Technology and Economics of Smart Grids and Sustainable Energy (2021) 6: 23
https://doi.org/10.1007/s40866-021-00119-6

ORIGINAL PAPER

A Hybrid Neuro-Fuzzy & Bootstrap Prediction System for Wind Power Generation

Ahmed A. Abdullah[1] · Tarek M. Hassan[1]

Received: 8 May 2021 / Accepted: 29 September 2021 / Published online: 17 November 2021

Abstract
Increasing integration of wind turbines in the electrical grid creates more challenges daily because of the unstable power of these units because they rely mainly on the wind. One of the solutions used to deal with these challenges is to predict the power produced from the wind turbine for a short or long term. Prediction of the power to be produced from the wind turbines gives a future vision of how to deal with these sources. Many techniques are used to improve the prediction of wind turbine energy. In this paper, a bootstrap hybrid neuro-fuzzy short-term prediction system for wind turbine power is presented. The bootstrap technique is used to increase the accuracy of the predicted value by creating multiple small datasets with the same features of the main dataset. A neuro-fuzzy model is created for each dataset using the hybrid optimization agent in MATLAB. The output of each neuro-fuzzy model is a prediction of wind turbine power. Therefore, the output of all models is combined using the combination model to calculate the final predicted wind turbine power. The system is simulated using MATLAB and the result show that the bootstrap hybrid neuro-fuzzy system predicts the wind turbine power for the next 24 h with accuracy 94.3%. The relative percentage error and average error for the system outputs are calculated and showed that the error is minimized to ±5.7%. Therefore, this prediction system could be useful for the challenges of integration wind turbine in electrical grid

Keywords Forecasting · Wind turbine · Bootstrap · Neuro-fuzzy · ANFIS

Introduction

The amount of electricity generated by wind energy fed into the electric grid has increase rapidly in recent years and wind energy penetration levels have reached significant values in many countries [1]. Increasing the level of wind power penetration in the future, with the uncertainty of wind power, the electric grid will affect the stability of the power system and increase the risk of power outages.

Therefore, it is necessary to have a type of system that predicts the energy production of the wind turbines over the next 1–2 days to control the fully dispatchable generation, to increase its value in markets operating on a 48-h time scale, and to make full use of the wind energy produced.

The power generated by a wind turbine (WT) generator varies randomly over time due to changing wind speed, temperature, and other factors. Therefore, the necessity and advantages of short-term wind power forecasting are generally accepted and highly evaluated by most wind power facilities [2, 3]. Therefore, short-term (0–36 h) wind energy forecast is a central issue for proper grid-connected wind farm management and to make wind energy a more competitive and reliable resource [4]. Due to the difference in temperature, pressure, air density, terrain, and other factors, wind speed is one of the most difficult meteorological parameters for forecasting [5]. As a result, it will be difficult to predict the nonlinear power generated by the WT with acceptable accuracy.

Several methods have been proposed to predict wind power, namely, physical [6] and statistical methods [7]. The physical method has advantages in long-term prediction, while the statistical method does well in short-term prediction [8].

In recent years, with the continuous increase of computer calculation speed, researchers proposed a number of power

✉ Ahmed A. Abdullah
Ahmed.abdelaleem@deltauniv.edu.eg

Tarek M. Hassan
tarek.hassan@deltauniv.edu.eg

[1] Deptartment of Communication and Computer Engineering, Faculty of Engineering, Delta University for Science & Technology, International Coastal Road, Gamasa City, Mansoura, Dakhliya, Egypt

prediction models based on complex statistics and artificial intelligence techniques as found in [9–16]. Statistical models have the advantage that they use available historical measurements for stochastic approximation between wind prediction and wind power output measurements. However, those models are not suitable for long-term prediction and it is very difficult for the prediction models based on statistics to further improve the prediction accuracy [5].

Generally, the prediction models of wind power generation are categorized into direct and indirect models. Direct prediction models use the historical information of wind power output as the prediction model's input, and the output of the prediction model is the predicted value of wind power generation. Where, indirect prediction models predict wind power generation by predicting wind speed and then use the power curve to convert wind speed into power output [5]. This may cause a delay and some errors in calculations. The power curve depends on the wind speed, but not on the efficiency and other mechanical parameters of the wind turbine which could be changed with time. Therefore, the power curve should be correct every specified time interval.

A double-stage hierarchical genetic algorithm trained artificial neural network (double-stage hybrid GA-ANN) for short-term wind power forecast of a microgrid wind farm in Beijing, China, is presented in [9].

In [10], a novel two-stage hybrid approach based on the combination of the Hilbert-Huang transform (HHT), genetic algorithm (GA), and artificial neural network (ANN) is proposed for day-ahead wind power forecasting.

In [11], a short-term wind farm output power prediction model is presented using fuzzy modeling derived from the raw data of the wind farms. The proposed model uses a new framework based on a combination of prediction intervals (PI) to overcome the performance instability of neural networks (NNs) used in the proposed method. In addition, a new fuzzy-based cost function is proposed with the purpose of having more freedom and flexibility in adjusting NN parameters used for the construction of PIs.

A combination between the adaptive neuro-fuzzy inference system (ANFIS) and an artificial neural network (ANN) for 1 h ahead wind speed forecast is proposed in [12].

In [13], an energy forecasting model is presented for analysis and prediction of power generation from wind farms. The model has been developed with the help of neural network methodology. It involves three input variables—wind speed, relative humidity, and generation hours and one output variable energy output of wind farms.

In [14], an output prediction system of wind power generation is proposed using a complex-valued neural network (called CVNN). The CVNN is suitable for treating complex numbers and nonlinear data. In this model, the wind information (wind speed and direction) is expressed by complex numbers with complex coordinates, and use them as input information for the CVNN.

In [15], a new method of BP neural network to predict the wind power output in a period of time is presented. In this method, to discuss the predictive performance of BP neural networks, a different number of input variables to observe the prediction effect of BP neural network is set.

The neural network has a major role in establishing and development new models for the prediction of wind power. Therefore, an optimal wind power forecasting model has been studied using various conventional, neuro-fuzzy and artificial intelligence techniques in previous literature. Traditional methods take more time than that by artificial neural network (ANN) or neuro-fuzzy to calculate the predicted value of Wind Turbine Power (WTP) [9–16]. The error in the learning of neuro-fuzzy can be reduced by bootstrap aggregating, also called bagging algorithm that will increase the accuracy of the system [16].

Weather statistical or mathematical method that was displayed before needs a large amount of data to build the mathematical or statistical model. Dealing with the large amount of data takes a lot of time and effort and makes the created model be complex with percentage error. Moreover, the created complex model takes more time to predict the WTP value.

The neuro-fuzzy technique proposes a lot of features that can be used to predict the WTP value using a large amount of data to train the model and build the fuzzy inference system. If the amount of data is very large, training the model to be accurate will take a lot of time and the prediction process is more complex [12]. Therefore, the bootstrap technique could help to deal with the large amount of data without losing data features.

In [17], an Adaptive Neural Fuzzy Inference System (ANFIS) approach for short-term wind power prediction in Portugal is proposed using a hybrid particle swarm optimization (PSO). The proposed approach is based on the combination of PSO and ANFIS. In this approach, the prediction error sometimes arises to more than 21% in different seasons of the year, which is considered as a high error percentage.

In this paper, the bootstrap technique is used with ANFIS to increase the accuracy of prediction for WTP. Therefore, the objective of this paper is to predict the optimal WTP by using the bootstrap algorithm. Bootstrap technique is used to deal with a large amount of data by selecting multiple random samples of data without losing data features. Each sample is used to build a fuzzy inference model (FIS) to predict the wind power value. After build the multiple FIS, the final combination of the multiple FIS is produced to be more accurate. Wind speed, temperature, air pressure, humidity, and wind angle will play key roles in the training of neuro-fuzzy and to predict the optimal value of WTP. The bootstrap aggregation technique, which called bagging technique [18], is an ensemble method that can be used in regression and classification. Bagging is composed of two parts: aggregation and bootstrapping. It is used to reduce the error in the predicted value by distributing the error to datasets that have the same characteristics as the main dataset.

This paper is organized as following: a background for WT modeling, bootstrapping aggregating, neuro-fuzzy technology is presented in "Background" section. In "Proposed System" section, the proposed prediction system is illustrated, including collecting the data used in the prediction model (historical and real-time), bootstrapping algorithm, prediction neuro-fuzzy model, and combing model. Finally, the data from Alaska Center for Energy and Power (ACEP) is used to train and evaluate the prediction model as a case study and the result is shown. Case study demonstrates that the prediction system has a good agreement with the data selected and can describe input-output relations using a set of rules in the IF-THEN form. Short-term wind farm power output prediction model is presented using fuzzy modeling derived from the raw data of the wind farm. The comparative study of model prediction with different periods is conducted.

Background

Wind Turbine Modeling

This section describes the details of wind turbine modeling. First, the wind energy conversion system is briefly discussed. Next, the modeling and key factors affecting wind farm output power are presented for use in the proposed wind power prediction model. The wind farm's production capacity comes from the wind energy captured by WTs. The power generated by an ideal and practical wind turbine is discussed in the next section [19].

Essentially, wind energy is the kinetic energy, and it can be calculated using (1).

$$E = \frac{1}{2} m v^2 (\text{jouls}) = \frac{1}{2} (\rho A x) v^2 \qquad (1)$$

where m is the air mass (kg) and v is the air velocity in the upstream wind direction at the inlet of the rotor blades (m/s.), A is the cross-sectional area in m2, is the air density in kg/m3, and x is the piece thickness in metres.

Wind energy Pw (in watts) is the time derivative of kinetic energy and can be derived from the wind kinetic energy using:

$$P_w = \frac{dE}{dt} = \frac{1}{2} \left(\rho A \frac{dx}{dt} \right) . v^2 = \frac{1}{2} (\rho . A v) . v^2 = \frac{1}{2} . \rho A v^3 \qquad (2)$$

Actually, the output mechanical power Pm extracted from WT blades can be obtained using (2) multiplied by coefficients as follows [20].

$$P_m = k\, C_p \frac{1}{2} . \rho . A v^3 \qquad (3)$$

Where:

Pm Mechanical power output, kilowatts.

Cp Maximum power coefficient, ranging from 0.25 to 0.45, dimension less (theoretical maximum = 0.59).

A Rotor swept area, ft2 or π D2/4 (D is the rotor diameter in ft., π = 3.1416).

v Wind speed, mph.

k 0.000133, A constant to yield power in kilowatts. (Multiplying the above kilowatt by 1.340 converts it to horse- power [i.e., 1 kW 1.340 hp]).

ρ Air density, lb./ft3, The air density ρ is calculated by [21]:

$$\rho = \frac{P}{R_{\text{Specific}} . T} \qquad (4)$$

Where P is the absolute pressure, T is the absolute temperature (K), and R is the specific gas constant for dry air.

According to eq. (3), the extracted mechanical power of a WT generator is proportional to the air density ρ, and the air speed in the upstream wind direction at the entrance of the rotor blade v, where ρ is determined mainly by the air temperature as shown in eq. (4) [21].

Therefore, the major factors affecting wind farm output power are the relationship of the factors temperature, wind speed, density, and pressure which have been shown above in (3, 4). These factors are assembled in the proposed model to predict the expected wind power.

Bootstrap Aggregating

For a given data set, the neuro-fuzzy works well without changing in the dataset. However, when the data set changes in the next hour or the next day, two problems could be occurred, overfitting or underfitting. Even after training several times of neuro-fuzzy, the result between the actual and desired values does not match and the error is increase. The reason for this error is in the learning process due to overfitting and underfitting problems. Therefore, the error in learning depends on three main factors; noise, bias, and variance [22] as shown in (5).

$$Error = Noise + Bais + Variance \qquad (5)$$

The large bias causes with data underfitting while the large variance causes with overfitting of data. Therefore, if the dataset is bootstrapped to multi-dataset and neuro-fuzzy adapted to multi-dataset, the bias and variance will be reduced to a minimum as the data will not suffer the problem of overfitting and underfitting anymore. Therefore, the error in learning also reduces and the overall prediction improves.

The bootstrap bagging technique is used to improve the stability and precision of neuro-fuzzy models. It minimizes the variance which further helps in reducing the overfitting [23]. The bootstrap technique creates multiple small data sets from the main data set. The small data sets have statistical features like the main dataset. Therefore, a classifier is used to calculate the statistical features of each small data set. A combine classifier takes the statistical features from all small datasets and combines these features to one value.

The bootstrap technique uses two classifications, small and disjoint partition. In this study, small bootstrap bagging is implemented by using MATLAB to subset the data. In the small bag, the union of subsets may not necessarily be equal to the original data set because of the repetition of numbers.

Neuro-Fuzzy Technique

A fuzzy logical system is a nonlinear mapping of an input vector to a scalar output, but it can handle numerical values and linguistic knowledge. In general, a fuzzy logic system has four components: Fuzzifier, Grammar, Inference Engine, and Puzzle Remover. Fuzzifier transforms a clear input variable into a fuzzy representation, where the membership functions give the degree of belonging of the variable to a particular attribute. Ambiguous grammar is of the “if-then” type and can be derived from numerical data or from specialized linguistics. Mamdani and Sugeno inference engines are the main types of inference mechanisms [24].

The Mamdani engine combines fuzzy rules in mapping from fuzzy input group to fuzzy output group, while the Takagi-Sugeno type links blurred inputs to clear outputs. Eliminator converts a fuzzy set to a crisp number using the centroid of the region, the median of the region, the mean of the maximum, or the extreme criteria [25].

The advantage of a neural network with fuzzy logic models is that knowledge is acquired automatically during the learning process. However, this knowledge cannot be extracted from a trained network that acts as a black box. On the other hand, fuzzy systems can be understood through their rules, but these rules are difficult to define when the system has a lot of variables and their relationships are complex [26].

The combination of a neural network and a neuro-fuzzy system has its own advantages. In a neuro-fuzzy system, neural networks automatically extract fuzzy rules from digital data, and through the learning process, the membership functions are adaptively modified.

Proposed System

The proposed system has been built to forecast the WTP value with high accuracy and minimum time. Traditional neuro-fuzzy with the large amount of data takes more time to calculate the predicted value of WTP. Moreover, the error in the learning of neuro-fuzzy can be reduced by a bootstrap aggregating algorithm that will increase the accuracy and speed of the system. Therefore, the data is entered to the bootstrap algorithm to produce multiple sets of data, then a neuro-fuzzy model will be created for each data set. The output of each neuro-fuzzy model (predicted value) will be combined with each other to produce the final value of prediction.

Figure 1 shows the framework of the proposed prediction system. The proposed system is divided into the main data set which enters to the bootstrap algorithm. The bootstrap algorithm creates multiple small data sets and then path these data sets to the neuro-fuzzy creator. The neuro-fuzzy creator and learning (NF-agent) create the neuro-fuzzy prediction module for each data set and apply learning for each. After the prediction, the neuro-fuzzy module is learned and predicts the value of WTP, The combining model combines the neuro-fuzzy model output with each other to calculate the final output predicted value.

The proposed system is divided into the following steps:

1- Determine and collect the data parameters to be collected from the historical database.
2- Design the bootstrap algorithm to create multiple data sets.
3- Design a neuro-fuzzy model for each data set and learn each model.
4- Combine the output of each neuro-fuzzy model to compute the final predicted output of WTP.
5- Compare the system with the traditional neuro-fuzzy model.

Determine and Collect Data Parameters

Wind turbine data is usually collected by a Supervisory Control and Data Acquisition (SCADA) system in the monitoring center [27]. Therefore, historical data can be obtained from a SCADA system installed at a wind farm. The data used in this paper are based on the ones collected at one 1 h interval from Alaska Center for Energy and Power for a period of 1 year [28].

As mentioned earlier, the major factors affecting wind farm output power are air temperature, wind speed, air density, and air pressure. Considering all these factors will increase the prediction accuracy of the proposed

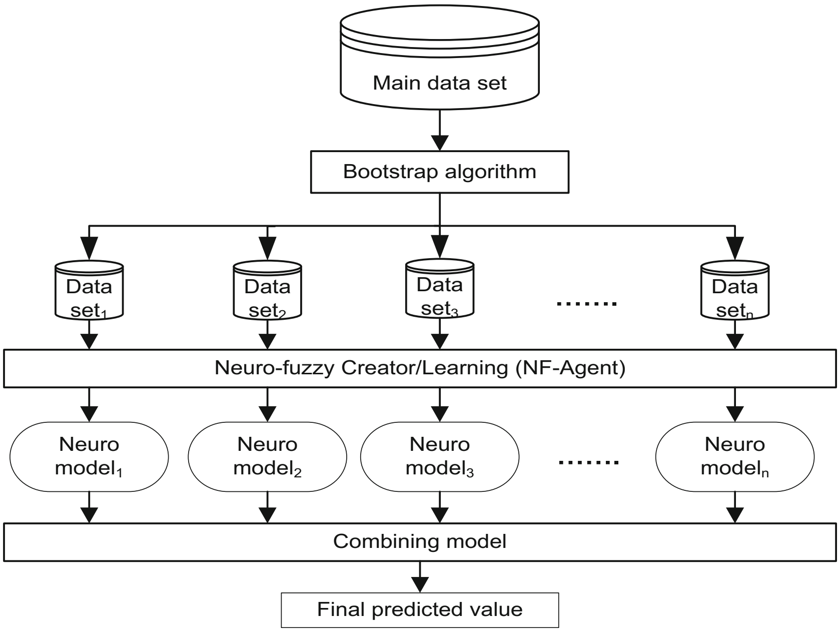


Fig. 1 Framework of the prediction system

model. The historical power output collected by the Energy Management System (EMS) can be used as input to improve the accuracy of prediction. The data used in this paper includes the parameters of output power, temperature, density, pressure, and wind speed for 1 year by using as input to bootstrap algorithm to produce the multidata set. The proposed model uses the latest 5 real values for all measured parameters as inputs to the prediction system to predict the power in the next day, i.e., the model uses historical data for 4 days and current real-time data for predicting the power in the next day. Therefore, the input parameters of the model are listed in Table 1.

The parameters air temperature T (t), wind speed V (t), air density ρ (t), pressure P (t), and output power S (t) are the real-time measured values coming from SCADA historical database.

Bootstrap Algorithm

Bootstrap is a recently developed technique for making certain kinds of statistical inferences. The bootstrap ensemble method creates individual data set samples with N sizes of the original dataset. Many of the original datasets may be repeated in the resulting datasets while others may be left out. Therefore, the bootstrap algorithm is responsible to create multiple data sets depending on the original data parameters collected. Each data set should have the same features of the main database. Algorithm 1 shows the bootstrapping method to generate R number of the dataset.

The bootstrap algorithm starts to calculate the mean and standard deviation of the main dataset (M_avg, M_Dev), and then generate R number of the dataset. The algorithm generates the dataset by generating N random numbers and selects the dataset members according to the random numbers. For each dataset, it computes the mean and standard deviation (Sj_avg, Sj_Dev), and then compares them with the main mean and main standard deviation. Flowchart 1 shows the bootstrap method to generate multiple datasets for the prediction of WTP.

Algorithm (1): bootstrap method to create multiple datasets.

Table 1 Input parameters of the model

input No.	Parameter name	Abbreviation	Symbol	Unit
1–5	Air temperature	T	T(t-4), T(t-3), T(t-2), T(t-1), T(t)	C
5–10	Wind Speed	V	V(t-4), V(t-3), V(t-2), V(t-1), V(t)	mph
10–15	Air density	*P*	ρ(t-4), ρ(t-3), ρ(t-2),ρ(t-1), ρ(t)	lb./ft^3
15–20	Pressure	P	P(t-4), P(t-3), P(t-2), P(t-1), P(t)	Pa
20–25	Output power	S	S(t-4), S(t-3), S(t-2), S(t-1),S(t)	kw

Bootstrap algorithm

- ***Input:***
  - ○ Main data set Z= {z_1, z_2,….z_L}.with z_i=(T_i(t-4), T_i(t-3)..T_i(t), V_i(t-4), V_i(t-3)..V_i(t), P_i(t-4), P_i(t-3)..P_i(t), ρ_i(t-4), ρ_i(t-3)..ρ_i(t), S_i(t-4), S_i(t-3)..S_i(t))
  - ○ Number of dataset want to be generated (R)
  - ○ Number of rows in each created dataset (N)
- ***Output:***
  - ○ S_j(Z) data sets, where j=1 to R
- ***Steps:***

```
                                   // calculate the average of main dataset

$M\_Dev = \frac{1}{L}\sum_{i=1}^{L}(z_i - M\_avg)^2$            // calculate the standard deviation of main dataset
For each input j=1 to R do
Resample:
  For k=1 to N
    Rand=create random number (1 to L)
    $S_j(k)=Z_{(Rand)}$
  End loop
                                   // calculate the average of created dataset

$S_j\_Dev = \frac{1}{N}\sum_{j=1}^{N}(S_j - S_j\_avg)^2$          // calculate the standard deviation of main dataset
If |M_avg - S_j_avg|> (M_avg/10) then goto resample     //recreate the dataset if the average is not the same as main average with 10%
                         error
If |M_Dev - S_j_Dev|> (M_Dev/10) then goto resample   //recreate the dataset if the deviation is not the same as main deviation with 10%
                         Error.
End loop
```

Prediction Neuro-Fuzzy Model

The datasets created by the bootstrap algorithm are representing the input data space and it is used to create, train, and validate the neuro-fuzzy models. For each dataset, a neuro-fuzzy model is created. The datasets are clustered using a modified Fuzzy C-Means (FCM), which is a data clustering algorithm to decide the optimal number of fuzzy rules by grouping the dataset into M clusters with every data point in the dataset belonging to every cluster to a certain degree and assign a rule for each cluster [29]. Based on the data in each cluster, the number of rules and membership functions are adjusted. Hybrid optimization method, which combines least-squares and back propagation gradient descent method, is used to train the membership function parameters to emulate the training data [30].

Based on ANFIS tool in MATLAB simulation program, the datasets are clustered and the neuro-fuzzy model is created and trained [11]. The ANFIS tool clusters each input in each dataset into k subspaces based on factor range of influence, squash factor, accept ratio, and reject ratio.

After clustering each training dataset parameter to C clusters, each input is illustrated to C membership function with Gaussian membership function. ANFIS tools design the neuro-fuzzy model as a set of rules depending on the number of clusters number. The rules are set in the IF-THEN form to describe input-output relations using Takagi-Sugeno with a linear function of output. A multi-input and single-output fuzzy model is represented as a collection of fuzzy rules in a Sugeno fuzzy model which has the following form.

$$\mathrm{Rk} : \mathrm{IF}\ \mathrm{X1}\ \mathrm{is}\ \mathrm{A1\,j}\ \mathrm{and}\ldots..\mathrm{Xn}\ \mathrm{is}\ \mathrm{Anj}\ \mathrm{then\ output} = \mathrm{fk}\ (\mathrm{Xi})$$

¬ where X = (X1,X2,…Xn) are the inputs to the neuro-fuzzy model, A1j,A2j,….Anj are linguistic variables, Rk represents the kth rule, k, j = 1,2,…C which represent the cluster number, f(Xi) is Linear relation between inputs and output which is illustrated in (6).

$$f_k(x) = b_{0j} + \sum_{i=1}^{n} b_{ij}x_i \tag{6}$$

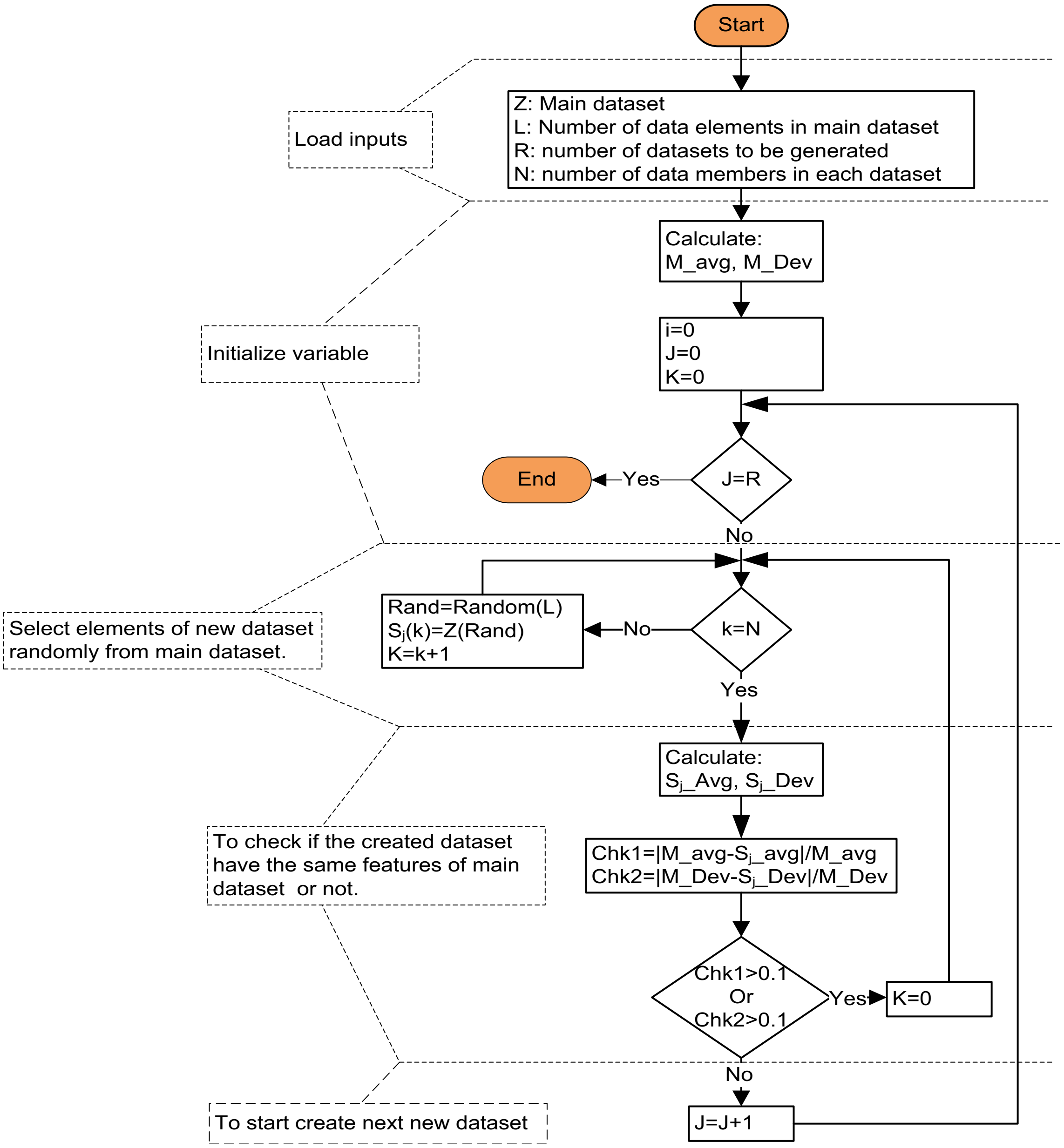
The crisp output is calculated using eq. (7). The output level fp of each ru le is weighted by the firing strength wp of the rule.

$$\mathrm{output} = \frac{\sum_{p=1}^{k} w_p f_p}{\sum_{p=1}^{k} w_p} \tag{7}$$

Wp is the weight of a and membership value for a linguistic variable for the inputs in the STATEMENT of the rule P. The weight wp relation is illustrated in eq. (8).

$$w_p = \mathrm{andmethod}\left(\mu_{ij}\right) \tag{8}$$

The and-method may be in the form of product or minimum for membership value. In this paper, the product and-method is used. The block diagram of the prediction model is shown in Fig. 2. The input of the neuro-fuzzy model is coming from the bootstrap algorithm which generates R number of datasets. These data are fuzzified using the Gaussian membership function which generated using the parameters from the


Start
Load inputs
Z: Main dataset
L: Number of data elements in main dataset
R: number of datasets to be generated
N: number of data members in each dataset
Calculate:
M_avg, M_Dev
Initialize variable
i=0
J=0
K=0
End
Yes
J=R
No
Select elements of new dataset randomly from main dataset.
Rand=Random(L)
S_j(k)=Z(Rand)
K=k+1
No
k=N
Yes
Calculate:
S_j_Avg, S_j_Dev
To check if the created dataset have the same features of main dataset or not.
Chk1=|M_avg-S_j_avg|/M_avg
Chk2=|M_Dev-S_j_Dev|/M_Dev
Chk1>0.1
Or
Chk2>0.1
Yes
K=0
No
To start create next new dataset
J=J+1

Flowchart 1 Is a bootstrap method to generate multiple datasets for the prediction of wind turbine power

clustering block. The clustering block generates the center and width of the membership function depending on the formatting of the data. The crisp input enters to the rule base to inference inputs and calculate the output. Number of rules (C) is calculated by the clustering block. The fuzzy logic system with wtaver de-fuzzifier, product inference rule, and Gaussian-minimum fuzzifier is designed to predict the output power of the wind turbine for the next future 1 days. The wtaver is a defuzzifier method in ANFIS tool in matlab.

Algorithm 2 shows the neuro-fuzzy prediction algorithm for WTP generation. The input for this algorithm is real and historical data of the wind turbine parameters as shown in Table 1. The output is the predicted value of the WTP generation in the next future one day. The steps of the algorithm begin with fetching the real and historical values of parameters. These data are clustered using the c-means technique and calculate the membership function for each value. The rule base for these values is built in the fuzzy inference system (FIS) according to the rule stock generated by the clustering algorithm. The prediction value of the WTP generated is calculated and evaluated by the human-machine interface (HMI) users.

An Adaptive Network based Fuzzy Inference System can incorporate fuzzy if - then rules and which provide fine-tuning of the membership function according to a desired input-output data pair. Figure 3 shows the adaptive network with 5 layers for the neuro-fuzzy model of the prediction model. As mentioned before, a first-order linear sugeno fuzzy model is used as a means of modeling fuzzy rules into desired outputs.

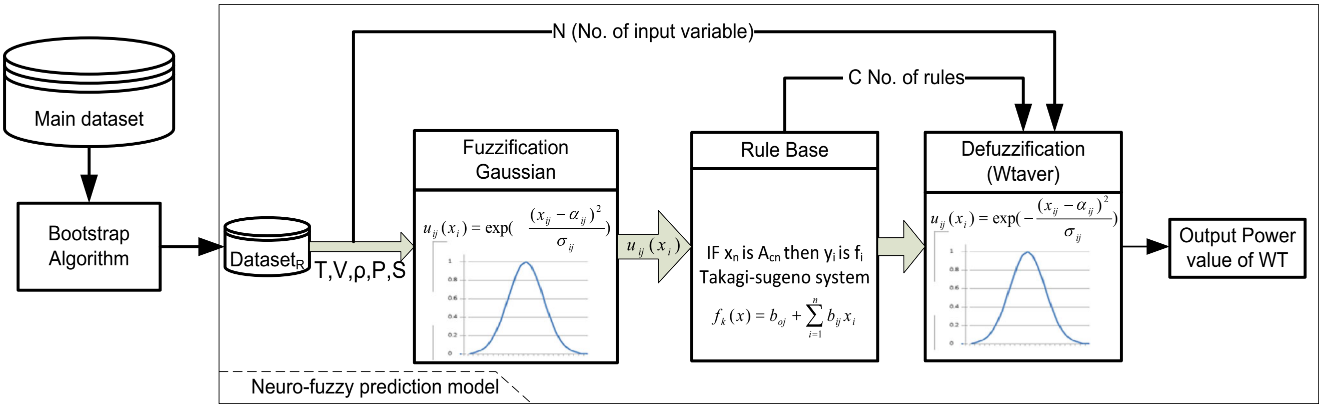

N (No. of input variable)
C No. of rules
Main dataset
Bootstrap Algorithm
Dataset$_R$
T,V,ρ,P,S
Fuzzification Gaussian
$u_{ij}(x_i) = \exp(\frac{(x_{ij}-\alpha_{ij})^2}{\sigma_{ij}})$
$u_{ij}(x_i)$
Rule Base
IF x_n is A_{cn} then y_i is f_i
Takagi-sugeno system
$f_k(x) = b_{oj} + \sum_{i=1}^{n} b_{ij} x_i$
Defuzzification (Wtaver)
$u_{ij}(x_i) = \exp(-\frac{(x_{ij}-\alpha_{ij})^2}{\sigma_{ij}})$
Output Power value of WT
Neuro-fuzzy prediction model

Fig. 2 Block diagram of neuro-fuzzy prediction model for dataset R

Algorithm 2: Wind Power Generation Prediction Algorithm.

WTP Generation neuro-fuzzy Prediction Algorithm

- ***Input:***
  - Real time value of parameter (T,V,ρ,P,S).
  - Historical value of (T,V,ρ,P,S) for 4 periods before.
    - T is Air temperature
    - V is wind speed.
    - ρ is air density.
    - P is pressure.
    - S is output power.
- ***Output:***
  - Power generation in next future 1day.
- ***Steps:***

```
1:  Read received real time data from SCADA for (T,V,ρ,P,S) parameters.
2:  Fetchhistorical data from SCADA database of (T,V,ρ,P,S) for 4 periods before.
        $T,V,ρ,P,S=T_i,V_i,ρ_i,P_i,S_i$ where i=0, 1,2,3,4
3:  For each input $X_i$
        4:  Clustering the data using C-Means.
        5:  Calculate the membership function using centroids and width
6:  Next
7:  Build fuzzy inference system (FIS) which contains the rule base.        // rule base is recognized using hybrid method.
8:  Train the FIS.
9:  Calculate the predicted value for next future 1 day using neuro-fuzzy model.
10: Validate the output with actual value and validate the prediction model.
```

The adaptive network consists of 5 layers. Layer 1 corresponds to a linguistic label and the membership function of this linguistic label. Layer 2 estimates the firing strength of a rule, which is found from the and-method (multiplication or minimum) of the incoming signals. Layer 3 estimates the ratio (wi) of the i-th rule's firing strength to sum of the firing strength of all rules, C. Layer 4 is the product of the previously found relative firing strength of the ith rule and the rule. The final layer computes the overall output as the summation of all incoming signals from layer 4 [11].

Combining Model

The combining model combines the outputs of neuro-fuzzy models to calculate the final output of WTP. The combining model calculates the mean of the output of neuro-fuzzy models as shown in (9).

$$WTPP = \frac{1}{R}\sum_{i=1}^{R} P_i \tag{9}$$

Where WTPP is the wind turbine predicted power for the next day, R is the number of datasets created by the bootstrap algorithm, Pi is the output predicted power of the neuro-fuzzy model.

Experimental Results

The data used in this paper is based on the ones collected at 1 h intervals from Alaska Center for Energy and Power for a period of 1 year [28]. For the selected wind turbine data, the bootstrapping algorithm is set to generate 10 datasets (R = 10 in the bootstrap algorithm), where each dataset contains 200 row data (N = 200) selected randomly with the same

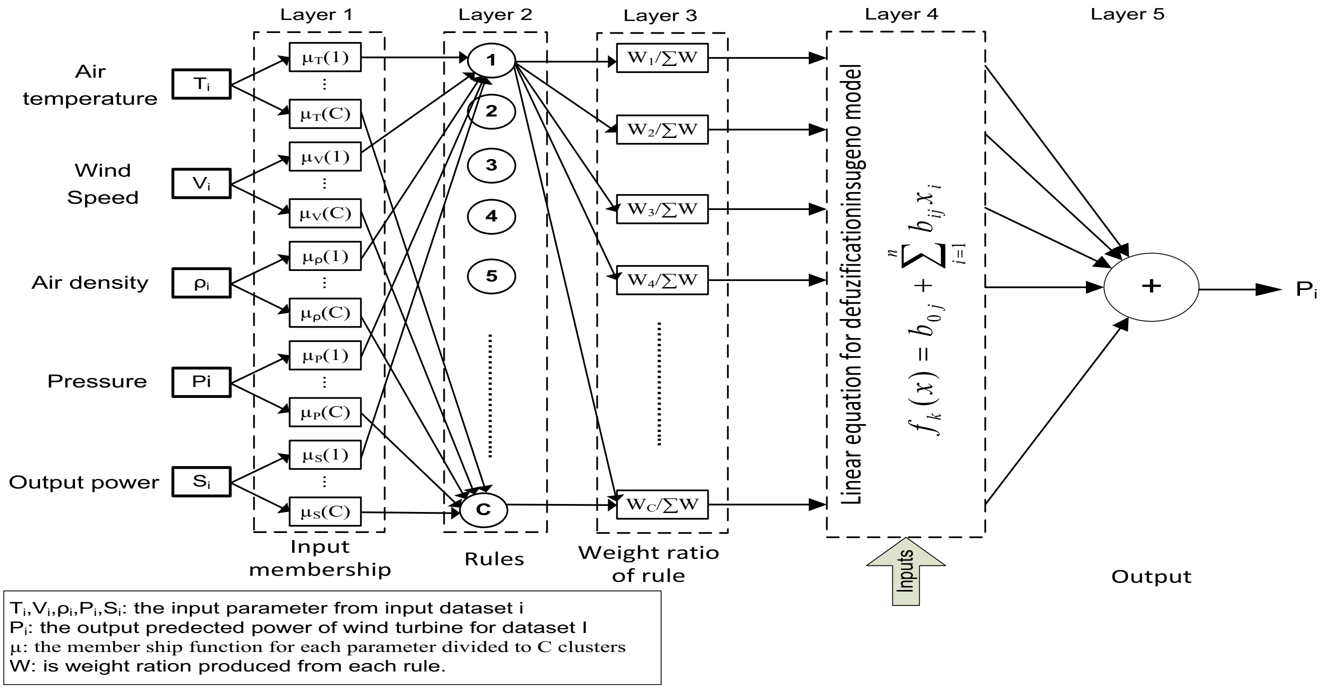

Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
Air temperature
Wind Speed
Air density
Pressure
Output power
Tᵢ
Vᵢ
ρᵢ
Pi
Sᵢ
μT(1)
μT(C)
μV(1)
μV(C)
μρ(1)
μρ(C)
μP(1)
μP(C)
μS(1)
μS(C)
1
2
3
4
5
C
W1/∑W
W2/∑W
W3/∑W
W4/∑W
WC/∑W
Linear equation for defuzificationinsugeno model
$f_k(x) = b_{0j} + \sum_{i=1}^{n} b_{ij} x_i$
Inputs
+
Pᵢ
Input membership
Rules
Weight ratio of rule
Output
Tᵢ,Vᵢ,ρᵢ,Pᵢ,Sᵢ: the input parameter from input dataset i
Pᵢ: the output predected power of wind turbine for dataset I
μ: the member ship function for each parameter divided to C clusters
W: is weight ration produced from each rule.

Fig. 3 Adaptive network with 5 layers for the neuro-fuzzy model for the prediction of WTP model

features of the main dataset. Table 2 shows the datasets created by the bootstrap algorithm from the main dataset collected.

The flowchart for the steps of creating 10 datasets, building neuro-fuzzy models, train, and evaluating the prediction models using MATLAB is shown in flowchart 2. In Step-1, the main dataset is loaded, and then the key values to be divided are initiated. The number of divided datasets is R = 10 and every dataset contains N = 200 data rows. In step-2, each dataset was clustered using the clustering technique in ANFIS. The ANFIS cluster parameters are set, then a cluster is created for dataset and passed to the next step. In step-3, the fuzzy rules are created and then the model training is started. The RMSE is calculated and compared with the acceptable RMSE. If RMSE is accepted, the system will check if it isn't the last dataset, it will repeat step-2 and step-3 for the next dataset. In step-4, the outputs of the neuro-fuzzy models were combined. Step-5 is to evaluate the model and compute error.

Table 2 Dataset created with main features

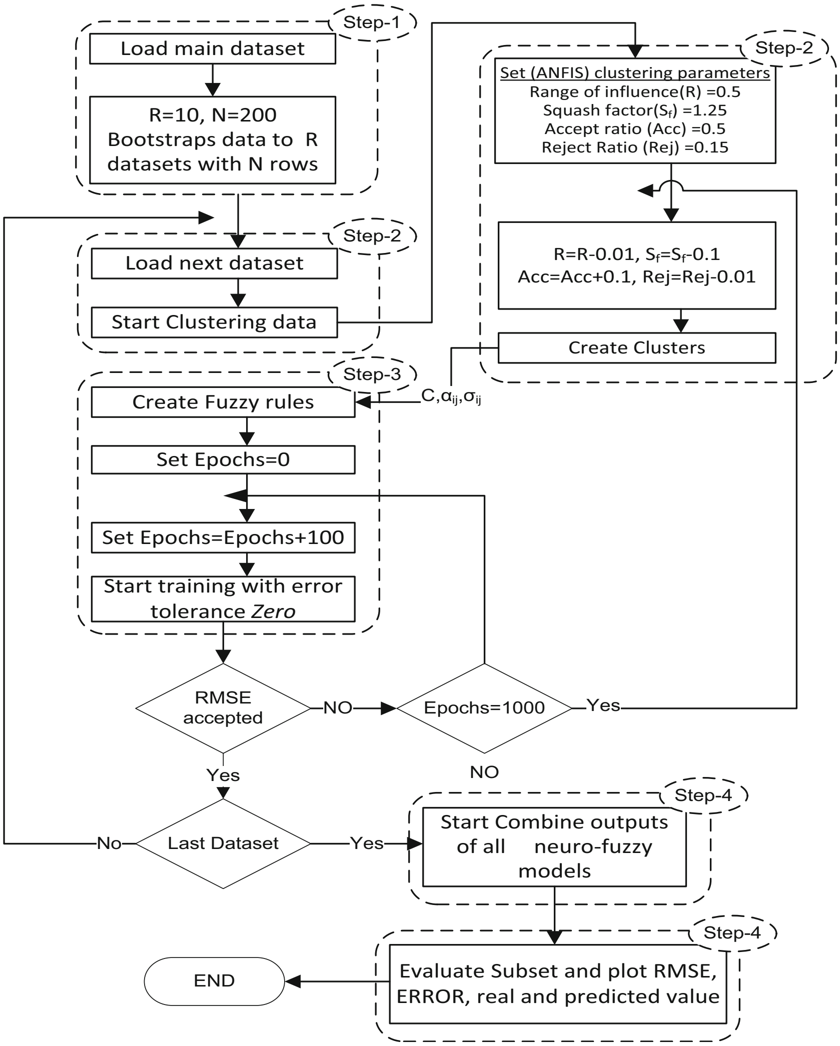
Main dataset=2000 row (mean=43.4, standard deviation=8.0025)
Period from 15 to 3-2014 to 1-2-2015

dataset	Checked values		No. of data row
	Mean	Standard deviation	
1	40.75	8.8	200
2	41.21	8.7	200
3	45.65	8.3	200
4	39.84	7.5	200
5	42.41	8.3	200
5	44.92	8.8	200
6	46.72	7.7	200
7	39.25	7.4	200
8	47.53	8.4	200
9	43.94	7.9	200
10	41.01	7.9	200

To measure the accuracy of the bootstrap prediction model, Root Mean Square Error (RMSE), Average Root Mean Square Error (ARMSE), relative Error (ReErr), and average of Err (AErr) are measured. The considered performance metrics were illustrated in Table 3. The RMSE assesses the quality of a prediction model. The use of RMSE is widespread and it makes an excellent general purpose error metric for numerical prediction. ARMSE is the average of RMSE for all datasets. ReErr gives an indication of how good a predicted value is relative to the measured value. AErr is the average of ReErr for all datasets.

The prediction system was evaluated using the evaluated dataset containing 1000 data rows from Alaska Center for Energy and Power [28]. The data is bootstrapped to 10 datasets (R = 10) and a neuro-fuzzy model is built to each of them.

The main dataset was used to create and train the neuro-fuzzy model without bootstrapping algorithm and calculate the root mean square error (RMSE) and relative Error (ReErr). The main neuro-fuzzy model was used to determine the performance of the bootstrapping technique and multiple neuro-fuzzy models. Moreover, the bootstrapping output dataset was used to create a prediction model and train it


Step-1
Load main dataset
R=10, N=200
Bootstraps data to R datasets with N rows
Step-2
Load next dataset
Start Clustering data
Step-2
Set (ANFIS) clustering parameters
Range of influence(R) =0.5
Squash factor(S_f) =1.25
Accept ratio (Acc) =0.5
Reject Ratio (Rej) =0.15
R=R-0.01, S_f=S_f-0.1
Acc=Acc+0.1, Rej=Rej-0.01
Create Clusters
C,α_{ij},σ_{ij}
Step-3
Create Fuzzy rules
Set Epochs=0
Set Epochs=Epochs+100
Start training with error tolerance *Zero*
RMSE accepted
NO
Epochs=1000
Yes
NO
Yes
No
Last Dataset
Yes
Step-4
Start Combine outputs of all neuro-fuzzy models
Step-4
Evaluate Subset and plot RMSE, ERROR, real and predicted value
END

Flowchart 2 Steps of creating 10 datasets, clustering, building neuro-fuzzy models, train, and evaluating the prediction models using MATLAB

evaluated. Each prediction model was trained on one dataset and RMSE and ReErr was calculated for each dataset. After calculating RMSE and ReErr for each data set, the average root mean square error (ARMSE) and the average of Err (AErr) were calculated as the average of the RMSE and ReErr. The RMSE for the main prediction fuzzy model was compared with ARMSE for the R pridection fuzzy models as shown in Fig. 4.

Table 4 shows the maximum and minimum RMSE for training neuro-fuzzy models on each dataset created using the bootstrap algorithm. The RMSE is calculated to 10 datasets (R = 10). Then the average of the 10 values of maximum and minimum RMSE is calculated as ARMSE. The ARMSE is ranged between 0.22712 and 0.0426 as shown in the results in this table.

Figure 4 shows the relation between ARMSE of the 10 datasets (R = 10) compared with the RMSE of the main dataset prediction model. The maximum value of ARMSE for 10 datasets is 0.22712, where the maximum value of the RMSE for the main dataset is 0.3173. The minimum value of ARMSE for 10 datasets is 0.0426, where the minimum value of the RMSE for the main dataset is 0.1966. Therefore, the prediction model using the bootstrapping algorithm minimizes the RMSE.

The real measured value of the output power and predicted power which combined from 10 predicted values was plotted and compared with the prediction value from the main dataset predicted model.

Table 5 shows the neuro-fuzzy model number and the corresponding ReErr (%) value. The ReErr percentage was calculated using eq. 7 between real WTP and the predicted power from the 10 neuro-fuzzy models. The average of error percentage (AErr %) was calculated using eq. 8 between real WTP and predicted WTP from each 10 neuro-fuzzy models with

Table 3 Metrics calculated in the experimental results section

Metrics	Description
Root Mean Square Error (RMSE)	The RMSE assesses the quality of a prediction model. The use of RMSE is widespread and it makes an excellent general purpose error metric for numerical prediction. The RMSE is calculated using (5) $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(P_{i_{meas}} - P_{i_{\mathrm{Pred}}}\right)}^{2}$ (5) Where: n is the number of rows in the dataset, P_{meas} is the real power measured, and P_{pred} is the predicted power value.
Average Root Mean Square Error (ARMSE)	ARMSE is the average of RMSE for all datasets. ARMSE is calculated using (6) $ARMSE = \frac{1}{R}\sum_{i=1}^{R}\sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(P_{\mathrm{j}_{meas}} - P_{ij_{\mathrm{Pred}}}\right)}^{2}$ (6) Where: n is the number of rows in the dataset, R is the number of datasets (10 datasets), P_{meas} is the real power measured, and P_{pred} is the predicted power value.
relative Error (ReErr)	ReErr gives an indication of how good a predicted value is relative to the measured value. The ReErr is calculated using (7) $\mathrm{Re}Err_{i}(\%) = \frac{\left(P_{i_{meas}} - P_{i_{pred}}\right)}{P_{i_{meas}}} \times 100\%$ (7) Where: I is the row number of data, $\mathrm{P_{meas}}$ is the real power measured, and $\mathrm{P_{pred}}$ is the predicted power value.
average of Err (AErr)	AErr is the average of ReErr for all datasets. AErr is calculated using (8) $AErr(\%) = \left(\frac{1}{R}\sum_{i=1}^{R}\frac{1}{n}\sum_{j=1}^{n}\frac{P_{j_{meas}} - P_{ij_{pred}}}{P_{ij_{meas}}}\right) \times 100\%$ (8) Where: n is the number of rows in the dataset, R is the number of datasets (10 datasets), P_{meas} is the real power measured, and P_{pred} is the predicted power value.

value ±5.7%. The ReErr percentage was calculated using eq. 7 between real WTP and the predicted power from the main neuro-fuzzy model with value ±15.3%. This means that the error in the prediction value of WTP using the bootstrap technique with neuro-fuzzy was reduced form ±15.3% to ±5.7%.

Table 4 The maximum and minimum RMSE for training neuro-fuzzy models on each dataset created using the bootstrap algorithm

Dataset NO.	RMSE value	
	Maximum	Minimum
1	0.255	0.055
2	0.232	0.05
3	0.203	0.047
4	0.198	0.034
5	0.211	0.031
6	0.253	0.047
7	0.2112	0.031
8	0.234	0.067
9	0.249	0.015
10	0.225	0.049
Average ARMSE	**0.22712**	**0.0426**

Figure 5 shows the real measured WTP and the average of predicted WTP for the output of the combined models which combine the output of 10 neuro-fuzzy models. The output predicted power using the main neuro-fuzzy model which created on the main dataset without bootstrapping was plotted and compared with the real measured WTP. As shown in this figure, the WTP predicted using the bootstrap neuro-fuzzy model is much near to the real WTP than the WTP predicted using the main neuro-fuzzy model without bootstrap.

The AErr (%) between real measured WTP and predicted WTP from 10 neuro-fuzzy models was calculated using eq. 8 as shown in Fig. 6. In this figure, the AErr (%) was ±7%. Figure 7 shows the ReErr (%) between the real measured WTP and predicted WTP from the main neuro-fuzzy model without bootstrap. This figure shows that the ReErr (%) was ±16%. Thus, the error in predicted WTP using the bootstrap algorithm was reduced from ±16% to ±7% compared with the predicted WTP from the main neuro-fuzzy without bootstrap.

Conclusion

The power generated by a WT generator varies randomly with time due to the variability of factors of wind (temperature, wind speed, and other factors). This change in the generated

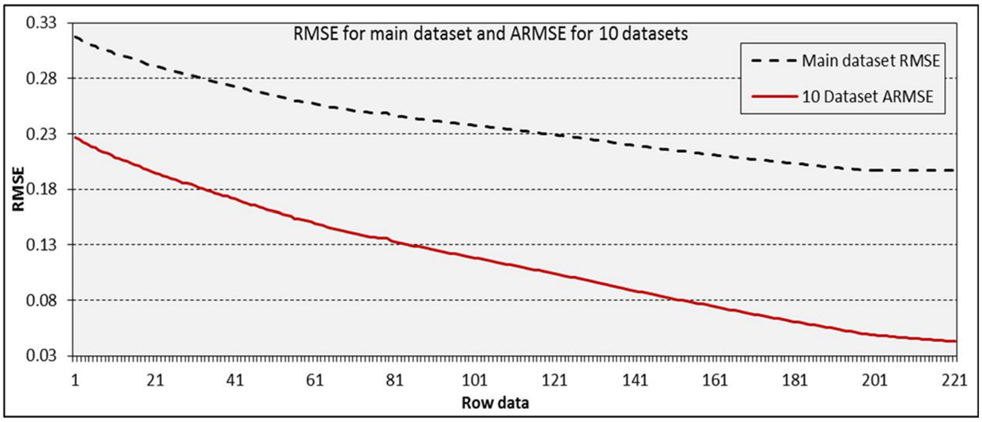


Fig. 4 ARMSE for 10 datasets compared to RMSE of the main dataset

Table 5 ReErr the predicted value from each neuro-fuzzy model

Neuro-fuzzy NO.	ReErr (%) Value
1	±5.57
2	±3.89
3	±5.98
4	±6.27
5	±5.24
6	±6.71
7	±5.76
8	±5.85
9	±5.79
10	±5.94
Average (AErr)	**±5.7**

power from WT creates challenges for the electric grid operators. One of the solutions used to deal with these challenges is to predict the power produced from the wind turbine for a short or long term. Prediction of the power to be produced from the wind turbine gives a future vision of how to deal with these sources. Many techniques were used to improve the prediction of wind turbine energy.

Therefore, the necessity and advantages of wind power short-term forecasting are generally accepted and highly evaluated by most wind energy utilities. In this paper, short-term(24 h) forecasting system is proposed for wind power forecast for the correct management of a grid-connected wind farm and to make wind energy a more competitive and reliable resource.

The bootstrap with neuro-fuzzy technique was used to implement the forecasting system of wind turbine power. The bootstrap algorithm was used to generate R datasets and the neuro-fuzzy agent in MATLAB generates a neuro-fuzzy model for each dataset. The output of the models was combined in the combination model to calculate the final predicted wind turbine power. A main neuro-fuzzy model was generated using the main data without bootstrap. This main model was used to determine the efficiency of our system by comparing the statistical parameters with it.

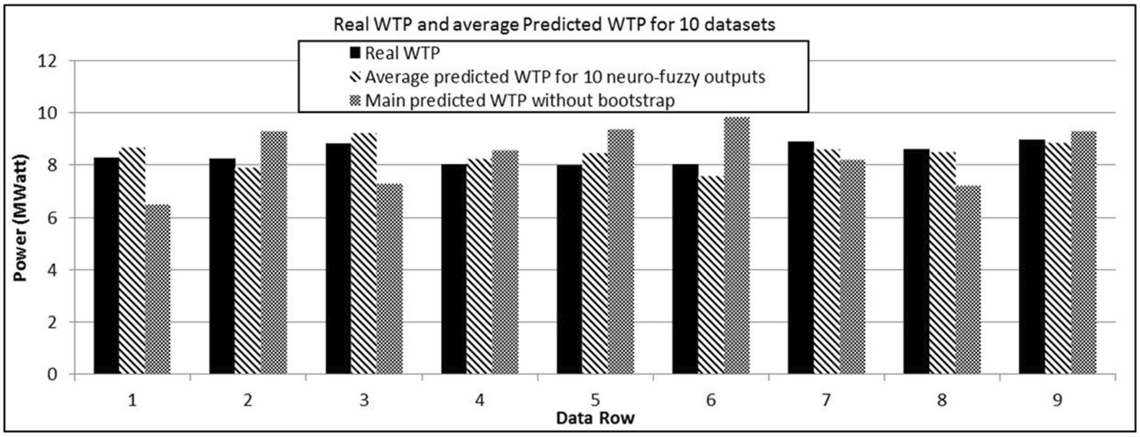


Fig. 5 Real WTP and average predicted WTP of 10 neuro-fuzzy outputs compared with the main predicted WTP without bootstrap

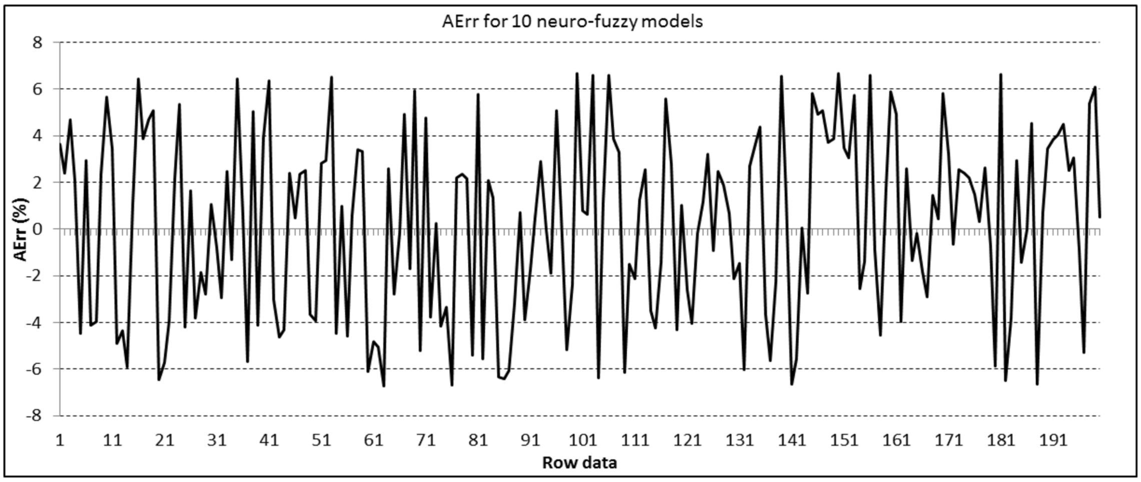


Fig. 6 AErr for 10 neuro-fuzzy models

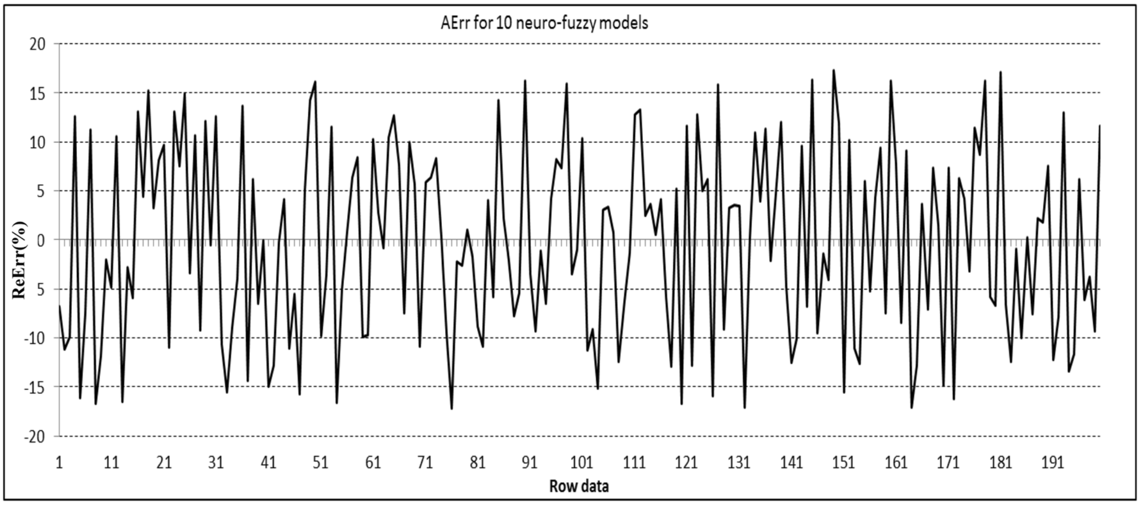


Fig. 7 ReErr for the main neuro-fuzzy model

The forecasting system was tested using MATLAB. The relative percentage error and average error for the model outputs was calculated and showed that the error was minimized to ±5.7% compared to the main neuro-fuzzy model with an error ± 15.3%. The accuracy of the prediction system shows that the prediction system of WTP for 24 h is accurate by 94.3%, where the accuracy of the prediction system using the main dataset with the main neuro-fuzzy module is 84.7%.

References

1. Natarajan VA, Karatampati P (2019) “Survey on renewable energy forecasting using different techniques”, 2019 2nd international conference on power and embedded drive control (ICPEDC), IEEE
2. Nielson J, Bhaganagar K, Meka R, Alaeddini A (2020) “Using atmospheric inputs for Artificial Neural Networks to improve wind turbine power prediction”, Energy, Elsevier, vol. 190
3. Ackermann T (2005) "Wind Power in Power Systems", Royal Institute of Technology, John Wiley and Sons Ltd
4. Lazar L, Goran P, Momčilo Z, Luka I (2014) Improved wind forecasts for wind power generation using the eta model and MOS (model output statistics) method. Energy, Elsevier 73:567–574
5. Zheng ZW, Chen YY, Huo MM, Zhao B (2011) An overview: the development of prediction technology of wind and photovoltaic power generation. Energy Procedia, Elsevier 12:601–608
6. Dou J, Liu C, Wang B (2018) “Short-term wind power forecasting based on convolutional neural networks”, 2nd international symposium on resource exploration and environmental science. IOP
7. Azimia R, Ghofranib M, Ghayekhlooa M (2016) A hybrid wind power forecasting model based on data mining and wavelets analysis. Energy Convers Manag, Elsevier 127:208–225
8. Ma L, Luan S, Jiang C, Liu H, Zhang Y (2009) A review on the forecasting of wind speed and generated power. Renew Sust Energ Rev, Elsevier 13(4):915–920
9. Eseye AT, Zhang J, Zheng D, Ma H, Jingfu G (2017) “Short-term wind power forecasting using a double-stage hierarchical hybrid GA-ANFIS approach”, IEEE 2nd international conference on big data analysis (ICBDA), IEEE
10. Zheng D, Shi M, Wang Y, Eseye AT, Zhang J (2018) “Day-Ahead Wind Power Forecasting Using a Two-Stage Hybrid Modeling

Approach Based on SCADA and Meteorological Information, and Evaluating the Impact of Input-Data Dependency on Forecasting Accuracy", Energies, IET, Vol. 10

11. Kavousi-Fard A, Khosravi A, Nahavandi S (2015) A new fuzzy-based combined prediction interval for wind power forecasting. IEEE Trans Power Syst 31:18–26
12. Okumusa I, Dinlerb A (2016) Current status of wind energy forecasting and a hybrid method for hourly predictions. Energy Convers Manag, Elsevier 123:362–371
13. Mabel MC, Fernandez E (2008) Analysis of wind power generation and prediction using ANN: a case study. Renew Energy, Elsevier 33(5):986–992
14. Kitajima T, Yasuno T (2010) Output prediction of wind power generation system using complex-valued neural network. SICE Annu Conference IEEE 10:3610–3613
15. Ren Y (2019)Short-term Power Prediction of Wind Farm Power Based on BP Neural Network. J Phys Conf Ser, IOP Science 1302:042053
16. Tahir MF, Saqib MA (2016) Optimal scheduling of electrical power in energy-deficient scenarios using artificial neural network and Bootstrap aggregating. Electr Power Energy Syst, Elsevier 83:49–57
17. Pousinho HMI, Mendes VMF, Catalao JPS (2011) A hybrid PSO–ANFIS approach for short-term wind power prediction in Portugal. Energy Convers Manag, Elsevier 52(1):397–402
18. Dantas TM, Oliveira FLC (2018) Improving time series forecasting: an approach combining bootstrap aggregation, clusters and exponential smoothing. Int J Forecast, Elsevier 34(4):748–761
19. Murata T, Muyeen SM, Tamura J (2009) "Stability augmentation of a grid-connected wind farm", Green Energy and Technology, book. Springer
20. Jinsong K, Zhiwen Z, Yongqiang L (2007) "Development and trend of wind power in China", IEEE Canada Electrical Power Conference, EPC, pp. 330–335
21. Temperature and density relations for air (2016): http://en.wikipedia.org/wiki/Density_of_air
22. Davidson R, MacKinnon JG (2002) Bootstrap J tests of non-nested linear regression models. J Econ, Elsevier 109(1):167–193
23. Kotsiantis S (2010) Combining bagging, boosting, rotation forest, and random subspace methods. Artif Intell Rev 35:223–240
24. Jang J-SR, Sun CT (1997)Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence. IEEE Trans Auto Control (book review) 42(10):1482–1484
25. Saleh I, Alhady SSN, Rahiman W (2016) Comparison of Mamdani and Sugeno fuzzy logic performance as speed controller", 9th international conference on robotic, vision, signal processing and power applications, Springer, vol. 398, Pp. 369–376
26. Precup R-E, Hellendoorn H (2011) A survey on industrial applications of fuzzy control. Comput Ind, Elsevier 62:213–226
27. Schlechtingen M, Santos IF (2014) Wind turbine condition monitoring based on SCADA data using normal behavior models. Part 2: Application examples. Appl Soft Comput, Elsevier 14:447–460
28. Alaska Center for Energy and Power (2016): http://islandedgrid.org/?directory=alaska-center-for-energy-and-power-acep
29. Linkens DA, Chen M-Y(1999) Input selection and partition validation for fuzzy modeling using neural network. Fuzzy Sets Syst, Elsevier 107(3):299–308
30. Wang L-X, Mendel JM (1992) "Back propagation fuzzy system as nonlinear dynamic system identifiers", 1ST IEEE Int. Conf. on Fuzzy Systems, pp. 1409–1418