{6}$$

$$a^3 = f^3(LW^{3,2}a^2 + b^3) \tag{7}$$

where X, IW, LW, b and *a* are input vector, input weight matrix, hidden weight matrix, bias vector and layer output vector, respectively.

A three-layer feed-forward neural network is developed to forecast

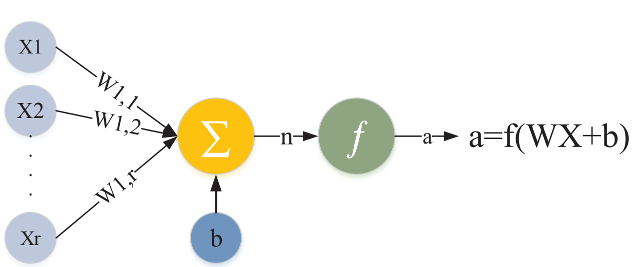


**Fig. 5.** Single neuron network.

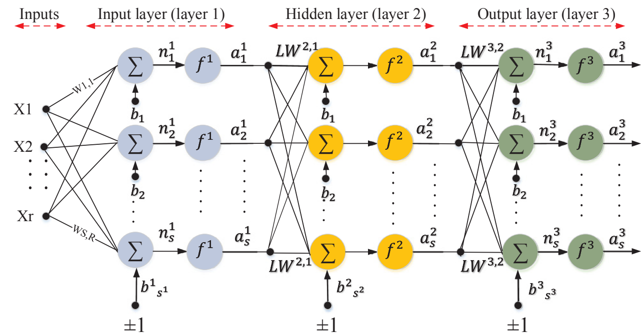


**Fig. 6.** Multilayer neural network.

the time-series wind speed data. The first layer comprises the input vectors; the second layer is the hidden layer (this layer contains a nonlinear transfer function that *tansig* is considered for that); the third layer (output layer) includes a linear function (*purelin* is selected for this layer) [47]. In this study, the MLFFNN is tested by different data training algorithms that are: BFGS Quasi-Newton (BFG) (*trainbfg*), Bayesian Regularization (BR) (*trainbr*), Levenberg Marquardt (LM) (*trainlm*), Scaled Conjugate Gradient (SCG) (*trainscg*) and Resilient Backpropagation (RP) (*trainrp*).

### 3.2. Support vector regression (SVR)

For the first time, the idea of support vector machine was proposed by Vapnik [19] for classification and regression applications. For training data $(x_i,y_i),\ldots,(x_n,y_n)$, in which $x_i$ are defined as the vectors with input values and $y_i$ are the appropriate output values for $x_i$. The regression variant was employed with the release of the ε-intensive loss function (ε-SVR). The initial target of this loss function is to develop a function where all errors lie under a predefined value ε but with the best generalization capacity possible (generally related to model flatness). Eqs. (8) and (9) are given for this definition:

$$\min_{\omega,b,\xi,\xi^*} \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{8}$$

$$s.\ t.\ \begin{cases} y_i - (\omega,x_i + b) \leqslant \varepsilon + \xi_i \\ (\omega,x_i + b) - y_i \leqslant \varepsilon + \xi_i^* \\ \xi,\xi^* \geqslant 0 \end{cases} \tag{9}$$

where n, $\xi_i$, $\xi_i^*$ and C are the number of samples, the upper training error, the lower training error and the regularized constant, respectively.

### 3.3. Fuzzy inference system (FIS)

Fuzzy inference system (FIS) is defined as the process of mapping from a given input to an output using an approach based on fuzzy logic. Membership functions (MFs), fuzzy logic operators and if-then rules are three main parts of each FIS model. Two methods have been introduced for designing a FIS that are Mamdani (is the most commonly seen FIS method) and Sugeno or Takagi–Sugeno–Kang. This two types of FIS are different in the consequent of fuzzy rules. In this study, a FIS model based on Mamdani is developed to predict the time-series wind speed data. Fig. 7 indicates a structure of FIS to compute the output from the given inputs [48].

### 3.4. Adaptive neuro-fuzzy inference system (ANFIS)

An ANFIS model is defined as a combination of artificial neural network with fuzzy inference system. This combination is done to obtain the knowledge of human expert for determining the fuzzy

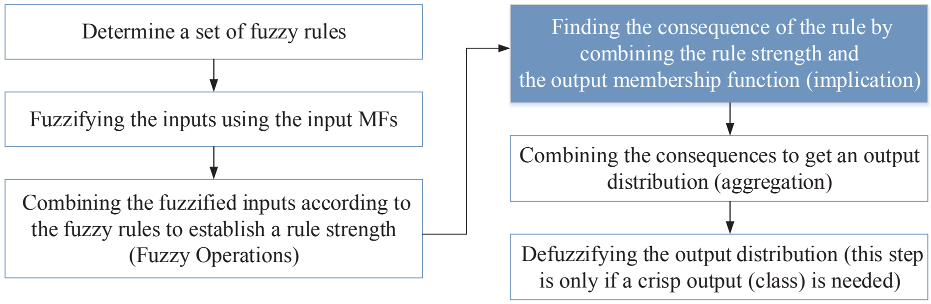


**Fig. 7.** A structure for designing a FIS model.

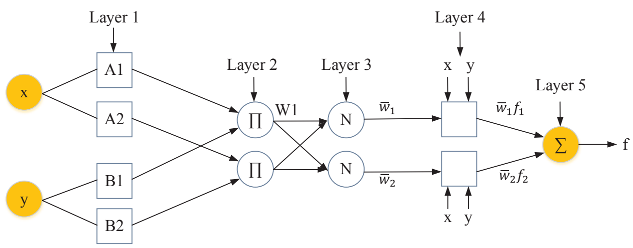


**Fig. 8.** ANFIS structure with two inputs and one output.

parameters such as if-then rules. For this target, radial basis function neural network (RBFNN) is used to incorporate into fuzzy system. Also, ANFIS model is developed based on Takagi-Sugeno type fuzzy inference system. Fig. 8 presents a basic ANFIS architecture that has two inputs and one output. The rule base contains two Takagi-Sugeno if-then rules, which are given by the following equations:

$$\text{Rule 1: If (x is } A_1\text{) and (y is } B_1\text{) then } f_1 = p_1x + q_1y + r_1 \tag{10}$$

$$\text{Rule 2: If (x is } A_2\text{) and (y is } B_2\text{) then } f_2 = p_2x + q_2y + r_2 \tag{11}$$

In which $r_i$, $p_i$, and $q_i$ are the design parameters that will be obtained during the period of training phase. As can be observed in Fig. 8, a basic ANFIS architecture has five layers that are described in the following:

Layer 1: This layer is fuzzification layer in which the signal that is obtained from each node is transferred to the other layer. For this layer the cells outputs ($O_i^1$) are:

$$O_i^1 = \mu_{A_i}(x), \quad i = 1, 2 \tag{12}$$

where $\mu_{A_i}$ is associated to the membership function and $A_i$ is linguistic variable associated with this node function.

In the most ANFIS models $\mu_{A_i}$ is selected as:

$$\mu_{A_i}(x) = exp\left\{-\left[\left(\frac{x-c_i}{a_i}\right)^2\right]^{b_i}\right\} \tag{13}$$

In which $x$ is the input and $\{a_i,b_i,c_i\}$ are premise parameters.

Layer 2 (rule layer), is achieved with the membership degrees in which each node output demonstrates the firing strength of a fuzzy rule.

$$O_2^i = w_i = \mu_{A_i}(x).\ \mu_{B_i}(y), \quad i = 1, 2 \tag{14}$$

Layer 3 (normalization layer), every node in this layer is a fixed node labeled N. The $i$th node calculates the ratio of the rule's firing strength to the sum of all rules' firing strengths:

$$O_3^i = \bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2 \tag{15}$$

Layer 4 (defuzzification layer), the output value for each rule is calculated from the value of the previous layer.

$$O_4^i = \bar{w}_i f_i = \bar{w}_i(p_i x + q_i y + r_i), \quad i = 1, 2 \tag{16}$$

In which $\bar{w}_i$ is a normalized firing strength from layer 3 and $\{p_i,q_i,r_i\}$ are the consequent parameters.

Layer 5 (sum layer), the output of ANFIS model is achieved by collecting the output values of each rule that are obtained from the previous layer.

$$O_5^i = overall\ output = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}, \quad i = 1, 2 \tag{17}$$

In this study, three types of ANFIS model are developed to predict the time-series wind speed data. The first one generates a FIS structure using fuzzy c-means clustering (ANFIS-FCM). This method is developed based on *genfis3* in Matlab software. The second one is ANFIS-SC, which generates a Sugeno-type FIS structure using subtractive clustering (based on *genfis2* in Matlab software). The third one generates a FIS structure from data using grid partition and it is called ANFIS-GP (based on *genfis1*).

### 3.5. PSO algorithm

Particle swarm optimization (PSO) algorithm was developed by Eberhart and Kennedy in 1995, which inspires from nature social behavior and dynamic movements with communications of birds, insects and fish. This algorithm uses a number of particles that create a swarm moving around in the search space looking for the best solution. Flying experience of each particle and also the flying experience of the other particles help each particle to adjust its “flying” in the search space. Indeed, each particle amends its position based on its current velocity, its current position, the distance between its current position and *pbest*, and the distance between its current position and *gbest*.

### 3.6. Genetic algorithm (GA)

Holland in 1970 introduced genetic algorithm as a class of probabilistic optimization algorithm that inspires by the biological evolution process [49]. Indeed, genetic algorithms are defined as a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as mutation, inheritance, selection and crossover. The evolution usually begins from a population of randomly generated individuals and happens in generations. Individuals in the population, during single iteration of the genetic algorithm, compete for their right to reproduce, favoring those who maximize the value of a fitness function which customarily maps every individual to a single decimal value between the [0; 1] interval. Also, fit individuals are selected and allowed to produce an offspring population, on which the next iteration of the GA operates (at the end of the iteration). This behavior creates an iterative loop.

### 3.7. Training ANFIS using optimization algorithms

The forecasting accuracy of the ANFIS (here ANFIS-FCM) model is improved by applying the optimization algorithms (PSO and GA). The ANFIS parameters are updated by the optimization algorithms. The ANFIS parameters were described as premise and consequent

 

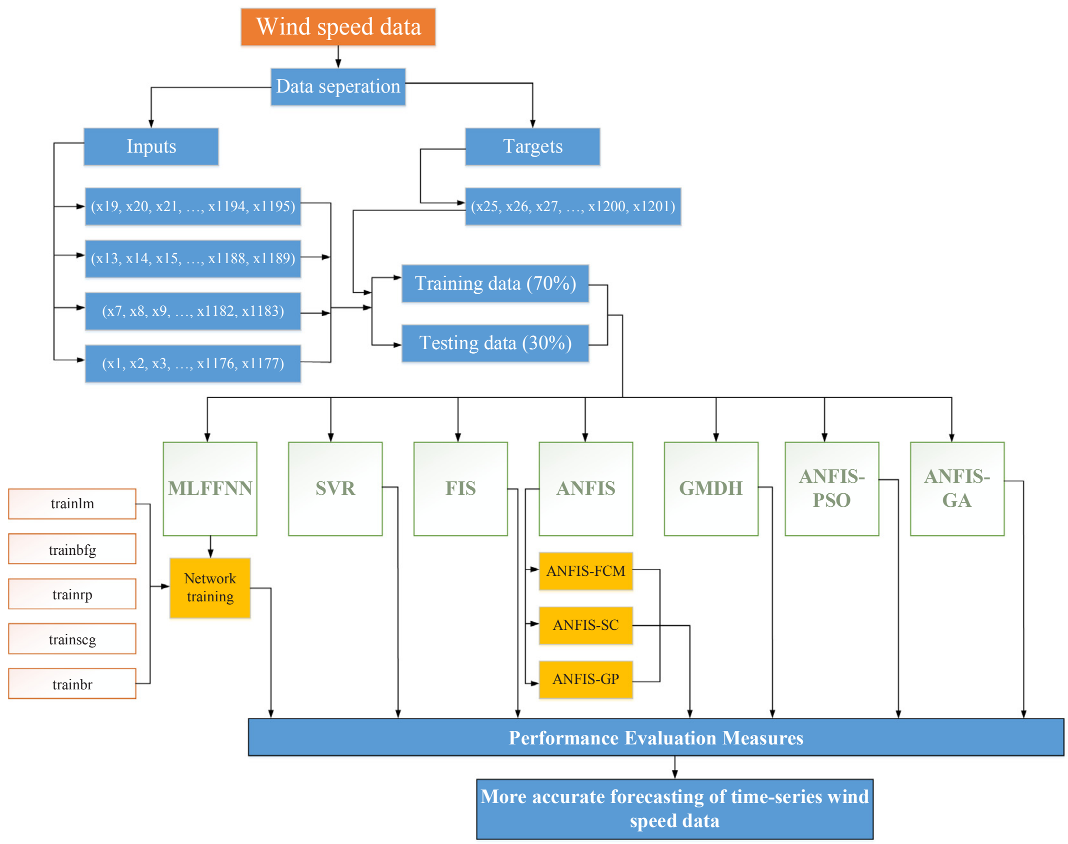


**Fig. 9.** Wind speed forecasting procedure.

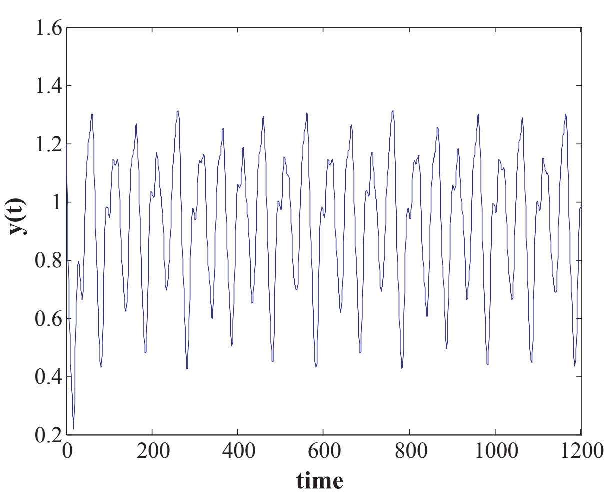


**Fig. 10.** Mackey-Glass series.

parameters (in Section 3.4). The membership functions are assumed Gaussian as in Eq. (10), and their parameters are $\{a_i,b_i,c_i\}$ ($a_i$ is the variance of membership functions, $b_i$ is a trainable parameter and $c_i$ is the center of membership functions). These parameters are premise parameters. The consequent parameters were given in Eq. (16) that are $\{p_i,q_i,r_i\}$. Therefore, there are three sets of trainable parameters in the premise parameters that each of them has N genes. Here, N demonstrates the number of membership functions. Moreover, the consequent parameters are trained during the optimization process. In conclusion part, each chromosome has $(I+1)\times R$ genes that $I$ denotes dimension of data inputs and $R$ is equal to the number of rules.

**Table 1**
Performances of the proposed models to predict Mackey-Glass series.

| | RMSE | R | MSE |
|---|---|---|---|
| MLFFNN | 0.0044 | 0.9997 | 2.280e−05 |
| SVR | 0.0125 | 0.9986 | 0.000157 |
| FIS | 0.0842 | 0.9610 | 0.007106 |
| ANFIS-FCM | 0.0035 | 0.9998 | 1.24e−05 |
| ANFIS-SC | 0.0026 | 0.9999 | 6.78e−06 |
| ANFIS-GP | 0.0009 | 0.9999 | 8.27e−07 |
| GMDH | 0.0406 | 0.9841 | 0.001650 |
| ANFIS-PSO | 0.0140 | 0.9980 | 0.000199 |
| ANFIS-GA | 0.0415 | 0.9849 | 0.001700 |

### *3.8. Group method of data handling (GMDH)*

GMDH type neural network uses mathematical functions to obtain the complex nonlinear relationships among the given inputs-output datasets. Indeed, this method is defined as a data-driven modeling technique. For each identification problem, the target is to find a function $\hat{f}$ that this function can be employed instead of the actual one, $f$ in order to predict of the output $\hat{y}$ for the considered input vector $X=(x_1,x_2,x_3,...,x_n)$ approximately close to the actual output y. Eq. (18) describes M observation of multi-input-single-output data pairs that is:

$$y_i=f(x_{i1},x_{i2},x_{i3},...,x_{in})\quad (i=1,2,...,M) \tag{18}$$

Now, by training a GMDH method the output values $\hat{y}_i$ can be predicted for each input vector $X=(x_{i1},x_{i2},x_{i3},...,x_{in})$, which is defined by the following equation:

$$\hat{y}_i=\hat{f}(x_{i1},x_{i2},x_{i3},...,x_{in})\quad (i=1,2,...,M) \tag{19}$$

The target is now to develop a GMDH method to minimize the square of difference between the predicted and actual data, which is

**Table 2**
MLFFNN to predict wind speed.

| Model | | Algorithm | RMSE (m/s) | | R | | MSE (m/s) | |
|---|---|---|---|---|---|---|---|---|
| | | | Train | Test | Train | Test | Train | Test |
| *5-min interval* | | | | | | | | |
| 1 | LM | trainlm | 0.0075 | 0.0120 | 0.9999 | 0.9999 | 5.659e − 05 | 1.442e − 04 |
| 2 | RP | trainrp | 0.0411 | 0.0437 | 0.9995 | 0.9995 | 0.0016 | 0.0019 |
| 3 | SCG | trainscg | 0.0470 | 0.0485 | 0.9994 | 0.9993 | 0.0022 | 0.0023 |
| 4 | CGP | traincgp | 0.0473 | 0.0020 | 0.9994 | 0.9994 | 0.0022 | 0.0020 |
| 5 | BR | trainbr | 0.0424 | 0.0424 | 0.9995 | 0.9995 | 0.0018 | 0.0018 |
| *10-min interval* | | | | | | | | |
| 1 | LM | trainlm | 0.0217 | 0.0388 | 0.9999 | 0.9996 | 0.0004 | 0.0015 |
| 2 | RP | trainrp | 0.1497 | 0.1552 | 0.9952 | 0.9949 | 0.0224 | 0.0241 |
| 3 | SCG | trainscg | 0.1008 | 0.1044 | 0.9978 | 0.9977 | 0.0101 | 0.0109 |
| 4 | CGP | traincgp | 0.1032 | 0.1103 | 0.9977 | 0.9974 | 0.0106 | 0.0121 |
| 5 | BR | trainbr | 0.1306 | 0.1332 | 0.9964 | 0.9962 | 0.0170 | 0.0177 |
| *15-min interval* | | | | | | | | |
| 1 | LM | trainlm | 0.0713 | 0.1116 | 0.9991 | 0.9978 | 0.0050 | 0.0124 |
| 2 | RP | trainrp | 0.2549 | 0.2961 | 0.9886 | 0.9851 | 0.0649 | 0.0877 |
| 3 | SCG | trainscg | 0.2436 | 0.2556 | 0.9896 | 0.9886 | 0.0593 | 0.06535 |
| 4 | CGP | traincgp | 0.3757 | 0.3924 | 0.9755 | 0.9750 | 0.1411 | 0.1539 |
| 5 | BR | trainbr | 0.2356 | 0.2440 | 0.9902 | 0.9896 | 0.0555 | 0.0595 |
| *30-min interval* | | | | | | | | |
| 1 | LM | trainlm | 0.2500 | 0.4129 | 0.9887 | 0.9689 | 0.0625 | 0.1705 |
| 2 | RP | trainrp | 0.464 | 0.6058 | 0.9606 | 0.9325 | 0.2153 | 0.3670 |
| 3 | SCG | trainscg | 0.6000 | 0.6692 | 0.9349 | 0.9191 | 0.3609 | 0.4478 |
| 4 | CGP | traincgp | 0.5090 | 0.5883 | 0.9524 | 0.9364 | 0.2591 | 0.3461 |
| 5 | BR | trainbr | 0.5879 | 0.6653 | 0.9358 | 0.9170 | 0.3457 | 0.4427 |

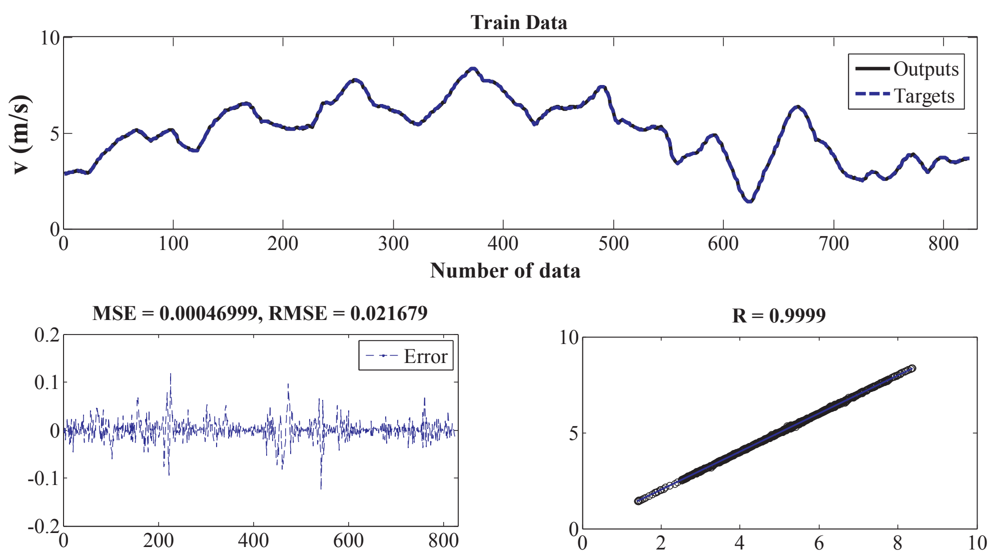


**Fig. 11.** Training phase of MLFFNN for 10-min interval (model No. 1).

given by:

$$\sum_{i=1}^{M} [\hat{f}\,(x_{i1},x_{i2},x_{i3},\ldots,x_{in})-y_i]^2 \rightarrow min \tag{20}$$

Here, Volterra functional series is employed to create a connection between the inputs and output, which is given by the following equation (this equation also is known as the Kolmogorov-Gabor polynomial):

$$y = a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_i x_j + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} a_{ijk}x_i x_j x_k + \cdots \tag{21}$$

A system of partial quadratic polynomials (consisting of two neurons) can be used to represent this Volterra functional series that is:

$$\hat{y} = G(x_i,x_j) = a_0 + a_1 x_i + a_2 x_j + a_3 x_i x_j + a_4 x_i^2 + a_5 x_j^2 \tag{22}$$

Regression method is employed to calculate the coefficient $a_i$ in which the difference between the predicted output $\hat{y}$ and the actual one y, is minimized for each pair of ($x_i$, $x_j$) as inputs variables. A quadratic form was explained in Eq. (22) that is used to construct a three of polynomials. In order to optimally fit the output in the whole set of input-output data pair, the coefficient of each quadratic function $G_i$ is obtained, which is defined by the following equation:

$$E = \frac{\sum_{i=1}^{M} (y_i - G_i)^2}{M} \rightarrow min \tag{23}$$

The regression polynomial in the form of Eq. (22) is generated by considering all possibilities of two independent variables out of total $n$ input variables (in the basic GMDH model). Respectively, $\binom{n}{2} = \frac{n(n-1)}{2}$

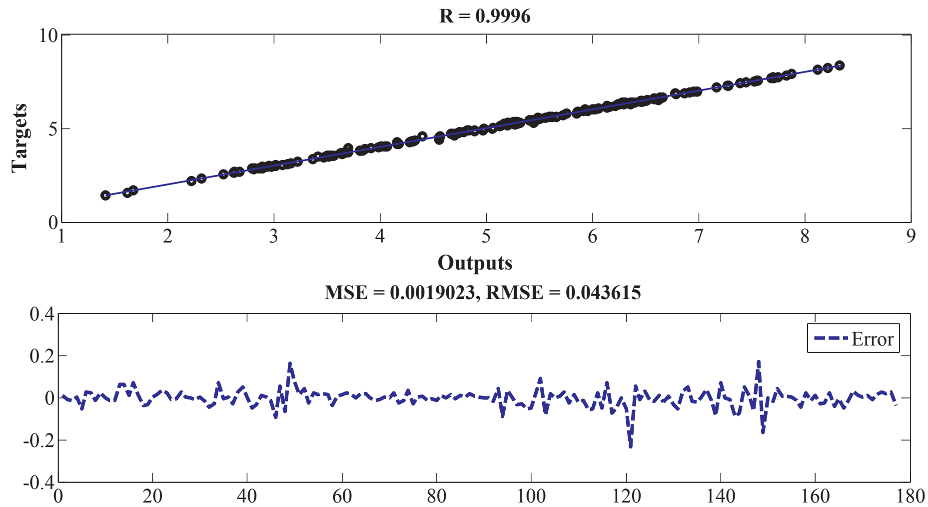


**Fig. 12.** Testing phase of MLFFNN for 10-min interval (model No. 1).

**Table 3**
The performance of SVR model.

| Model | | RMSE (m/s) | | R | | MSE (m/s) | |
|---|---|---|---|---|---|---|---|
| | | Train | Test | Train | Test | Train | Test |
| *5-min interval* | | | | | | | |
| 1 | ε = 1, C = 1, σ = 1 | 0.6079 | 0.6287 | 0.8802 | 0.7234 | 0.3695 | 0.3952 |
| 2 | ε = 0.5, C = 10, σ = 1 | 0.2639 | 0.2877 | 0.9786 | 0.9488 | 0.0696 | 0.0827 |
| 3 | ε = 0.2, C = 100, σ = 0.5 | 0.1176 | 0.1363 | 0.9958 | 0.9887 | 0.0138 | 0.0186 |
| 4 | ε = 0.2, C = 1000, σ = 0.5 | 0.1179 | 0.1372 | 0.9958 | 0.9886 | 0.0139 | 0.0188 |
| *10-min interval* | | | | | | | |
| 1 | ε = 1, C = 1, σ = 1 | 0.5453 | 0.5892 | 0.9006 | 0.8577 | 0.2974 | 0.3471 |
| 2 | ε = 0.5, C = 10, σ = 1 | 0.2583 | 0.3284 | 0.9786 | 0.9581 | 0.0667 | 0.1078 |
| 3 | ε = 0.2, C = 100, σ = 0.5 | 0.1262 | 0.1546 | 0.9949 | 0.9909 | 0.0159 | 0.0239 |
| 4 | ε = 0.2, C = 1000, σ = 0.5 | 0.1257 | 0.1547 | 0.9950 | 0.9908 | 0.0158 | 0.0239 |
| *15-min interval* | | | | | | | |
| 1 | ε = 1, C = 1, σ = 1 | 0.5572 | 0.4960 | 0.9316 | 0.8760 | 0.3105 | 0.2460 |
| 2 | ε = 0.5, C = 10, σ = 1 | 0.3145 | 0.2688 | 0.9787 | 0.9653 | 0.0989 | 0.0722 |
| 3 | ε = 0.2, C = 100, σ = 0.5 | 0.1469 | 0.1492 | 0.9954 | 0.9894 | 0.0216 | 0.0223 |
| 4 | ε = 0.2, C = 1000, σ = 0.5 | 0.1498 | 0.1470 | 0.9952 | 0.9897 | 0.0224 | 0.0216 |
| *30-min interval* | | | | | | | |
| 1 | ε = 1, C = 1, σ = 1 | 0.6568 | 0.6254 | 0.9223 | 0.9134 | 0.4314 | 0.3912 |
| 2 | ε = 0.5, C = 10, σ = 1 | 0.5020 | 0.3919 | 0.9554 | 0.9669 | 0.2520 | 0.1536 |
| 3 | ε = 0.2, C = 100, σ = 0.5 | 0.1969 | 0.1715 | 0.9933 | 0.9938 | 0.0388 | 0.0294 |
| 4 | ε = 0.2, C = 1000, σ = 0.5 | 0.1817 | 0.1733 | 0.9943 | 0.9936 | 0.0330 | 0.0300 |

neurons will be generated in the first hidden layer of the feed-forward network from the observations $\{(y_i,x_{ip},x_{iq});\ (i=1,2,\ldots,M)\}$ for different $p,q\in\{1,2,\ldots,n\}$. It is feasible to generate M data triples $\{(y_i,x_{ip},x_{iq});\ (i=1,2,\ldots,M)\}$ from observation using such $p,q\in\{1,2,\ldots,n\}$ in the form,

$$\begin{bmatrix} x_{1p} & x_{1q} & \vdots & y_1 \\ x_{2p} & x_{2q} & \vdots & y_2 \\ \cdots & \cdots & \cdots & \cdots \\ x_{Mp} & x_{Mq} & \vdots & y_M \end{bmatrix} \tag{24}$$

The matrix equation can be determined as $Aa=Y$ using the quadratic sub-expression in the form of Eq. (20) for each row of M data triples. In which $a$ is the vector of unknown coefficients of the quadratic polynomial in Eq. (22).

$$a=\{a_0,a_1,a_2,a_3,a_4,a_5\} \tag{25}$$

Also, $Y=\{y_1,y_2,y_3,\ldots,y_M\}^T$ is the vector of output's value from observation. Therefore

$$A=\begin{bmatrix} 1 & x_{1p} & x_{1q} & x_{1p}x_{1q} & x_{1p}^2 & x_{1q}^2 \\ 1 & x_{2p} & x_{2q} & x_{2p}x_{2q} & x_{2p}^2 & x_{2q}^2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_{Mp} & x_{Mq} & x_{Mp}x_{Mq} & x_{Mp}^2 & x_{Mq}^2 \end{bmatrix} \tag{26}$$

The least-squares technique from multiple-regression analysis leads to the solution of the normal equations in form of

$$a=(A^TA)^{-1}A^TY \tag{27}$$

This equation will be determined the vector of the best coefficient of the quadratic Eq. (22) for the all set of M data triples.

### 3.9. Wind speed prediction procedure

Fig. 9 illustrates the process of time-series wind speed prediction using MLAs. Wind speed data (1201 data is considered in this section) are divided into two sections as inputs and targets. For this investigation, the values of delays are defined to be {6, 12, 18 *and* 24}. By considering this delays for designing the networks, datasets for inputs data, as can be seen in Fig. 10, are divided into 4 samples. In addition, 70% of

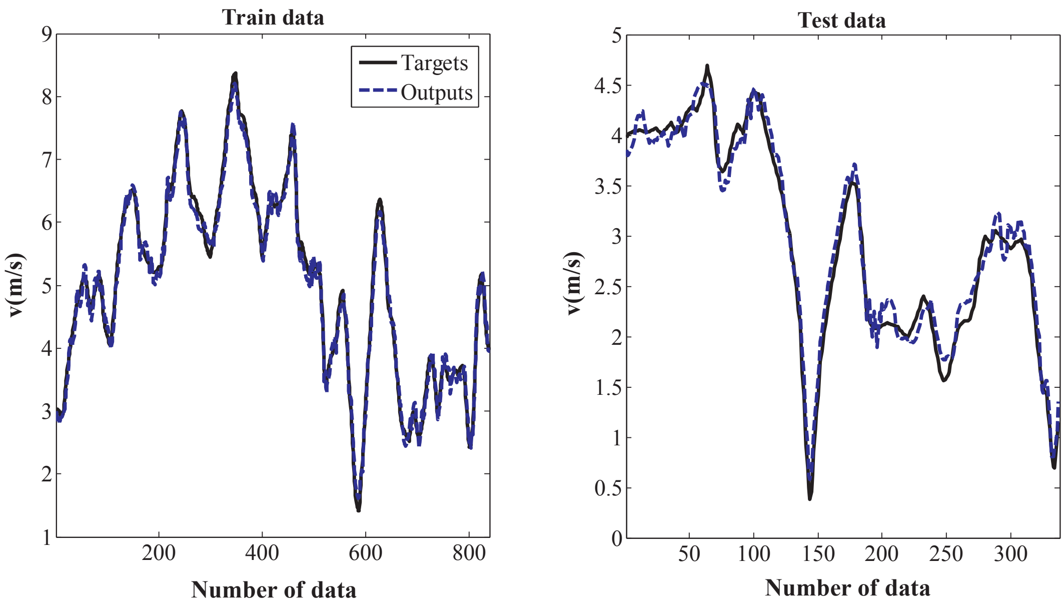


**Fig. 13.** Train and test datasets for SVR model (15-min, model No. 4).

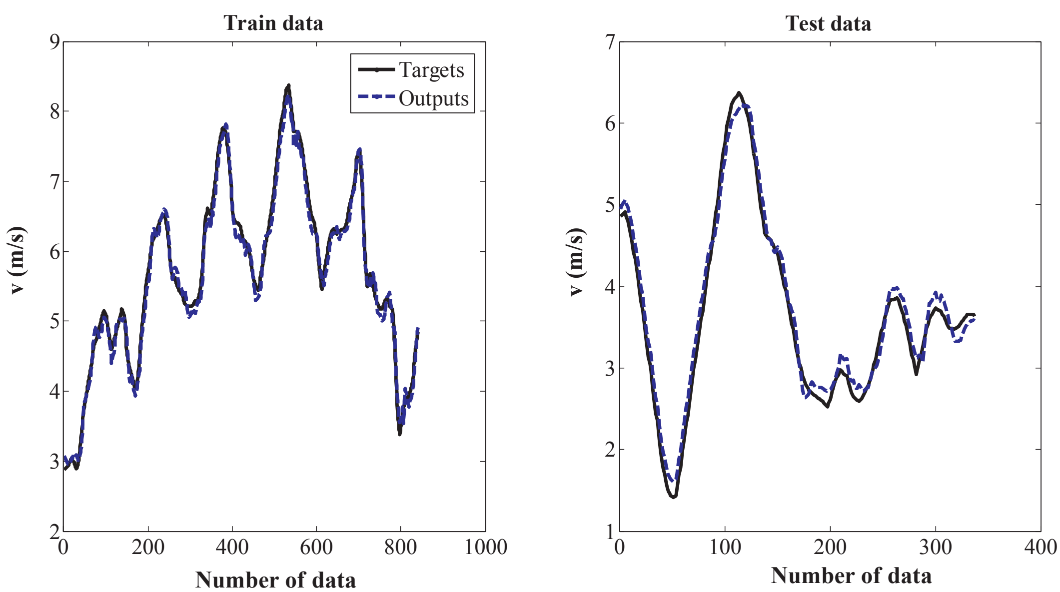


**Fig. 14.** Training and testing samples of SVR model (10-min, model No. 3).

the inputs and targets are determined as training data and 30% are as testing data. Therefore, for forecasting wind speed in the future, it is needed to have four measured values of wind speed as before described. This designing is for 6-step ahead forecasting. And for other time-step ahead forecasting, the value of delays will be changed. The developed models are trained and tested and the best performance of the models is introduced.

### 3.10. Performance evaluation measures

Several evaluation criteria are applied to evaluate the performance of the developed models in terms of forecast accuracy. To obtain the difference between the targets and outputs the mean square error (MSE) is a broadly utilized criterion. The lower values of MSE indicate the high performance of the predicted values.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2 \tag{28}$$

Also, the relationship between the outputs and targets is shown by the correlation coefficient (R). If R = 0, this illustrates that there is no linear relationship between outputs and targets and R = 1 demonstrates an exact linear relationship between outputs and targets.

$$R = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{29}$$

Root mean square error (RMSE) is the standard deviation of the prediction errors (targets-outputs).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2} \tag{30}$$

where $x_i$, $y_i$, $\bar{x}$, $\bar{y}$ and $n$ are observed value, predicted value, mean of observed data, mean of predicted data and number of data, respectively.

**Table 4**
Investigation on performance of the FIS model.

| Model | | RMSE (m/s) | | R | | MSE (m/s) | |
|---|---|---|---|---|---|---|---|
| | | Train | Test | Train | Test | Train | Test |
| *5-min interval* | | | | | | | |
| 1 | 3-2-3-2-3 | 0.5779 | 0.5199 | 0.9210 | 0.9169 | 0.3340 | 0.2703 |
| 2 | 3-3-3-3-3 | 0.5579 | 0.4622 | 0.9243 | 0.9392 | 0.3113 | 0.2137 |
| 3 | 4-4-4-4-4 | 0.3852 | 0.5477 | 0.9754 | 0.9339 | 0.1484 | 0.3000 |
| *10-min interval* | | | | | | | |
| 1 | 3-2-3-2-3 | 1.058 | 1.2395 | 0.8477 | 0.7553 | 1.1772 | 1.5364 |
| 2 | 3-3-3-3-3 | 1.0752 | 1.1937 | 0.8554 | 0.7596 | 1.1561 | 1.4250 |
| 3 | 4-4-4-4-4 | 0.9234 | 0.8527 | 0.9415 | 0.8409 | 1.064 | 1.1320 |
| *15-min interval* | | | | | | | |
| 1 | 3-2-3-2-3 | 1.3396 | 1.1362 | 0.7207 | 0.5046 | 1.7945 | 1.2910 |
| 2 | 3-3-3-3-3 | 1.1053 | 0.9001 | 0.8209 | 0.8162 | 1.2216 | 0.8102 |
| 3 | 4-4-4-4-4 | 0.8894 | 0.9951 | 0.9337 | 0.6586 | 0.7911 | 0.9902 |
| *30-min interval* | | | | | | | |
| 1 | 3-2-3-2-3 | 3.5150 | 3.3072 | 0.56161 | 0.3342 | 12.3550 | 10.9378 |
| 2 | 3-3-3-3-3 | 1.0853 | 1.1234 | 0.7739 | 0.7886 | 1.1778 | 1.2621 |
| 3 | 4-4-4-4-4 | 2.0538 | 2.0008 | 0.8613 | 0.8162 | 4.2182 | 4.0031 |

## 4. Result and discussion

### 4.1. Testing of the proposed models

A time-series of Mackey-Glass (Fig. 10) is used as an example to confirm the correct implementation of the proposed models [30]. This time-series is represented by Eq. (31).

$$x(t+1) = x(t) + \frac{bx(t-\tau)}{1+x^{c}(t-\tau)} - ax(t) \tag{31}$$

Three statistical parameters that are RMSE, R and MSE are considered to analyze the performance of the models. As can be seen in Table 1, the developed models for prediction of time-series data were successfully implemented. The developed models report a high performance for the predicted data in terms of RMSE, R and MSE. The experimental conditions of the models are as follows.GMDH: $N_l = 10$, $N_n = 10$, $S_p = 0.8$; MLFFNN: 250 neurons in the hidden layer, LM algorithm, *tansig* and *purelin* as transfer functions; FIS: 4 fuzzy rules for inputs and target; SVR: ε = 0.2, c = 1000, σ = 0.5; ANFIS-FCM: number of clusters = 10, partition matrix exponent = 2; ANFIS-SC: influence radius = 0.2; ANFIS-GP: number of membership functions: 2; ANFIS-PSO: number of clusters: 10; ANFIS-GA: number of clusters: 10.

### 4.2. MLFFNN model

A three-layer feed-forward neural network was implemented to predict the wind speed in the case study region. The developed network was trained with different data training algorithms. The best performance was obtained with 250 neurons in the hidden layer. Table 2 shows the results of MLFFNN technique for prediction of wind speed in four different lengths of time. As can be observed in the Table, LM training algorithm has the best performance to predict the time-series wind speed data in terms of RMSE, R and MSE. Generally, this model for all time intervals illustrates an acceptance prediction accuracy. For 5-min and 10-min intervals, all data training algorithms can predict the target with high accuracy. Investigation of the predicted wind speed data for 15-min interval shows the higher adequacy of LM training algorithm with RMSE = 0.1116 (m/s), R = 0.9978 and MSE = 0.0124 (m/s) (for the test data). For the four datasets, SCG and CGP show a lower accuracy for the outputs. In general, for four datasets, MLFFNN generates a correlation coefficient, which indicates the goodness of fit of the model, approximately above 0.97 for the test data. It is noteworthy to mention that to avoid the overfitting problem in this investigation, the data were divided into two sections. One sample was used in order to train the network and the second one was used for testing the network.

The training phase of MLFFNN with LM algorithm (for 10-min interval, model No. 1) along with error and correlation graphs of this process are shown in Fig. 11. In addition, the testing phase of wind speed prediction using MLFFNN is shown in Fig. 12. Both figures show the predicted wind speed data follow the actual data with a small disagreement.

### 4.3. SVR model

SVR model has three user-defined parameters that are ε, $C$ and σ. Table 3 presents the prediction accuracy with different user-defined parameters. To obtain the best configurations several trials have been carried out. The results indicate that the SVR model is a powerful tool to predict the time-series wind speed data. This model for 5-min, 10-min, 15-min and 30-min intervals reports a high prediction accuracy. Generally, for four time intervals the SVR model with ε = 0.2, σ = 0.5 and $C$ = 1000 performs better than the other configurations.

Moreover, Figs. 13 and 14 demonstrate training and testing datasets of SVR model for 15-min (model No. 4) and 10-min (model No. 3) intervals of wind speed data, respectively. The figures show the prior and predicted data agree well with each other. It can be concluded that SVR model seems to be a strong technique in predicting the wind speed data.

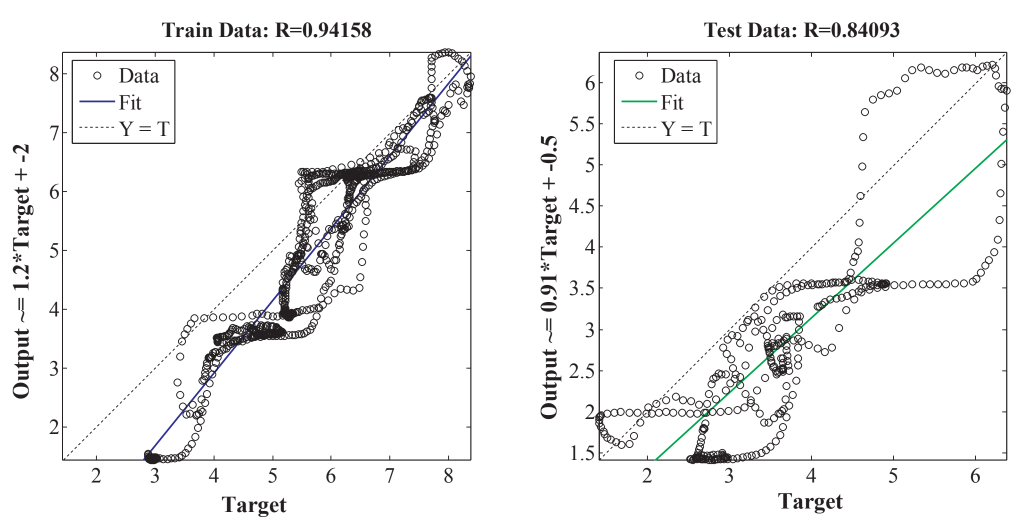


**Fig. 15.** Training and testing phases of FIS for 10-min time interval (model No. 3).

**Table 5**
Three types of ANFIS method.

| FCM model | Number of clusters | Partition matrix exponent | RMSE (m/s) Train | RMSE (m/s) Test | R Train | R Test | MSE (m/s) Train | MSE (m/s) Test |
|---|---|---|---|---|---|---|---|---|
| *5-min interval* | | | | | | | | |
| 1 | 5 | 2 | 0.031993 | 0.051061 | 0.9996 | 0.9984 | 0.0010 | 0.0026 |
| 2 | 10 | 2 | 0.027297 | 0.11688 | 0.9997 | 0.9939 | 0.0007 | 0.0136 |
| 3 | 15 | 4 | 0.018928 | 0.11508 | 0.9998 | 0.9923 | 0.0003 | 0.0132 |
| *10-min interval* | | | | | | | | |
| 1 | 5 | 2 | 0.1054 | 0.2100 | 0.9964 | 0.9865 | 0.0111 | 0.0441 |
| 2 | 10 | 2 | 0.0710 | 0.3418 | 0.9984 | 0.9660 | 0.0050 | 0.1168 |
| 3 | 15 | 4 | 0.0596 | 0.5359 | 0.9988 | 0.9010 | 0.0035 | 0.2865 |
| *15-min interval* | | | | | | | | |
| 1 | 5 | 2 | 0.2540 | 0.5122 | 0.9863 | 0.9198 | 0.0645 | 0.2623 |
| 2 | 10 | 2 | 0.1868 | 0.4376 | 0.9926 | 0.9189 | 0.0348 | 0.1914 |
| 3 | 15 | 4 | 0.1674 | 0.9331 | 0.9941 | 0.7655 | 0.0280 | 0.8706 |
| *30-min interval* | | | | | | | | |
| 1 | 5 | 2 | 0.5732 | 0.7466 | 0.9404 | 0.8765 | 0.3285 | 0.5574 |
| 2 | 10 | 2 | 0.5274 | 0.7363 | 0.9498 | 0.8810 | 0.2781 | 0.5421 |
| 3 | 15 | 4 | 0.4733 | 0.9730 | 0.9598 | 0.7789 | 0.2240 | 0.9467 |
| Sub-clustering model | Influence radius | | RMSE (m/s) Train | RMSE (m/s) Test | R Train | R Test | MSE (m/s) Train | MSE (m/s) Test |
| *5-min interval* | | | | | | | | |
| 1 | 0.1 | | 0.020445 | 0.090028 | 0.9998 | 0.9966 | 0.0004 | 0.0081 |
| 2 | 0.2 | | 0.028352 | 0.097064 | 0.9997 | 0.9959 | 0.0008 | 0.0094 |
| 3 | 0.6 | | 0.038284 | 0.052091 | 0.9995 | 0.9984 | 0.0014 | 0.0027 |
| *10-min interval* | | | | | | | | |
| 1 | 0.1 | | 0.0364 | 0.3571 | 0.9995 | 0.9613 | 0.0013 | 0.1275 |
| 2 | 0.2 | | 0.0539 | 0.3547 | 0.9990 | 0.9525 | 0.0029 | 0.1258 |
| 3 | 0.6 | | 0.1053 | 0.2456 | 0.9964 | 0.9788 | 0.0110 | 0.0603 |
| *15-min interval* | | | | | | | | |
| 1 | 0.1 | | 0.1028 | 1.0724 | 0.9977 | 0.5945 | 0.0105 | 1.1500 |
| 2 | 0.2 | | 0.1279 | 0.8506 | 0.9965 | 0.8276 | 0.0163 | 0.7235 |
| 3 | 0.6 | | 0.2712 | 0.4345 | 0.9844 | 0.92868 | 0.0735 | 0.1887 |
| *30-min interval* | | | | | | | | |
| 1 | 0.2 | | 0.3744 | 1.2074 | 0.9750 | 0.7117 | 0.1401 | 1.4578 |
| 2 | 0.4 | | 0.5625 | 0.7286 | 0.9427 | 0.8868 | 0.3164 | 0.5308 |
| 3 | 0.6 | | 0.6571 | 0.7068 | 0.9209 | 0.8907 | 0.4317 | 0.4995 |
| Grid-part model | Number of MFs | | RMSE (m/s) Train | RMSE (m/s) Test | R Train | R Test | MSE (m/s) Train | MSE (m/s) Test |
| *5-min interval* | | | | | | | | |
| 1 | 2 | | 0.029912 | 0.055398 | 0.9997 | 0.9985 | 0.0008 | 0.0030 |
| 2 | 3 | | 0.018971 | 0.11284 | 0.9998 | 0.9952 | 0.0003 | 0.0127 |
| 3 | 4 | | 0.0098605 | 0.59829 | 0.9999 | 0.8027 | 9.7e−05 | 0.3579 |
| *10-min interval* | | | | | | | | |
| 1 | 2 | | 0.0764 | 0.3285 | 0.9981 | 0.9658 | 0.0058 | 0.1079 |
| 2 | 3 | | 0.0398 | 1.1438 | 0.9994 | 0.6279 | 0.0015 | 1.3082 |
| 3 | 4 | | 0.0224 | 2.3649 | 0.9998 | 0.2901 | 0.0050 | 5.5927 |
| *15-min interval* | | | | | | | | |
| 1 | 2 | | 0.1962 | 0.5952 | 0.9919 | 0.8669 | 0.038 | 0.3542 |
| 2 | 3 | | 0.0962 | 1.484 | 0.9982 | 0.4936 | 9.25E−3 | 2.2022 |
| 3 | 4 | | 0.0542 | 1.4776 | 0.9993 | 0.6804 | 2.93E−3 | 2.1833 |
| *30-min interval* | | | | | | | | |
| 1 | 2 | | 0.5313 | 0.8391 | 0.9490 | 0.8484 | 0.2822 | 0.7040 |
| 2 | 3 | | 0.2599 | 3.2494 | 0.9880 | 0.278 | 0.0675 | 10.5586 |
| 3 | 4 | | 0.1654 | 4.6133 | 0.9951 | 0.0459 | 0.0273 | 21.2825 |

### 4.4. FIS model

Fuzzy inference system was developed to estimate the time-series wind speed data. Table 4 shows the performance of FIS model with different number of fuzzy rules. For 5-min interval, the model No. 3 with 4 fuzzy rules for each input and output show the maximum prediction accuracy. For 10-min interval, it is observed that the prediction of wind speed with FIS model approximately agrees well with the target data and the value of correlation coefficient is 0.9415 for train data and 0.8409 for test data (model No. 3). Also, for 15-min and 30-min intervals, the FIS (model No. 2) and FIS (model No. 3) performs better than other FIS models. For this analysis, membership functions were determined as *gaussmf* for the inputs and output, and the Mamdani model was selected as FIS structure. For 10-min time interval, the

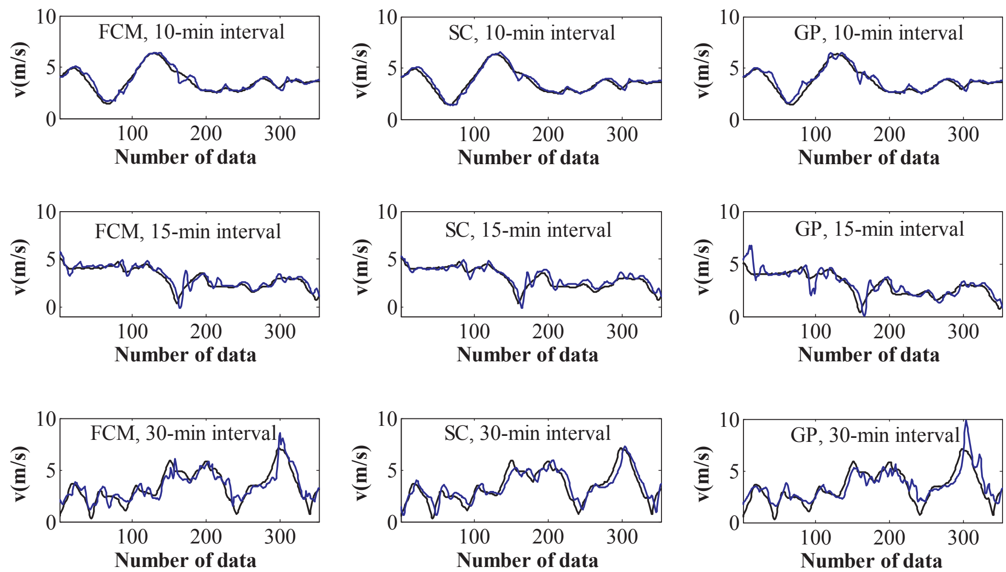


**Fig. 16.** Test data for ANFIS models (10-min, 15-min and 30-min intervals).

**Table 6**
Performance evaluation of ANFIS-PSO model for prediction of wind speed data.

| Model | No. of population | RMSE (m/s) | | R | | MSE (m/s) | |
|---|---|---|---|---|---|---|---|
| | | Train | Test | Train | Test | Train | Test |
| *5-min interval* | | | | | | | |
| 1 | 50 | 0.0361 | 0.0408 | 0.9997 | 0.9996 | 0.0013 | 0.0016 |
| 2 | 100 | 0.0362 | 0.0375 | 0.9997 | 0.9996 | 0.0013 | 0.0014 |
| 3 | 200 | 0.0337 | 0.0339 | 0.9997 | 0.9997 | 0.0011 | 0.0011 |
| 4 | 500 | 0.0329 | 0.0374 | 0.9997 | 0.9996 | 0.0010 | 0.0014 |
| 5 | 1000 | 0.0326 | 0.0364 | 0.9997 | 0.9997 | 0.0010 | 0.0013 |
| *10-min interval* | | | | | | | |
| 1 | 50 | 0.1258 | 0.126 | 0.9968 | 0.9964 | 0.0158 | 0.0158 |
| 2 | 100 | 0.1191 | 0.1356 | 0.9970 | 0.9961 | 0.0142 | 0.0184 |
| 3 | 200 | 0.1083 | 0.1192 | 0.9976 | 0.9969 | 0.0117 | 0.0142 |
| 4 | 500 | 0.1054 | 0.1122 | 0.9977 | 0.9974 | 0.0111 | 0.0126 |
| 5 | 1000 | 0.1133 | 0.1064 | 0.9972 | 0.9978 | 0.0128 | 0.0113 |
| *15-min interval* | | | | | | | |
| 1 | 50 | 0.2744 | 0.2799 | 0.9867 | 0.9863 | 0.0753 | 0.0783 |
| 2 | 100 | 0.2587 | 0.279 | 0.9886 | 0.9855 | 0.0669 | 0.0778 |
| 3 | 200 | 0.2656 | 0.2797 | 0.9877 | 0.9863 | 0.0705 | 0.0782 |
| 4 | 500 | 0.2514 | 0.2548 | 0.9890 | 0.9886 | 0.0632 | 0.0649 |
| 5 | 1000 | 0.2365 | 0.2767 | 0.9904 | 0.9866 | 0.0559 | 0.0765 |
| *30-min interval* | | | | | | | |
| 1 | 50 | 0.5838 | 0.6422 | 0.9356 | 0.9263 | 0.3409 | 0.4125 |
| 2 | 100 | 0.5595 | 0.6490 | 0.9412 | 0.9233 | 0.3130 | 0.4212 |
| 3 | 200 | 0.5731 | 0.6001 | 0.9392 | 0.9341 | 0.3248 | 0.3601 |
| 4 | 500 | 0.6305 | 0.6955 | 0.9279 | 0.9028 | 0.3976 | 0.4838 |
| 5 | 1000 | 0.5428 | 0.6451 | 0.9443 | 0.9265 | 0.2946 | 0.4162 |

training and testing datasets are shown in Fig. 15.

### 4.5. ANFIS model

In this study, three types of ANFIS model were implemented to estimate the time-series wind speed data. The first one is ANFIS-FCM that uses fuzzy c-means clustering to create a FIS. Obtaining the optimum performance of the model is determined by finding the optimum number of clusters and partition matrix exponent for this model. Therefore, a large number of trials have been done to find the optimum values of these parameters. Table 5 shows the performance prediction of the ANFIS-FCM model with different design parameters. This table illustrates the results of this investigation in which R, RMSE and MSE were used for evaluating the outputs (for train and test data). For 5-min and 10-min intervals, ANFIS-FCM (model No. 1) has the best performance by 5 clusters and partition matrix exponent of 2 (this parameter controls the amount of fuzzy overlap between clusters that larger values demonstrate a greater degree of overlap). Moreover, the input and output membership functions were considered *gaussmf* and *linear*, respectively. The second method of ANFIS that was implemented in this study is ANFIS-SC which uses subtractive clustering to create a Sugeno-type FIS structure. For this model, the main design parameter is the influence radius (radii). This parameter is defined as a vector that specified a clusters center's range of influence in each of data dimensions. Table 5 demonstrates the performance accuracy of the ANFIS-SC with different values of the influence radius. Regarding the table, the ANFIS-SC model No. 3 can predict the targets with high accuracy in terms of R, RMSE and MSE. The third one is ANFIS-GP that generates a fuzzy system using grid partitioning on data. For this model, the input and output membership functions were selected as *gaussmf* and *linear*, respectively. Also, the model evaluated with different number of membership functions that the results were shown in Table 5. As can be observed in the table, the ANFIS-GP model No. 1 (number of membership functions: 2) outperforms the other models.

Fig. 16 demonstrates the testing datasets and provides a term-by-term comparison of the predicted wind speed data via the proposed ANFIS models. As can be seen, for 10-min interval, the predicted data pattern for ANFIS-FCM (model No. 1), ANFIS-SC (model No. 3) and ANFIS-GP (model No. 1) are observed to follow the actual data pattern closely, with little disagreement. Also, for 15-min interval, the best performance is obtained with ANFIS-SC (model No. 3). Fig. 16 shows ANFIS-SC (model No. 3) performs better than ANFIS-FCM and ANFIS-GP for 30-min interval.

### 4.6. ANFIS-PSO model

Three types of ANFIS models were implemented to forecast the time-series wind speed data. The results have shown that the ANFIS-FCM outperformed the ANFIS-SC and ANFIS-GP. Particle swarm optimization was employed to optimize the prediction accuracy of the ANFIS model (here, ANFIS-FCM). Table 6 presents the result of this hybrid model for different time intervals. This analysis was carried out with 10 clusters. Also, the PSO parameters were: the personal learning coefficient: 1, the maximum iteration: 1000, the global learning coefficient: 2, and Inertia weight: 1. Table 6 shows the forecast accuracy of the ANFIS-PSO for different population. For 5-min and 10-min

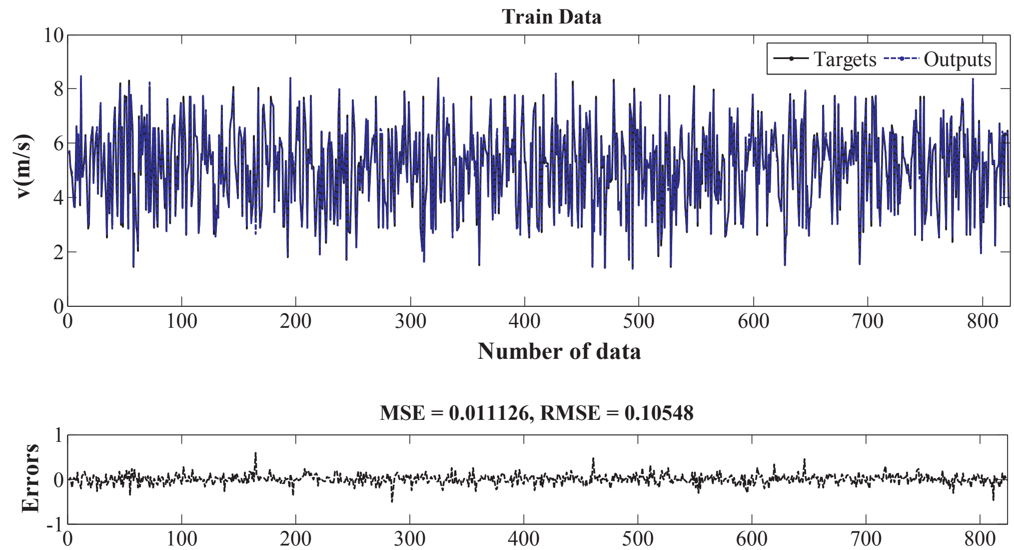


**Fig. 17.** Training phase of the ANFIS-PSO model for 10-min interval.

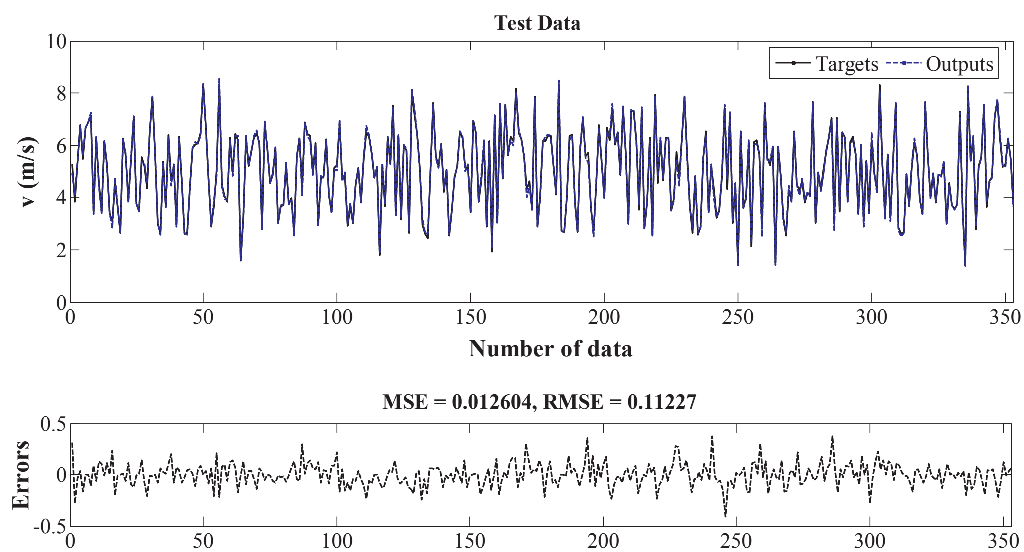


**Fig. 18.** Testing phase of the ANFIS-PSO model for 10-min interval.

intervals, the model No. 5 has more performance with number of population of 1000 (for testing datasets the statistical indicators were obtained as RMSE = 0.0364 (m/s), R = 0.9997 and MSE = 0.0013 (m/s) (for 5-min interval), and RMSE = 0.1064 (m/s), R = 0.9978 and MSE = 0.0113 (for 10-min interval)). For 15-min interval, the ANFIS-PSO model No. 4 shows more prediction accuracy according to the statistical indices. For 30-min interval, model No. 3 performs better than the other models. Regarding the table, it can be found that the PSO algorithm can successfully increase the performance of the ANFIS model. This accomplishment is significant for 15-min and 30-min intervals. Figs. 17 and 18 demonstrate the training and testing phases of the ANFIS-PSO model, respectively. As can be seen in the graphs, the predicted data by the ANFIS-PSO model can successfully follow the actual data for training and testing datasets.

### 4.7. ANFIS-GA model

Genetic algorithm was applied to optimize the prediction accuracy of the ANFIS model. Table 7 shows the performance of the ANFIS-GA model with the different population size of the genetic algorithm. According to the table, for 5-min and 10-min intervals, respectively the ANFIS-GA model Nos. 5 and 3 have the maximum performance based on the statistical indicators. For 15-min interval, model No. 1 with RMSE = 0.3007 (m/s), R = 0.9851 and MSE = 0.0904 (for test data) outperforms the other models. Also, for 30-min interval, model No. 3 with number of population of 200 shows better performance compared to the other models. For this analysis, the number of clusters was selected to be 10. Also, the GA parameters were as follows: maximum iteration was determined as 1000, crossover percentage was 0.7, mutation percentage was 0.5 and mutation rate was 0.1.

Figs. 19 and 20 illustrate the train and test datasets of the ANFIS-GA model for 10-min interval (model No. 3 in Table 7). Obviously, it is observed that the prediction of wind speed data (output data) with the ANFIS-GA model agrees well with the target data and the values of statistical indicators show the magnitude of the disagreement between the predicted and actual data is small.

### 4.8. GMDH model

In this study, GMDH type neural network was developed in order to predict wind speed in the case study wind farm. The network was trained with the time-series wind speed dataset of 840 samples with a 5-

**Table 7**
Performance evaluation of the ANFIS-GA model to forecast the wind speed data.

| Model | No. of population | RMSE (m/s) | | R | | MSE (m/s) | |
|---|---|---|---|---|---|---|---|
| | | Train | Test | Train | Test | Train | Test |
| *5-min interval* | | | | | | | |
| 1 | 50 | 0.0425 | 0.0432 | 0.9995 | 0.9995 | 0.0018 | 0.0018 |
| 2 | 100 | 0.0418 | 0.0410 | 0.9996 | 0.9995 | 0.0017 | 0.0016 |
| 3 | 200 | 0.0399 | 0.0425 | 0.9996 | 0.9995 | 0.0015 | 0.0018 |
| 4 | 500 | 0.0401 | 0.0401 | 0.9996 | 0.9996 | 0.0016 | 0.0017 |
| 5 | 1000 | 0.0413 | 0.0380 | 0.9996 | 0.9997 | 0.0017 | 0.0014 |
| *10-min interval* | | | | | | | |
| 1 | 50 | 0.1322 | 0.1478 | 0.9965 | 0.9952 | 0.0174 | 0.0218 |
| 2 | 100 | 0.1322 | 0.1409 | 0.9964 | 0.9958 | 0.0175 | 0.0198 |
| 3 | 200 | 0.1327 | 0.1242 | 0.9962 | 0.9969 | 0.0176 | 0.0154 |
| 4 | 500 | 0.1327 | 0.1367 | 0.9963 | 0.9960 | 0.0176 | 0.0186 |
| 5 | 1000 | 0.1295 | 0.1520 | 0.9965 | 0.9951 | 0.0167 | 0.0231 |
| *15-min interval* | | | | | | | |
| 1 | 50 | 0.3089 | 0.3007 | 0.9831 | 0.9851 | 0.0954 | 0.0904 |
| 2 | 100 | 0.2820 | 0.3288 | 0.9866 | 0.9791 | 0.0795 | 0.1081 |
| 3 | 200 | 0.2926 | 0.3164 | 0.9856 | 0.9811 | 0.0856 | 0.1001 |
| 4 | 500 | 0.2954 | 0.3130 | 0.9848 | 0.9826 | 0.0872 | 0.0979 |
| 5 | 1000 | 0.2895 | 0.3242 | 0.9850 | 0.9825 | 0.0838 | 0.1051 |
| *30-min interval* | | | | | | | |
| 1 | 50 | 0.6267 | 0.6441 | 0.9287 | 0.9172 | 0.3928 | 0.4149 |
| 2 | 100 | 0.6505 | 0.6594 | 0.9199 | 0.9218 | 0.4232 | 0.4348 |
| 3 | 200 | 0.6589 | 0.6487 | 0.9174 | 0.9308 | 0.4342 | 0.4208 |
| 4 | 500 | 0.6474 | 0.6181 | 0.9216 | 0.9298 | 0.4191 | 0.3821 |
| 5 | 1000 | 0.6255 | 0.6656 | 0.9280 | 0.9159 | 0.3913 | 0.4430 |

min, 10-min, 15-min and 30-min intervals. Also, to avoid overfitting problem 361 samples are used for testing the network. Table 8 illustrates three different designer-determined structures of GMDH type neural network that are selection pressure ($S_p$), maximum number of neurons in a layer ($N_n$) and maximum number of layers ($N_l$). In this table, it can be seen, for 5-min and 10-min intervals, the best performance was obtained for GMDH (model No. 4) with $N_l$, $N_n$ and $s_p$ equal to 10, 15 and 0.6, respectively. Also, for 15-min interval, the results show the error between prediction value and measure value is very small (this conclusion is seen in lower values of RMSE and MSE for the train and test data). For 30-min dataset, the correlation coefficient decreases about 4.5% compared to 10-min interval dataset.

For 10-min and 15 min time intervals, GMDH model No. 4; for 30-min dataset, GMDH model No. 3 were selected and plotted in Fig. 21. This graph shows the predicted data are in an excellent agreement with the actual data.

### 4.9. Comparison of the models

The obtained performances of the developed models during training and testing datasets were tabulated in Table 9. For 5-min interval, all the developed models can successfully predict the time-series wind speed data. MLFFNN (model No. 1), ANFIS-FCM (model No. 1), ANFIS-SC (model No. 3), ANFIS-GP (model No. 1), GMDH (model No. 2), ANFIS-PSO (model No. 3) and ANFIS-GA (model No. 5) report a correlation coefficient close to 1. For 10-min interval, it is concluded that MLFFNN (model No. 1), SVR (model No. 3), ANFIS-PSO (model No. 3), ANFIS-GA (model No. 3) and GMDH (model No. 4) outperform the other proposed models with the correlation coefficient close to 1. Also, ANFIS-FCM (model No. 1) shows the better performance than ANFIS-SC and ANFIS-GP. For 15-min interval, MLFFNN (model No. 1) and GMDH (model No. 4) generate a higher performance with the minimum RMSE and MSE for the training and testing samples. In addition, SVR (model No. 4), ANFIS-PSO (model No. 4) and ANFIS-GA (model No. 1) report satisfactory outputs with R of 0.9897, 0.9886 and 0.9851 (for the test data), respectively. For 30-min interval, SVR (model No. 3) has the maximum R and minimum RMSE and MSE. In addition, MLFFNN and GMDH type neural network can predict the targets with a high accuracy. As can be observed in the table, the PSO and GA algorithms successfully increase the performance of the ANFIS model in different time intervals.

## 5. Conclusion remarks

Wind energy is proposed as an important source of alternative energy in recent years. It has more advantages with respect to other sources in terms of installation and generation cost. Moreover, wind conditions are characterized as strong and stable. Brazil can be considered the most promising market for wind energy in Latin America, with an estimated wind potential of 300 GW and energy demand expected to increase by 2 GW per year until 2020. In this study, to predict the wind speed in Osorio wind farm (Brazil), MLAs were developed. The networks were applied to estimate wind speed for 5-min, 10-min, 15-min and 30-min time intervals. The main conclusions of this study are:

1. MLFFNN was implemented to predict the wind speed in Osorio wind farm. The maximum performance was obtained for 250 neurons in the hidden layer. Five data training algorithms (LM, RP, SCG, CGP

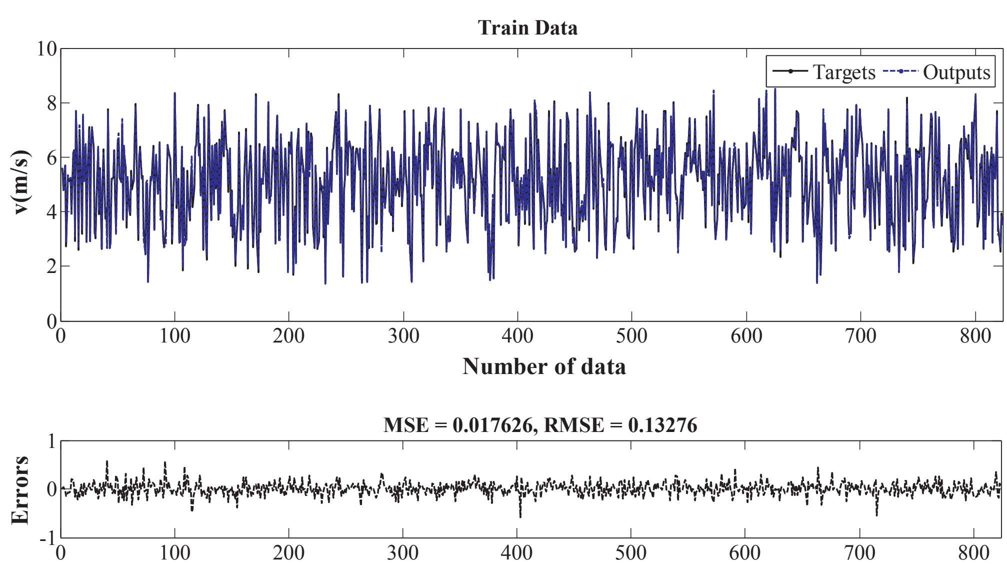


**Fig. 19.** Train data of the ANFIS-GA model for 10-min interval.

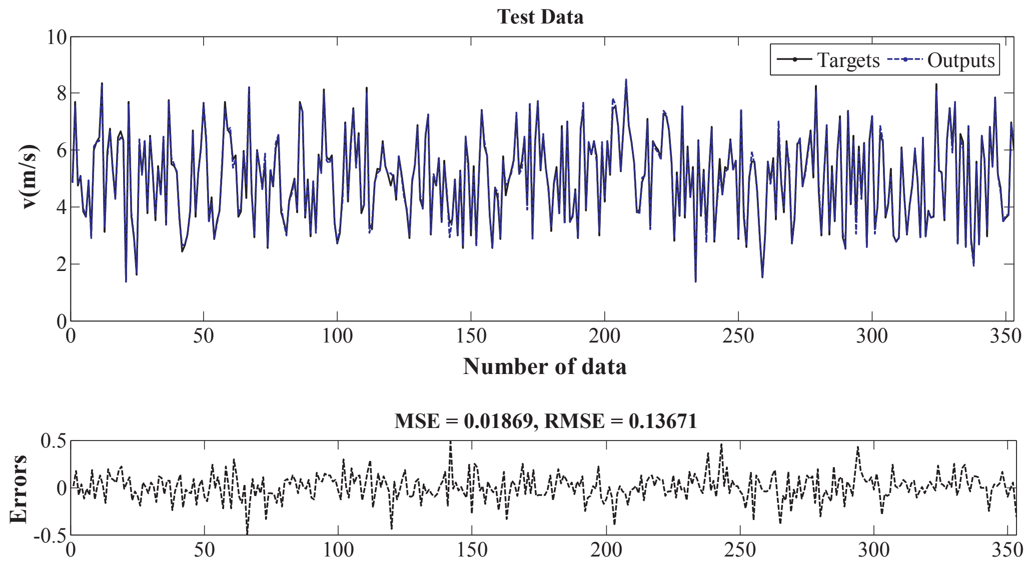


**Fig. 20.** Test data of the ANFIS-GA model for 10-min interval.

**Table 8**
GMDH type neural network with different design parameters.

| Model | | RMSE (m/s) | | R | | MSE (m/s) | |
|---|---|---|---|---|---|---|---|
| | | Train | Test | Train | Test | Train | Test |
| *5-min interval* | | | | | | | |
| 1 | $N_l = 5, N_n = 5, S_p = 0.8$ | 0.0441 | 0.0413 | 0.9994 | 0.9995 | 0.0019 | 0.0017 |
| 2 | $N_l = 10, N_n = 10, S_p = 0.8$ | 0.0444 | 0.0412 | 0.9994 | 0.9995 | 0.0019 | 0.0016 |
| 3 | $N_l = 10, N_n = 10, S_p = 0.6$ | 0.0422 | 0.0456 | 0.9995 | 0.9994 | 0.0017 | 0.0020 |
| 4 | $N_l = 10, N_n = 15, S_p = 0.6$ | 0.0433 | 0.0439 | 0.9995 | 0.9994 | 0.0018 | 0.0019 |
| *10-min interval* | | | | | | | |
| 1 | $N_l = 5, N_n = 5, S_p = 0.8$ | 0.1144 | 0.1121 | 0.9972 | 0.9973 | 0.0130 | 0.0125 |
| 2 | $N_l = 10, N_n = 10, S_p = 0.8$ | 0.1083 | 0.1150 | 0.9975 | 0.9972 | 0.0117 | 0.0132 |
| 3 | $N_l = 10, N_n = 10, S_p = 0.6$ | 0.1023 | 0.1093 | 0.9978 | 0.9974 | 0.0104 | 0.0119 |
| 4 | $N_l = 10, N_n = 15, S_p = 0.6$ | 0.1022 | 0.1114 | 0.9977 | 0.9975 | 0.0104 | 0.0124 |
| *15-min interval* | | | | | | | |
| 1 | $N_l = 5, N_n = 5, S_p = 0.8$ | 0.2499 | 0.2393 | 0.9894 | 0.9893 | 0.0624 | 0.0572 |
| 2 | $N_l = 10, N_n = 10, S_p = 0.8$ | 0.2483 | 0.2640 | 0.9887 | 0.9890 | 0.0616 | 0.0697 |
| 3 | $N_l = 10, N_n = 10, S_p = 0.6$ | 0.2472 | 0.2378 | 0.9894 | 0.9902 | 0.06114 | 0.0565 |
| 4 | $N_l = 10, N_n = 15, S_p = 0.6$ | 0.2466 | 0.2301 | 0.9894 | 0.99063 | 0.0608 | 0.0529 |
| *30-min interval* | | | | | | | |
| 1 | $N_l = 5, N_n = 5, S_p = 0.8$ | 0.5733 | 0.5507 | 0.9394 | 0.9442 | 0.3287 | 0.3033 |
| 2 | $N_l = 10, N_n = 10, S_p = 0.8$ | 0.5638 | 0.5490 | 0.9412 | 0.9451 | 0.3179 | 0.3014 |
| 3 | $N_l = 10, N_n = 10, S_p = 0.6$ | 0.5880 | 0.5074 | 0.9366 | 0.9522 | 0.3457 | 0.2575 |
| 4 | $N_l = 20, N_n = 10, S_p = 0.6$ | 0.5596 | 0.5718 | 0.9432 | 0.9381 | 0.3131 | 0.3270 |

and BR) were used in order to find the better one. The results demonstrated that the LM algorithm generated the smallest RMSE and MSE for predicted wind speed in the four time series samples. The developed model can successfully predict outputs with the correlation coefficient of 0.9999 (for 5-min), 0.9996 (for 10-min), 0.9978 (for 15-min) and 0.9689 (for 30-min) for the test data.

2. SVR technique was developed to predict the wind speed in the study zone. The developed model was shown a high accuracy with a high correlation between the targets and outputs as 0.9887 (for 5-min), 0.9909 (for 10-min), 0.9897 (for 15-min) and 0.9938 (for 30-min) for the test data.
3. The maximum performance of FIS technique was obtained with 4 membership functions (for 5-min 10-min and 30-min intervals) and 3 membership functions (for 15-min dataset) for the inputs and output. FIS technique has shown lower performance than other developed models for prediction of wind speed. But this method reported satisfactory outputs with R of 0.9339 (for 5-min), 0.8409 (for10-min) and 0.8162 (for 15-min and 30-min datasets).
4. Three types of ANFIS technique (ANFIS-FCM, ANFIS-SC and ANFIS-GP) were developed for prediction of the targets. In general, ANFIS technique outperforms the FIS method. For 5-min interval, the correlation coefficients for three types of the ANFIS model are approximately equal. For 10-min interval, ANFIS models were shown the high accuracy in term of R with 0.9865 (ANFIS-FCM), 0.9788 (ANFIS-SC) and 0.9658 (ANFIS-GP) for the test data. The predicted values of ANFIS-FCM and ANFIS-SC (for 15-min dataset) were seen to follow the measured values relatively well, with low magnitudes of disagreement. For 30-min interval dataset of wind speed, the correlation coefficient of ANFIS-SC, which indicates the goodness of fit of the model to be 0.8907.
5. The predicted data pattern of GMDH type neural network was observed to follow the actual data pattern closely, with little

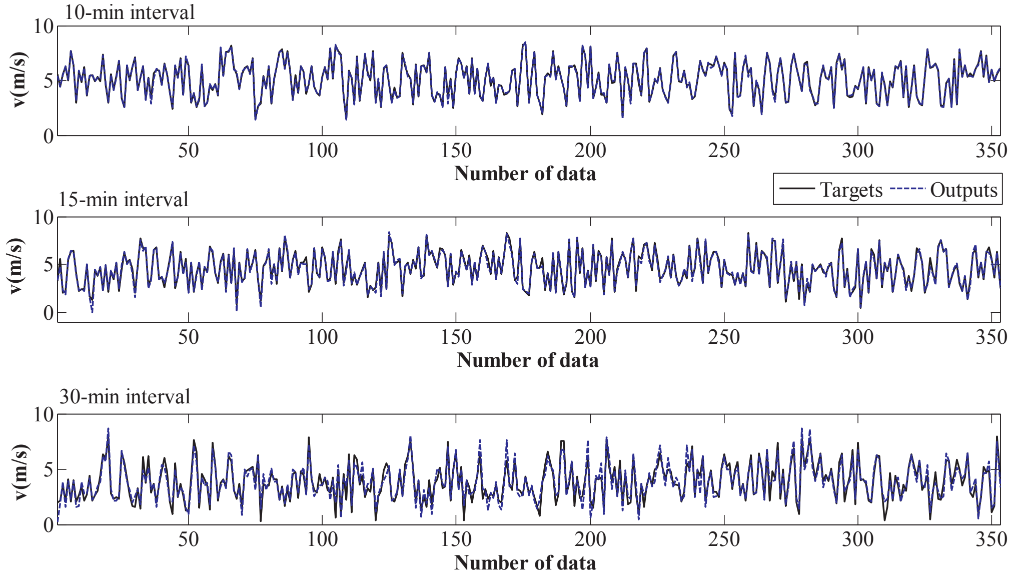


**Fig. 21.** Test data for GMDH in the three different time interval.

**Table 9**
A comparison between the proposed models.

| Method | Model | RMSE (m/s) | | R | | MSE (m/s) | |
|---|---|---|---|---|---|---|---|
| | | Train | Test | Train | Test | Train | Test |
| *5-min interval* | | | | | | | |
| MLFFNN | 1 | 0.0075 | 0.0120 | 0.9999 | 0.9999 | 5.659e − 05 | 1.442e − 04 |
| SVR | 3 | 0.1176 | 0.1363 | 0.9958 | 0.9887 | 0.0138 | 0.0186 |
| FIS | 3 | 0.3852 | 0.5477 | 0.9754 | 0.9339 | 0.1484 | 0.3000 |
| ANFIS-FCM | 1 | 0.031993 | 0.051061 | 0.9996 | 0.9984 | 0.0010 | 0.0026 |
| ANFIS-SC | 3 | 0.038284 | 0.052091 | 0.9995 | 0.9984 | 0.0014 | 0.0027 |
| ANFIS-GP | 1 | 0.029912 | 0.055398 | 0.9997 | 0.9985 | 0.0008 | 0.0030 |
| GMDH | 2 | 0.0444 | 0.0412 | 0.9994 | 0.9995 | 0.0019 | 0.0016 |
| ANFIS-PSO | 3 | 0.0337 | 0.0339 | 0.9997 | 0.9997 | 0.0011 | 0.0011 |
| ANFIS-GA | 5 | 0.0413 | 0.0380 | 0.9996 | 0.9997 | 0.0017 | 0.0014 |
| *10-min interval* | | | | | | | |
| MLFFNN | 1 | 0.0217 | 0.0388 | 0.9999 | 0.9996 | 0.0004 | 0.0015 |
| SVR | 3 | 0.1262 | 0.1546 | 0.9949 | 0.9909 | 0.0159 | 0.0239 |
| FIS | 3 | 0.9234 | 0.8527 | 0.9415 | 0.8409 | 1.064 | 1.1320 |
| ANFIS-FCM | 1 | 0.1054 | 0.2100 | 0.9964 | 0.9865 | 0.0111 | 0.0441 |
| ANFIS-SC | 3 | 0.1053 | 0.2456 | 0.9964 | 0.9788 | 0.0110 | 0.0603 |
| ANFIS-GP | 1 | 0.0764 | 0.3285 | 0.9981 | 0.9658 | 0.0058 | 0.1079 |
| GMDH | 4 | 0.1022 | 0.1114 | 0.9977 | 0.9975 | 0.0104 | 0.0124 |
| ANFIS-PSO | 3 | 0.1327 | 0.1242 | 0.9962 | 0.9969 | 0.0176 | 0.0154 |
| ANFIS-GA | 3 | 0.1327 | 0.1242 | 0.9962 | 0.9969 | 0.0176 | 0.0154 |
| *15-min interval* | | | | | | | |
| MLFFNN | 1 | 0.07137 | 0.11164 | 0.99911 | 0.9978 | 0.0050943 | 0.012463 |
| SVR | 4 | 0.1498 | 0.1470 | 0.9952 | 0.9897 | 0.0224 | 0.0216 |
| FIS | 2 | 1.1053 | 0.9001 | 0.8209 | 0.8162 | 1.2216 | 0.8102 |
| ANFIS-FCM | 1 | 0.2540 | 0.5122 | 0.9863 | 0.9198 | 0.0645 | 0.2623 |
| ANFIS-SC | 3 | 0.2712 | 0.4345 | 0.9844 | 0.92868 | 0.0735 | 0.1887 |
| ANFIS-GP | 1 | 0.1962 | 0.5952 | 0.9919 | 0.8669 | 0.038 | 0.3542 |
| GMDH | 4 | 0.2466 | 0.2301 | 0.9894 | 0.99063 | 0.0608 | 0.0529 |
| ANFIS-PSO | 4 | 0.2514 | 0.2548 | 0.9890 | 0.9886 | 0.0632 | 0.0649 |
| ANFIS-GA | 1 | 0.3089 | 0.3007 | 0.9831 | 0.9851 | 0.0954 | 0.0904 |
| *30-min interval* | | | | | | | |
| MLFFNN | 1 | 0.2500 | 0.4129 | 0.9887 | 0.9689 | 0.0625 | 0.1705 |
| SVR | 3 | 0.1969 | 0.1715 | 0.9933 | 0.9938 | 0.0388 | 0.0294 |
| FIS | 3 | 2.0538 | 2.0008 | 0.8613 | 0.8162 | 4.2182 | 4.0031 |
| ANFIS-FCM | 3 | 0.4733 | 0.9730 | 0.9598 | 0.7789 | 0.2240 | 0.9467 |
| ANFIS-SC | 3 | 0.6571 | 0.7068 | 0.9209 | 0.8907 | 0.4317 | 0.4995 |
| ANFIS-GP | 1 | 0.5313 | 0.8391 | 0.9490 | 0.8484 | 0.2822 | 0.7040 |
| GMDH | 3 | 0.5880 | 0.5074 | 0.9366 | 0.9522 | 0.3457 | 0.2575 |
| ANFIS-PSO | 3 | 0.5731 | 0.6001 | 0.9392 | 0.9341 | 0.3248 | 0.3601 |
| ANFIS-GA | 3 | 0.6589 | 0.6487 | 0.9174 | 0.9308 | 0.4342 | 0.4208 |

disagreement. This method has reported a high correlation between the outputs and targets. The best performance of the GMDH technique was obtained with 10 layers, 15 neurons in each layer and selection pressure of 0.6. Also, the network generated a low RMSE and MSE for the training and testing samples in the three datasets.

6. Two optimization algorithms (PSO and GA) were used to optimize the performance of the ANFIS model. The results have shown that the hybrid models performed better than the ANFIS model. Also, the ANFIS-PSO outperformed the ANFIS-GA.
7. In General, the maximum performance was obtained for the MLFFNN, SVR, GMDH, ANFIS-PSO and ANFIS-GA techniques. For the four datasets, ANFIS methods demonstrated the correlation coefficient approximately above 0.9 for test data. In this study, GMDH type neural network that for the first time was developed to predict the time-series wind speed data illustrated a high performance for the all datasets.

There is a direct influence of wind speed on the generated power of wind turbines. In this study, a proper knowledge of MLAs was employed to predict the wind speed in Osorio wind farm region. In different time intervals, the best intelligent model was introduced.

## References


[1] Khosravi A, Koury RNN, Machado L. Thermo-economic analysis and sizing of the components of an ejector expansion refrigeration system. Int J Refrig 2018;86:463–79.
[2] Song J, Wang J, Lu H. A novel combined model based on advanced optimization algorithm for short-term wind speed forecasting. Appl Energy 2018;215:643–58.
[3] Khosravi A, Koury RNN, Machado L, Pabon JJG. Energy, exergy and economic analysis of a hybrid renewable energy with hydrogen storage system. Energy 2018;148:1087–102.
[4] Zuluaga CD, Álvarez MA, Giraldo E. Short-term wind speed prediction based on robust Kalman filtering: an experimental comparison. Appl. Energy 2015;156:321–30.
[5] Fadare DA. The application of artificial neural networks to mapping of wind speed profile for energy application in Nigeria. Appl Energy 2010;87:934–42.
[6] Douak F, Melgani F, Benoudjit N. Kernel ridge regression with active learning for wind speed prediction. Appl Energy 2013;103:328–40.
[7] Chen K, Yu J. Short-term wind speed prediction using an unscented Kalman filter based state-space support vector regression approach. Appl Energy Jan. 2014;113:690–705.
[8] Jiang Y, Huang G. Short-term wind speed prediction: hybrid of ensemble empirical mode decomposition, feature selection and error correction. Energy Convers Manage 2017;144:340–50.
[9] Filik ÜB, Filik T. Wind speed prediction using artificial neural networks based on multiple local measurements in Eskisehir. Energy Proc 2017;107:264–9.
[10] Kirbas I, Kerem A. Short-term wind speed prediction based on artificial neural network models. Meas Control 2016;49:183–90.
[11] Akçay H, Filik T. Short-term wind speed forecasting by spectral analysis from long-term observations with missing values. Appl Energy 2017;191:653–62.
[12] Carta JA, Cabrera P, Matías JM, Castellano F. Comparison of feature selection methods using ANNs in MCP-wind speed methods. a case study. Appl Energy 2015;158:490–507.
[13] Celik AN, Kolhe M. Generalized feed-forward based method for wind energy prediction. Appl Energy 2013;101:582–8.
[14] Liu H, Tian H-Q, Liang X-F, Li Y-F. Wind speed forecasting approach using secondary decomposition algorithm and Elman neural networks. Appl Energy 2015;157:183–94.
[15] Zhao W, Wei YM, Su Z. One day ahead wind speed forecasting: a resampling-based approach. Appl Energy 2016;178:886–901.
[16] Wang J, Niu T, Lu H, Guo Z, Yang W, Du P. An analysis-forecast system for uncertainty modeling of wind speed: a case study of large-scale wind farms. Appl Energy 2018;211:492–512.
[17] Ivakhnenko A. Polynomial theory of complex systems. IEEE Trans Syst Man Cybern 1971;1(4):364–78.
[18] Ahmadi MH, Ahmadi MA, Mehrpooya M, Rosen MA. Using GMDH neural networks to model the power and torque of a stirling engine. Sustain 2015;7:2243–55.
[19] Vapnik V. The nature of statistical learning theory. Springer Science & Business Media; 2013.
[20] Khosravi A, Pabon JJG, Koury RNN, Machado L. Using machine learning algorithms to predict the pressure drop during evaporation of R407C. Appl Therm Eng 2018;133:361–70.
[21] Cheng HY, Yu CC, Lin SJ. Bi-model short-term solar irradiance prediction using support vector regressors. Energy 2014;70:121–7.
[22] Yaslan Y, Bican B. Empirical mode decomposition based denoising method with support vector regression for time series prediction: a case study for electricity load forecasting. Measurement 2017;103:52–61.
[23] Antonanzas J, Urraca R, Martinez-De-Pison FJ, Antonanzas-Torres F. Solar irradiation mapping with exogenous data from support vector regression machines estimations. Energy Convers Manage 2015;100:380–90.
[24] Mohammadi K, Shamshirband S, Anisi MH, Amjad Alam K, Petković D. Support vector regression based prediction of global solar radiation on a horizontal surface. Energy Convers Manage 2015;91:433–41.
[25] Khosravi A, Koury RNN, Machado L, Pabon JJG. Prediction of hourly solar radiation in Abu Musa Island using machine learning algorithms. J Clean Prod 2018;176:63–75.
[26] Khosravi A, Koury RNN, Machado L, Pabon JJG. Prediction of wind speed and wind direction using artificial neural network, support vector regression and adaptive neuro-fuzzy inference system. Sustain Energy Technol Assess 2018;25:146–60.
[27] Wang J, Zhai ZJ, Jing Y, Zhang C. Particle swarm optimization for redundant building cooling heating and power system. Appl Energy 2010;87(12):3668–79.
[28] Chang GW, Lu HJ, Chang YR, Lee YD. An improved neural network-based approach for short-term wind speed and power forecast. Renew Energy 2017;105:301–11.
[29] Ramasamy P, Chandel SS, Yadav AK. Wind speed prediction in the mountainous region of India using an artificial neural network model. Renew Energy 2015;80:338–47.
[30] Doucoure B, Agbossou K, Cardenas A. Time series prediction using artificial wavelet neural network and multi-resolution analysis: application to wind speed data. Renew Energy 2016;92:202–11.
[31] Liu H, Shi J, Erdem E. Prediction of wind speed time series using modified Taylor Kriging method. Energy 2010;35:4870–9.
[32] Noorollahi Y, Jokar MA, Kalhor A. Using artificial neural networks for temporal and spatial wind speed forecasting in Iran. Energy Convers Manage 2016;115:17–25.
[33] Schicker I, Papazek P, Kann A, Wang Y. Short-range wind speed predictions for complex terrain using an interval-artificial neural network. Energy Proc 2017;125:199–206.
[34] Kumar G, Malik H. Generalized regression neural network based wind speed prediction model for western region of India. Proc Comput Sci 2016;93:26–32.
[35] Sheela KG, Deepa SN, Newcomb RW. Neural network based hybrid computing model for wind speed prediction. Neurocomputing 2013;122:425–9.
[36] Petkovic D. Adaptive neuro-fuzzy approach for estimation of wind speed distribution. Int J Electr Power Energy Syst 2015;73:389–92.
[37] Santamaría-Bonfil G, Reyes-Ballesteros A, Gershenson C. Wind speed forecasting for wind farms: a method based on support vector regression. Renew Energy 2016;85:790–809.
[38] Mi X, Liu H, Li Y. Wind speed forecasting method using wavelet, extreme learning machine and outlier correction algorithm. Energy Convers Manage 2017;151:709–22.
[39] Li G, Shi J. On comparing three artificial neural networks for wind speed forecasting. Appl Energy 2010;87:2313–20.
[40] Koo J, Han GD, Choi HJ, Shim JH. Wind-speed prediction and analysis based on geological and distance variables using an artificial neural network: a case study in South Korea. Energy 2015;93:1296–302.
[41] Lodge A, Yu X-H. Short term wind speed prediction using artificial neural networks. In: 2014 4th IEEE international conference on information science and technology; 2014. p. 539–42.
[42] Mohandes MA, Halawani TO, Rehman S, Hussain AA. Support vector machines for wind speed prediction. Renew Energy 2004;29:939–47.
[43] Bilgili M, Sahin B, Yasar A. Application of artificial neural networks for the wind speed prediction of target station using reference stations data. Renew Energy 2007;32:2350–60.
[44] Salcedo-Sanz S, Ortiz-Garcıa EG, Pérez-Bellido ÁM, Portilla-Figueras A, Prieto L. Short term wind speed prediction based on evolutionary support vector regression algorithms. Expert Syst Appl 2011;38:4052–7.
[45] Liu D, Niu D, Wang H, Fan L. Short-term wind speed forecasting using wavelet transform and support vector machines optimized by genetic algorithm. Renew Energy 2014;62:592–7.
[46] Peng Y, Rysanek A, Nagy Z, Schlüter A. Using machine learning techniques for occupancy-prediction-based cooling control in office buildings. Appl Energy 2018;211:1343–58.
[47] Ceylan I, Gedik E, Erkaymaz O, Gürel AE. The artificial neural network model to estimate the photovoltaic module efficiency for all regions of the Turkey. Energy Build 2014;84:258–67.
[48] Katambara Z, Ndiritu J. A fuzzy inference system for modelling streamflow: case of Letaba River, South Africa. Phys Chem Earth 2009;34:688–700.
[49] Haznedar B, Kalinli A. Training ANFIS using genetic algorithm for dynamic systems identification. Int J Intell Syst Appl Eng 2016;4:44–7.