Application of Artificial Neural Network and Genetic Algorithm to Forecasting of Wind Power Output

Lin Tzu Chao

Master's thesis

University of Jyväskylä

Department of Physics

Master's Degree Programme in Renewable Energy

Supervisor: Dr. Jussi Maunuksela

3/8/2007

Preface

For several years, renewable energy has been utilized for easing the green house effect. The precise forecasting of renewable energy supply will help the promotion of renewable energy in open electricity market. Owing to this, we surveyed some forecasting models proposed and we found that artificial neural network performs well in nonlinear mapping. By combining artificial neural network with the genetic algorithm, this model can perform efficiently and accurately. We implemented this hybrid model and utilized the operational records of commercial wind turbines to observe the performance of this model.

I completed this work with the help of many people. Here I would appreciate my supervisor, Dr. Jussi Maunuksela. He led me through the whole work and instructed me about the research. With his great help, this work is finally done. I would also like to thank my colleagues, Su Yu-Cheng, Su Ging-Long and Chang Tzu-Gie, in Taiwan Power Company. Their kind help and supports in obtaining wind power operation records made my work proceed with smooth steps. Finally, I would like to share this achievement with my dear wife, Chiang Yu Chin. She takes care of everything well in my family, so I can focus on this work totally.

If you have any question about this work, please do not hesitate to inform me by email. I will try my best to answer any question.

In Jyväskylä 3/8/2007

Lin Tzu Chao
Renewable Energy Master's Programme
u772020@yahoo.com.tw

Abstract

In this work a hybrid wind power forecasting model using artificial neural network and genetic algorithm is implemented by Matlab coding. The genetic algorithm was used for optimizing the connecting weights and thresholds in the artificial neural network. The performance of the forecasting model was tested and analyzed with wind power data from the operation records of two commercial wind power stations in Taiwan. The hybrid model was constructed with help in fitting extracted data by the Boltzmann function. Numerical results show that this model can perform power output forecasting well except during the occurrences of gust. Based on this forecasting model, forecasting in different time-scales can be implemented which can make the wind energy production more competitive in open electricity markets.

Table of Content

1. Introduction 8

2. Theoretical Background 10

3. Methods and Processes 13

3.1 Artificial Neural Network 13

3.2 Genetic Algorithm 15

3.3 Process 18

3.4 Data Preprocessing 20

3.5 Training 22

3.6 Forecasting 23

3.7 Evaluation 24

4. Wind power stations 25

4.1 Chung Tung Wind Power Station 25

4.2 Heng Chun Wind Power Station 28

5. Results 29

5.1 Chung Tung Wind Power Station 29

5.2 Heng Chun Wind Power station 35

6. Discussion and Conclusion 42

6.1 Discussion 42

6.2 Conclusion 43

7. References 44

Appendices 46

A. windforecast01.m 46

B. RegAna.m 47

C. extractdata.m.....48

D. ANN.m 49

E. GA.m 50

F. FUN.m.....52

G. output.m.....53

H. drawplot.m.....56

Table of Figures

Fig. 1 Schematic representation of an artificial neuron 13

Fig. 2 The structure of a ANN, where w_{ij}^{l} is the weight of a neuron (i, j) at a layer l 14

Fig. 3 The chromosome structure 16

Fig. 4 Genetic Algorithm optimization process.....17

Fig. 5 GANN wind power forecasting process.....19

Fig. 6 GANN forecasting data flow.....20

Fig. 7 Data preprocessing 21

Fig. 8 Training Process 23

Fig. 9 Forecasting processing 24

Fig. 10 Distribution of WTGs in Chung Tung Wind Power Station 26

Fig. 11 Birdview of WTGs in Chung Tung Wind Power Station.....26

Fig. 12 Nacelle structure of Enercon E40.....27

Fig. 13 Location of 3 WTGs in Heng Chun wind power station.....28

Fig. 14 Nacelle Structure of GE 1.5se WTG (Source: GE training material) 29

Fig. 15 Chung Tung WPS wind speed and power output in November 2006.....30

Fig. 16 Scatter plot of power output vs. wind speed in November 2006.....30

Fig. 17 Scatter points of normalized data and the sigmoidal fitting curve 31

Fig. 18 Optimization process of GA 31

Fig. 19 Power output and training result of Chung Tung WPS.....32

Fig. 20 Training deviation of Chung Tung WPS.....32

Fig. 21 Chung Tung WPS wind speed and power output in December 2006 33

Fig. 22 Scatter points of power output vs. wind speed in December 2006 34

Fig. 23 Forecasting power output and the actual power output of Chung Tung WPS 34
Fig. 24 Forecasting deviation of Chung Tung WPS 35
Fig. 25 Training wind speed and power output of Heng Chun WPS 36
Fig. 26 Scatter points of power output vs. wind speed of Heng Chun WPS 36
Fig. 27 Scatter points of normalized data and the sigmoidal fitting curve 37
Fig. 28 Optimization process of GA 38
Fig. 29 Power output and training result of Heng Chun WPS 38
Fig. 30 Training power deviation of Heng Chun WPS 39
Fig. 31 Testing wind speed and power of Heng Chun WPS 39
Fig. 32 Scatter points of power output vs. wind speed 40
Fig. 33 Forecasting power output and the actual power output of Heng Chun WPS 40
Fig. 34 Forecasting deviation of Heng Chun WPS 41

1. Introduction

Global climate change has urged the utilization of renewable energy sources all over the world. Shortage and rising price of fossil fuels have become more and more apparent while the energy demand has been growing. So it has become urgent to secure an abundant supply of clean energy. In this situation, renewable energy is a good option for the future. With renewable energy we mean the energy derived from various forms of natural processes. It often derives from the sun or the heat within the earth and it includes solar, wind, biomass, geothermal, hydropower and ocean resources. During the last several years, the technology and the production of renewable energy have improved and, therefore, renewable energy sources are utilized for electricity production in larger scale.

The availability of renewable energy connected to electrical network is normally difficulty to forecast in open electricity markets. Because the price of electricity in exchange is decided by the future production of electricity, imbalance in demand and supply needs to be avoided. It is important for the producers of renewable electricity, *i.e.*, electricity supplied from renewable energy sources, to use strategic methods in their production. The electricity generation for the open electricity market in the future typically refers to the production in the next 24-72 hours. Because of the uncertainty in Nature, the wind production, which mainly depends on the wind speed, is frequently changing. Therefore, it will be more beneficial for the producers to forecast the renewable power production according to some crucial parameters. The purpose of this work is to implement such a generic process to forecast the wind power output that is based on the artificial neural network combined with the genetic algorithm.

In this work, the theoretical background is reviewed in Chapter 2, where several forecasting methods and models are discussed and their performance summarized, giving a rough understanding about different forecasting techniques. The artificial neural network and the genetic algorithm are introduced in Chapter 3 and applied to the forecasting process in the hybrid model. There are descriptions of the two wind power stations in Taiwan and the technical operation parameters of the wind turbines operated there in Chapter 4. In Chapter 5 the operation records of those wind power generation stations are used to realize the

performance of the hybrid model. At last the discussion about forecasting results and the conclusions of this work are presented in the Chapter 6.

2. Theoretical Background

First we define the meaning of forecasting based on time series. A time series is normally a sequence of data $\{x_1, x_2, x_3, \cdots, x_n\}$ of some physical quantity measured at regular intervals of time. When we talk about time series analysis, it usually includes three specific problems: forecasting, modeling, and characterization [1]. The goal of forecasting is to precisely predict the short-term evolution of the system, the aim of modeling is to precisely capture the features of the long-term behavior of the system, and the purpose of characterization is to identify parameters or properties of the system. In this work we will focus mainly on the forecasting problem of time series.

Both modeling and forecasting of time series involve mapping time series which means simulating the system behavior in time domain. It is easy for a linear system by adjusting parameters of curve fitting. Nevertheless, it will become more complex when the system becomes nonlinear and we need to apply identification methods to simulate or resolve the mapping of system in order to forecast the future behavior of the power output model.

Mathematically, forecasting the future of time series is to find some nonlinear mapping F_p with parameters as

$$\hat{x}(t+\tau) = F_p\{x(t), x(t-\Delta), \cdots, x[t-(n-1)\Delta]\}, \quad (1)$$

where Δ is a lag time and n is an embedding dimension. Equation (1) implies that an estimate $\hat{x}$ at the time $(t+\tau)$ can be obtained from the unknown mapping F_p with a proper combination of n points of the time series spaced Δ apart.

Normally, it is very difficult to derive an exact mathematical model for mapping due to the nonlinearities embedded in the time series. The forecasting techniques used for mapping nonlinear systems include methods such as linear regression [2-4], time series model [5-8], and autoregressive moving average method (ARMA) [9]. Linear regression method utilizes the common relationships among all variables in the model to predict the relative change of one variable by the change in other variables. When the relationship among the reference control variation is unclear, it will cause the forecasting error. In time series model, it is assumed that the power output is a stationary time series and has normal

distribution characteristics. Studies show that the error of forecast increases when the historical data does not support this condition. Finally, the ARMA model is known to be not efficient in modeling discrepancies in data, *i.e.*, weekends, holidays, and seasonal change periods [10]. The advantage of these models is the easy physical interpretation of model parameters. Also, in order to speed up the ANN process and to promote its accuracy many optimization techniques were proposed to be combined with the ANN.

Many studies are focusing on modern forecasting techniques based on artificial intelligence (AI) and artificial neural network (ANN) that has received most of the attention [11-20]. These works utilized ANN as a black box tool to identify the power output behavior. Because ANN has the good mapping ability in nonlinear function approximation, it can also perform forecasting. However, many optimization techniques combining ANN were proposed in order to speed up the process and to promote accuracy.

In the field of optimization methods, genetic algorithm (GA) is the most popular topic [1, 10, 21]. The GA is a directed random search technique for optimization problems where the number of parameters is large and the analytical solutions are difficult to obtain. The GA can assist in finding the optimal solution globally over a domain. It has been utilized in different areas such as fuzzy control, path planning, modeling and classification *etc.*. The GA optimization process consists of searching a group of feasible solutions. This search goes along the orientation of general optimum avoiding the local minimum. The ability of GA in training the ANN for short-term forecasting was shown in [21]. This gave us an idea about its applications to wind production forecasting. By referring to [17], we implemented a model that combines GA with ANN in forecasting the wind power output.

By utilizing the nonlinear mapping ability of ANN and the ability of GA to find the optimum solution, the hybrid model is constructed. The parameters of ANN will be optimized by GA with Roulette wheel selection, Gaussian mutation and random mutation. Then, we collected the operation records from two commercial wind power stations and fitted a power curve to the extracted dataset. We utilized the Boltzmann equation to fit the data rather than the polynomial

function that has been used before. The forecasting performance in these two commercial wind power stations was evaluated by calculating the variance of the actual and forecasting power output.

3. Methods and Processes

It is known that the behavior of the wind power is highly nonlinear, and we think that a properly constructed ANN can be used to model it and to forecast the future output power. With the GA optimization, ANN can search the optimum solution of any system efficiently and precisely. Our forecasting model is a hybrid optimization model combining a genetic algorithm (GA) and an artificial neural network (ANN).

3.1 Artificial Neural Network

The operation of artificial neural network simulates the function of human nervous system. The basic element of an ANN, called neuron, is shown in Fig. 1. The weighted summation function of inputs x_i and weights w_i is denoted as:

$$u_k = x_1 w_1 + x_2 w_2 + \cdots + x_n w_n = \sum_{i=1}^{n} x_i w_i \,. \tag{2}$$

The activation function of the neuron is used as the transfer function to stimulate the response as human neuron does. Normally, we adopt different types of activation functions such as threshold function, piecewise-linear function, and sigmoid function in accordance with the problem. The neuron output as a mathematical operation is:

$$y_k = \varphi(u_k - \theta_k), \tag{3}$$

where φ is the activation function, u_k is the weighted sum of the inputs, and θ_k is the threshold.

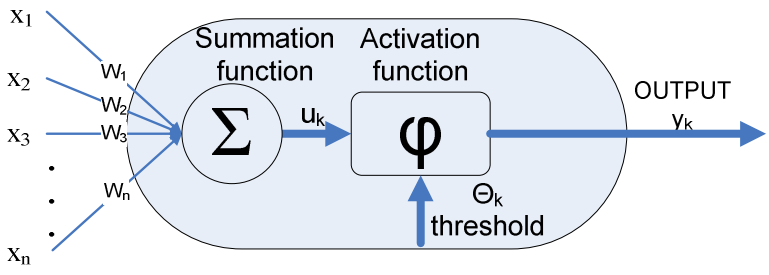

Fig. 1 Schematic representation of an artificial neuron

When we collect many groups of neurons in the structure as in Fig. 2, the whole framework can perform a human learning operation. Multilayer feed-

forward neural network, or back propagation artificial neural network (BPANN), is one of the most commonly used networks in many applications. The normal structure of ANN is composed of input, hidden, and output layers. In input layer, the neurons accept the outside stimulation as input. By inputting the training data, the experience values in the network are changed by the training. Such a "learning experience" is stored in thresholds and weights. Normally, these inputs are often critical parameters that affect the system. For example, in wind power forecasting, the wind speed and direction will be the critical parameters and the respective output is the power generated by the turbine.

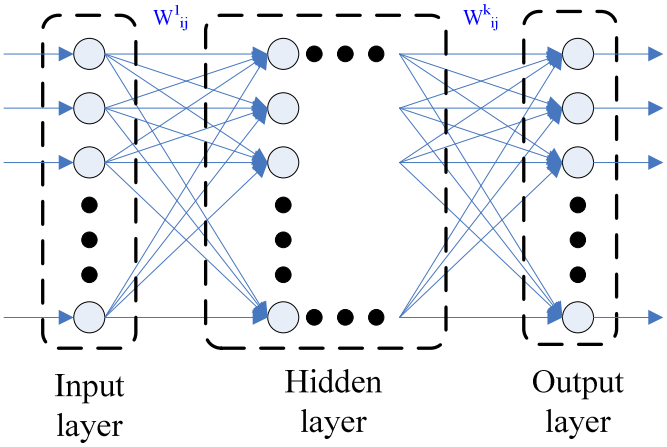

Fig. 2 The structure of a ANN, where w_{ij}^{l} is the weight of a neuron (i, j) at a layer l

The goal of the learning process is to modify the parameters (weights and thresholds) in the network for optimum fitting of real output. The connections between each layer and the thresholds in neurons are often initially random values between -1 and 1. These values will be taken as the experience for weighting the input and threshold in each neuron. By adopting a transformation function, these summation values will be decided through the function, transferred to the next layer, and vice versa. The final output of ANN will be compared with the desired output.

In this work, we apply a three-layer feed-forward network to our model. Then ANN will modify the weights and thresholds to learn the optimum behavior of the system by many epochs of learning samples. The learning method used during the process is gradient descent with moment weight change and historical

learning experience. The calculation of the weight adjustment for a given neuron (i, j) at layer k is denoted as:

$$\Delta W_{ij}^{k}(t+1) = \eta \Delta W_{ij}^{k}(t) + \alpha \Delta W_{ij}^{k}(t-1), \tag{4}$$

where $\Delta W_{ij}^{k}(t+1) = W_{ij}^{k}(t+1) - W_{ij}^{k}(t)$, η is the learning rate and α is the momentum term which is the coefficient of the last modification.

There are several things worthy to notice about the parameters of ANN. First, the number of hidden layers is often decided on the complexity of system. If the system is highly nonlinear and more complex, there could be two or more hidden layers. However, some researchers have revealed that this will cost more training time and not give better results than what is given by adding more neurons in a single hidden layer. We can see that the number of neurons in the hidden layer is a parameter which affects the performance of ANN. According to the experimental results [17], ANN can handle with highly nonlinear model very well after a training process with enough connections and learning epochs.

In the training process of ANN, there are two modification methods of the interior parameters: batch and sequential learning. Normally, the modification of parameters in ANN is done after accumulating the modification of all training datasets in batch learning. Such a training style often gives ANN an overall learning about the whole dataset. Unfortunately, it will miss some of the cyclic or repeated behavior characteristics of the system. On the other hand, sequential learning means that the network performs learning as each single data is inputted. In short, the modification performs in sequence by each data. This learning style can easily handle with different nonlinear problems, but it consumes more training time.

3.2 Genetic Algorithm

Genetic algorithm (GA) is a common tool in optimization engineering and was originally proposed by Professor John Holland and his students in 1970 [22]. The development of the GA was based on ideas proposed by Charles Darwin in his book “On the Origin of Species by Means of Natural Selection” or in the “Preservation of Favoured Races in the struggle of Life” (1859). The algorithm simulates the competition among species where the winners will propagate. There

are several steps in the GA to proceed searching the optimum solution. A normal GA process is shown in Fig. 4 and there is a brief introduction about the process following.

When we form the optimization problem, the space of variables should be defined. A value of one variable is often looked as a “gene” and the composition of these genes will form a “chromosome” (Fig. 3) as a point in hypercube space. Each chromosome represents a solution to our problem and our goal is to find the best solution, which is given by the optimum of the objective function (lowest/highest). There will be a group of chromosomes, or an initial population, randomly generated within the predefined range in the beginning. In general, there are two ways real and binary to transform variables as gene in chromosome. In binary type chromosome, the variables will be transformed as the combination of “0” and “1”. But the variables will remain the same in the real type chromosome. This depends on the nature of the problem and, thus, we will utilize real formed chromosome in our work.

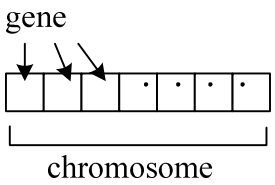

Fig. 3 The chromosome structure

After the population is generated, the values of the objective function of these chromosomes are evaluated, and then the GA process will proceed as shown in Fig. 4. Some of the chromosomes with a good objective solution will be selected into the mating pool that is completed with others selected according to some rules like Roulette wheel selection or ranked rule. We will choose ten best fitted chromosomes into the mating pool and others are selected by Roulette wheel selection.

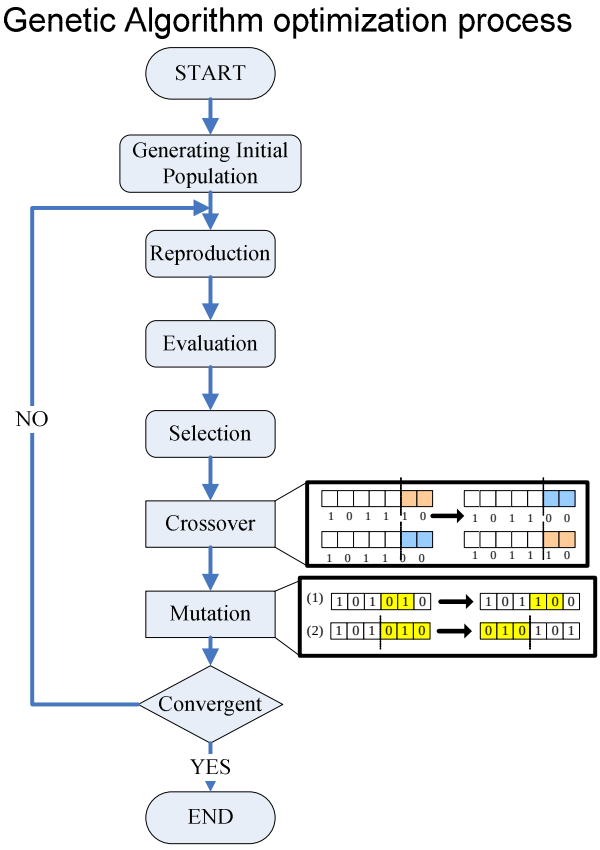

Fig. 4 Genetic Algorithm optimization process

Then GA will choose the chromosomes from mating pool for the crossover of the genes. There are several methods of crossover such as one-point, multi-point. The locations of a crossover in the chromosome are normally random. After crossover, the GA will perform the mutation of chromosome where the genes will be modified. There are some ways of mutation like Gaussian, random mutation *etc.*. The location of the chromosome is randomly selected by probability, and several studies revealed that this probability should not be high, or the search direction will be divergent. Either crossover or mutation makes a next generation offspring from the mating pool, prevents the searching from falling into a local optimum and searches the optimum multi-dimensionally. When the offsprings are generated, the objective fitness will be evaluated to check if the optimum is reached or not. If it is not, the same process will repeated again.

In general, the GA is a method based on natural selection for solving optimization problems, and the process is imitating the course of evolution. The process modifies a population of individual solutions repeatedly. The overall

effect of the GA is that a new population moves toward better solutions with fitness values of objective function. We often apply the genetic algorithm to solve various optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, nondifferentiable, stochastic or highly nonlinear.

This work will use GANN to denote the model of ANN in which the parameters are optimized by the GA. In such a model, a training dataset is first preprocessed and then inputted into the GANN. According to the desired output, the square error between output and target is calculated and then the total sum of these square errors will then be the objective fitness function. For searching the optimum, the GA will be applied for the solution.

3.3 Process

An adaptive artificial neural network (ANN) combined with the genetic algorithm (GA) was used to model mapping between wind speed and wind turbine output power. The learning step and the momentum coefficient in ANN were decided manually for steepest descent direction in the optimization. ANN learns the mapping relation by adjusting internal connecting weights and threshold and then searches the minimum square error.

Even though the ability of ANN has been proven in nonlinear mapping, the experience shows that it will perform forecasting more accurately if as many characteristic features as possible are extracted from raw data for training. In fact, it was very important to manage the learning dataset from raw operation records. Some pause states of the commercial wind turbines with no or minus power output are not reasonable, though the wind speed is high enough for wind turbine to work. They are sometimes caused by the maintenance or because of the protective shutdown due to a sudden overspeed gust. The operational data from this period is nonsense and will disturb the learning or the optimization search. It is recommended to remove these data in advance for better optimum result.

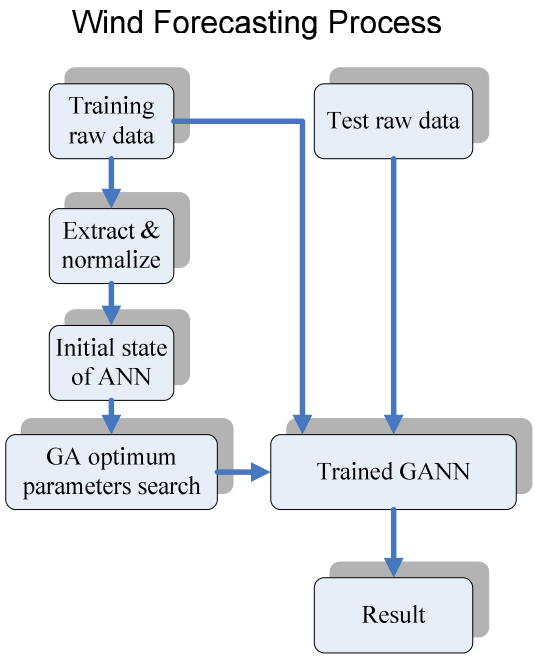

Fig. 5 GANN wind power forecasting process

The flow chart in Fig. 5 demonstrates the process of the forecasting model. The trained GANN model will be directly implemented to forecast wind power output from training and test raw data. In addition, the overall data flow in Fig. 6 denotes the total operation and data process in the simulation. These Matlab programs are documented in Appendices and the data flow between the Matlab programs is explained in the following sections.

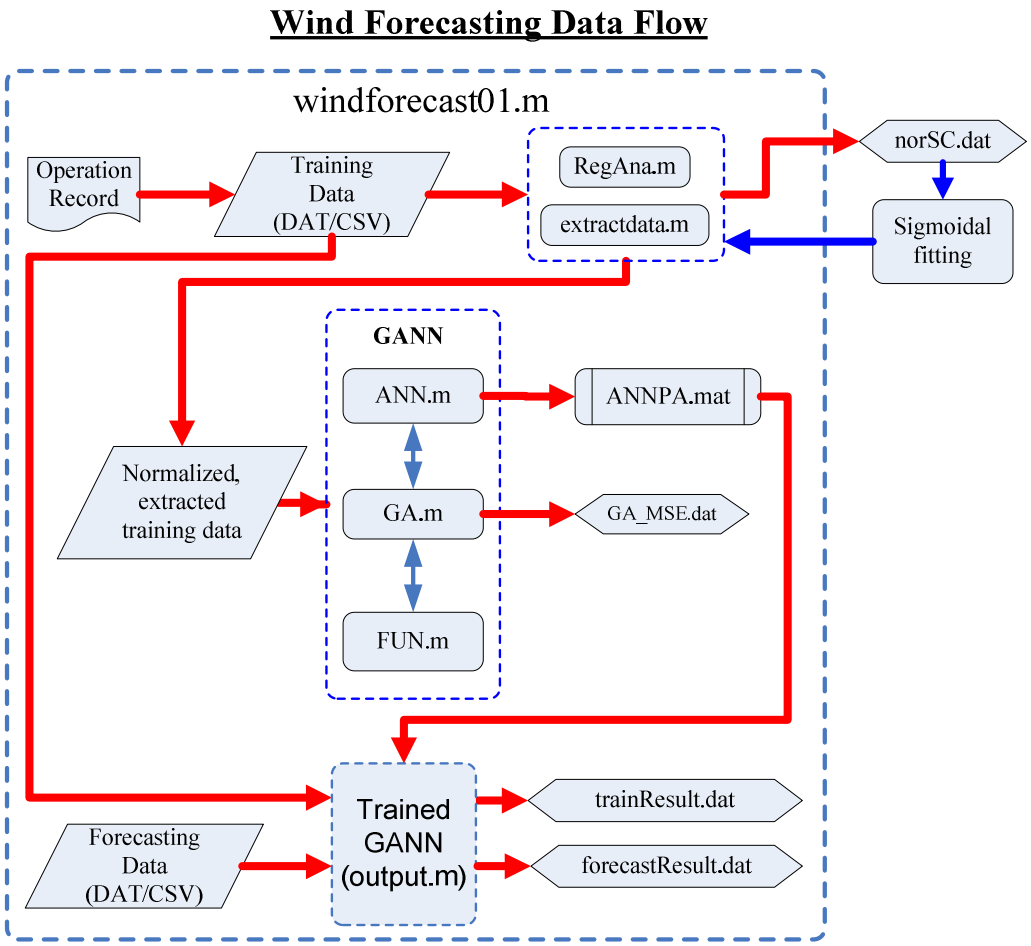

Fig. 6 GANN forecasting data flow

3.4 Data Preprocessing

When we get the operation records from wind power stations, the raw data should be preprocessed to get rid of abundant or unrequired data (Fig. 7). Because the SCADA (Supervisory Control And Data Acquisition) will record detailed data sequentially even in the maintenance state or shutdown by sudden gust, the raw data are not helpful for the training of GANN.

Data Prepocessing

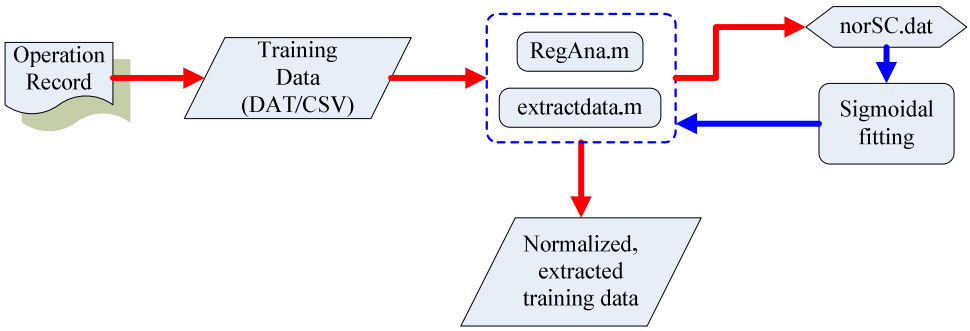

Fig. 7 Data preprocessing

The raw data comes into the data preprocessing where the characteristic data will be extracted. The characteristic data (v, p) is used to characterize the wind turbine by the following criteria:

1. $0 \le v \le v_{cutout}$ *and* $0 \le p \le P_{\max}$ AND
2. $v_{nom} \le v < v_{cutout}$ *and* $p \ge 0.9P_{nom}$ AND
3. $v_{cutin} \le v < v_{nom}$ *and* $p \ne 0$.

This prevents the noise signal to disturb the learning of GANN model.

Moreover, the extracted data still cannot be fed directly into the model, because the range of the data is large when considering different operation parameters of the wind turbine. This problem can be solved by normalizing the data to uniform space mapping. Therefore, operation parameters of the specific wind turbine should be known in advance.

Finally, the extracted and normalized operation data of wind turbine are obtained so that they are ready to be used in the training process. However, it was proposed in [23] that curve fitting can be utilized to present the characteristic behavior of wind turbine. Such an idea of curve fitting is adopted and it is modified as the sigmoidal fitting of the Boltzmann function (Eq. 5) rather than the more common fitting of the polynomial function:

$$f(x) = A_2 + \frac{A_1 - A_2}{1 + \exp[(x - x_0)/\Delta x]}, \tag{5}$$

where x_0 is the enter value of the exponential distribution, Δx is the width of the exponential distribution, A_1 is the initial value of $y(-\infty)$, and A_2 is the final value of $y(+\infty)$.

In general, when the raw data was collected, the necessary columns of data were stored in files using comma-separated values (CSV) format or in ASCII column formed files with sub filename “DAT”. The Matlab M-file “RegAna.m” deals with the function file “extractdata.m” to extract the useful data from the raw data and normalizes each column of data according to the operational parameters of wind turbine like the cut-in wind speed v_{cutin}, the nominal wind speed v_{nor}, the cut-out wind speed v_{cutout} and the nominal power output P_{nor}. The extracted and normalized data will be stored in ASCII format in “norSC.dat” for Boltzmann function fitting which is performed in OriginPro, a scientific graphing and analysis software of OriginLab Co.. We found that this fitting is more suitable for wind power curve. There are four coefficients in the function and they will be inputted in the popup windows by “RegAna.m” to form the fitting function line. Then the regular training data in lag of 0.5 on the fitting line are managed to form training dataset.

3.5 Training

In this process, the GANN model will be trained by the extracted dataset. The main part of the GANN model is founded by the three Matlab M-files: “ANN.m”, “GA.m” and “FUN.m”. The training process of the model is shown in Fig. 8. At first, the training data will be fed into the model and the total square error of all training data will be calculated as batch learning. The goal of the GA is to search the optimum parameters of ANN that minimize the total square error of the training data.

Second, the GA utilizes the Roullete wheel selection, real valued intermediate crossover, and the mutation of Gaussian and random. In Gaussian mutation GA performs mutation by the Gaussian probability distribution function and it enforces the accuracy of each gene in chromosome by shortening the range. In random mutation, each gene mutates with a random probability which enforces

the optimization. The number of chromosomes in “GA.m” is set as 150 to search the optimum and it will search 100 epochs. Theoretically, the smaller chromosomes will save more running time, but it affects the searching dimension of optimum solution and it prevents the searching from getting stuck into a local minimum.

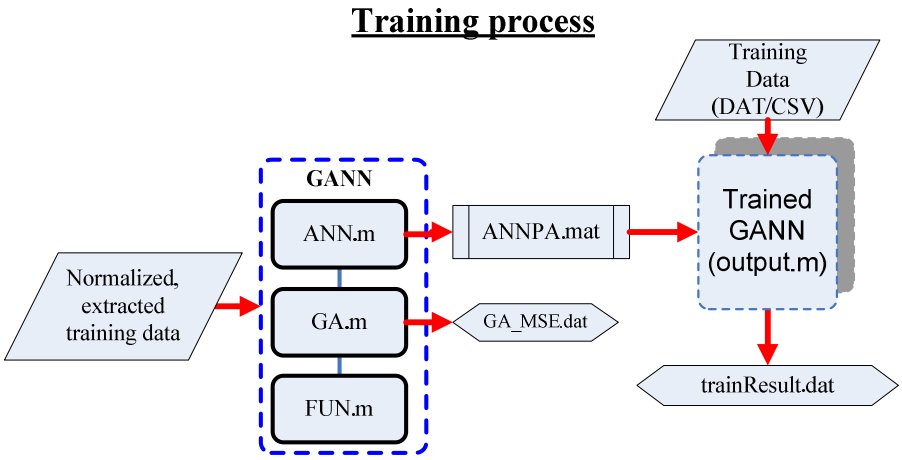

Fig. 8 Training Process

After the training process of GANN is finished, the final trained values of the connecting weights and thresholds of ANN are saved in “ANN.mat” and it is ready for “output.m” that checks the final performance of the training. The output file of “output.m” is “result_train.dat” where records of the real output and forecasting output of GANN are stored.

3.6 Forecasting

When the GANN model is trained, it is ready to be used for forecasting. Just as the GANN is trained by the training dataset, the forecasting dataset will be facilitated with “ANN.mat” to perform the forecasting process (Fig. 9) in “output.m”. The result is saved as “result_forecast.dat” from “output.m” as mentioned previously.

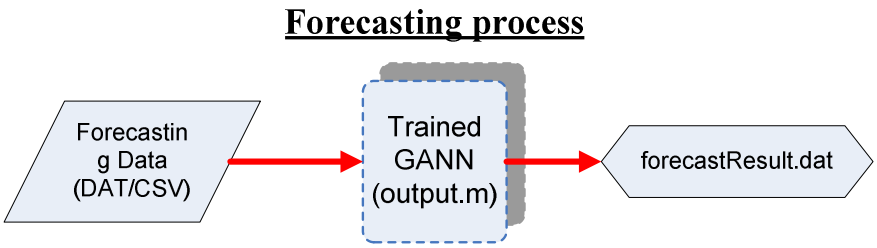

Fig. 9 Forecasting processing

It should be noticed that the output of GANN will be managed in reasonable range. Because of the prior knowledge of the maximum power output of wind turbine, the forecasting output of GANN maximum power output is limited. It is the same as in minimum output, though the minus power output sometimes happens on maintenance state. For those possibilities and realization, the training output or forecasting output will be limited in the prescribed range in which minimum is zero and maximum is set as the real maximum power. The result of forecasting will be evaluated to observe the performance of GANN in final.

3.7 Evaluation

The trained GANN as well as the optimized parameters are evaluated using either the training or the test dataset to observe the forecasting results. There are some evaluation indexes to evaluate the forecasting performance (Table 1). Normally, target and output in first two columns will be utilized to plot the completed forecasting results which help to recognize the power output trend and the curvature change. Then, the relative deviation of these two outputs will be demonstrated to identify the forecasting error. Such error will clearly show the variation of forecasting error in time series.

Table 1 Evaluation indexes

Target P_{OUT}	Output $P_{FORECAST}$	Error	Absolute Error
a	b	c a-b	d abs(c)

4. Wind power stations

Since 2003, the government of Taiwan has been promoting wind energy as an important source of renewable energy for the electricity supply. There have been more than five wind power stations (WPS) in commercial operation so far. However, each wind power station in Taiwan is installed with different brand or different power output depending on local wind energy potential.

In this work, two wind power stations were selected to be used for the wind power forecasting. The capacities of these two commercial wind power plants are shown in Table 2. Chung Tung wind power plant has eight Enercon's 600 kW wind turbines (E40), while Heng Chung has three GE's 1.5 MW turbines (GE 1.5se). The goal of this modeling is to investigate the accuracy and availability of GANN for real, commercial forecasting of wind energy production.

Table 2 Operation parameters of each WPS

Operation Parameters	Chung Tung Wind Power Station	Heng Chun Wind Power Station
Power Output	600 kW x 8 (Enercon E40)	1500 kW x 3 (GE 1.5se)
Transmission	gearless	gear box
Cutin wind speed (m/s)	2	4
Nominal wind speed (m/s)	12	12
Cutout wind speed (m/s)	28-34	25

To demonstrate the forecasting characteristics of our GANN model, the wind operation records from these wind power stations were selected to demonstrate the forecasting performance. Each wind power station is introduced separately with prevailing wind characteristics. Then, the operation records are used to implement the GANN, and the results will be demonstrated.

4.1 Chung Tung Wind Power Station

Chung Tung wind power station is located in PengHu which is a county composed of small islands in the western side of Taiwan. There is no mountain higher than 100 meters on most islands. Every year in September, the northern east season wind will start to blow until next spring comes. Because of abundant wind energy in this area, Taiwan Power Company set up a wind power station

composed of eight wind turbines on the ChungTung (Figs. 10 and 11) that are connected to the isolated electricity network.

Fig. 10 Distribution of WTGs in Chung Tung Wind Power Station
(Source: Taiwan Power Company Chung Tung WPS demo material)

Fig. 11 Birdview of WTGs in Chung Tung Wind Power Station
(Source: PengHu National Scenic Area Administration)

These Enercon E40 wind turbines were manufactured by a Germany company, Enercon GmbH, and the nominal power output of each wind turbine is 600 kW (Fig. 12). The cut-in wind speed is 2 m/s, the nominal wind speed is 12 m/s, and the cut-out wind speed is 28-34 m/s. The height of the turbine from ground to the center of rotor is 46 meters, and the diameter of the rotor is 44 meters. The wind turbine is a gearless wind turbine with three-blade rotor. Pitch angle of the blades is controlled by pitch motors to get optimum wind energy depending on the wind speed. Meanwhile, the direction of nacelle is driven by four motors to keep it up-wind.

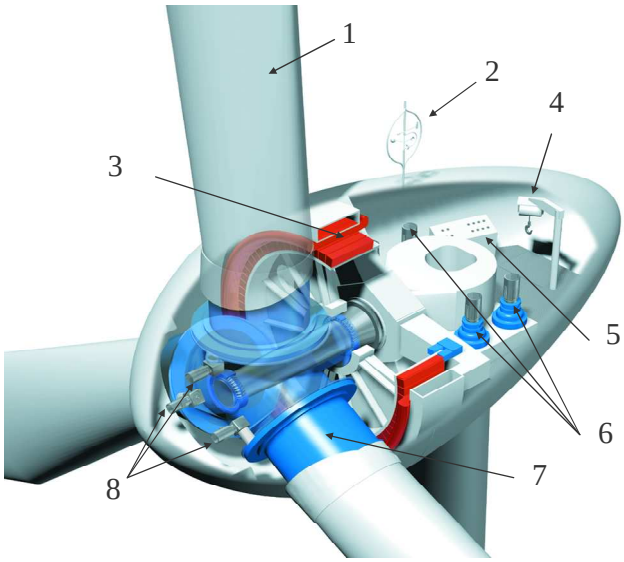

Fig. 12 Nacelle structure of Enercon E40

(Source: Enercon training material)

The acquired operation records of the wind power station were collected between October 24th and December 20th 2006, total of 58 days. The data on the first and the last day of the period is not complete. The records were collected by SCADA every 10 minute and they consisted of time, wind speed (average, maximum and minimum), rotational speed of rotor (average, maximum and minimum), power output (average, maximum and minimum), nacelle position, *etc.*. For this work, the dataset for November was first extracted, and then normalized by the nominal wind speed (12 m/s) and the nominal power output (600 kW). A curve was fitted into the normalized data with a sigmoidal fitting of the Boltzmann function. The training data is composed of sets of data with three consecutive wind speeds that were normalized and two consecutive power outputs of the fitting function as follows:

$$\begin{aligned} &\text{Dataset: } [v(t), v(t+\Delta), v(t+2\Delta), p(t), p(t+\Delta), p(t+2\Delta)], \\ &\text{where input is } [v(t), v(t+\Delta), v(t+2\Delta), p(t), p(t+\Delta)] \text{ and} \\ &\text{output is } [p(t+2\Delta)]. \end{aligned} \tag{6}$$

The testing dataset, *i.e.*, the data between December 1st and 20th is composed in the same way for forecasting.

4.2 Heng Chun Wind Power Station

Heng Chun wind power station is located on a small hill beside the Nuclear Power III Plant in the southern Taiwan (Fig. 13). This WPS comprises three GE 1.5se 1500 kW wind turbines. Every year the southern west wind in summer and northern east wind in winter contribute to the wind energy potential in this area. Nevertheless, there is often sudden gusts with wind speeds over 25 m/s that cause the wind turbine to shutdown for protection.

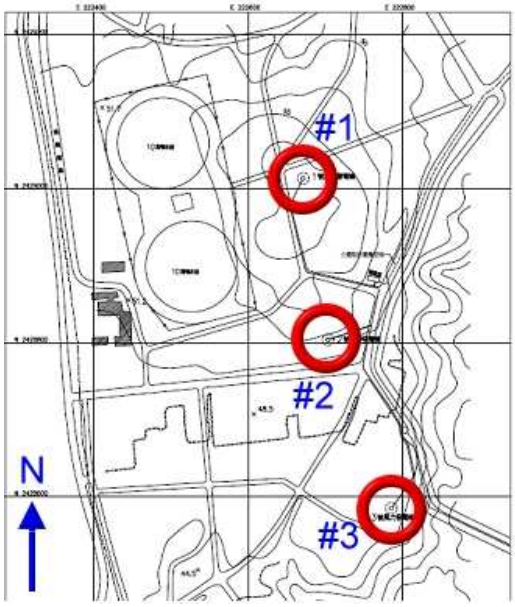

Fig. 13 Location of 3 WTGs in Heng Chun wind power station
(Source: Taiwan Power Company Heng Chung WPS construction drawing)

GE 1.5se is a wind turbine with three-blades rotor that is 70.5 meters in diameter. The height of the WTG is 64.7 meters from the ground to the center of hub. When it is operating, the wind energy will transform into mechanical form through main shaft and gearbox (Fig. 14), and then it transforms into electricity by generator. The nominal power output of generator is 1500 kW with rated voltage 575 V and rated frequency 60 Hz. The cut-in wind speed is 4 m/s, the cut-out wind speed is 25 m/s, and the nominal wind speed is 12 m/s.

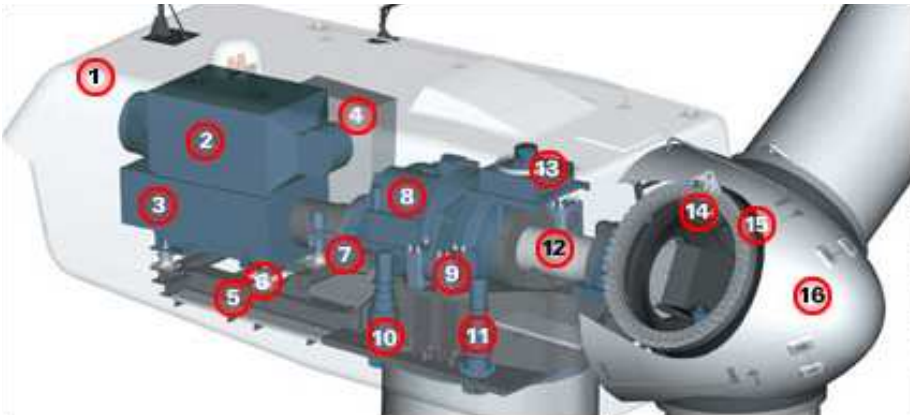

Fig. 14 Nacelle Structure of GE 1.5se WTG (Source: GE training material)

5. Results

We will implement the forecasting model and show the simulation results in this section. There are two stages in each case, *i.e.*, the training and the testing. The training result is the demonstration of the hybrid model after training. The testing result is the check of the forecasting with the trained model by other untrained dataset.

5.1 Chung Tung Wind Power Station

In Chung Tung WPS, the operation records of #1 WTG during November and December in 2006 were chosen to simulate the forecasting model. There are total of 4320 records for November. The power output and wind speed with 10 minutes intervals are shown in Fig. 15. The power will reach nominal power 600 kW when the wind speed is around 12 m/s which is denoted by the red, dashed line. It is clear that the power curve will be easy to recognize when pairs of the power and the wind speed are plotted in Fig. 16. We can also see that there are some unexpected data when the wind speed is higher than cut-in speed, but the power is still low or even zero. These are caused by the protective shutdown when the wind speed of gust exceeds the cutout limit.

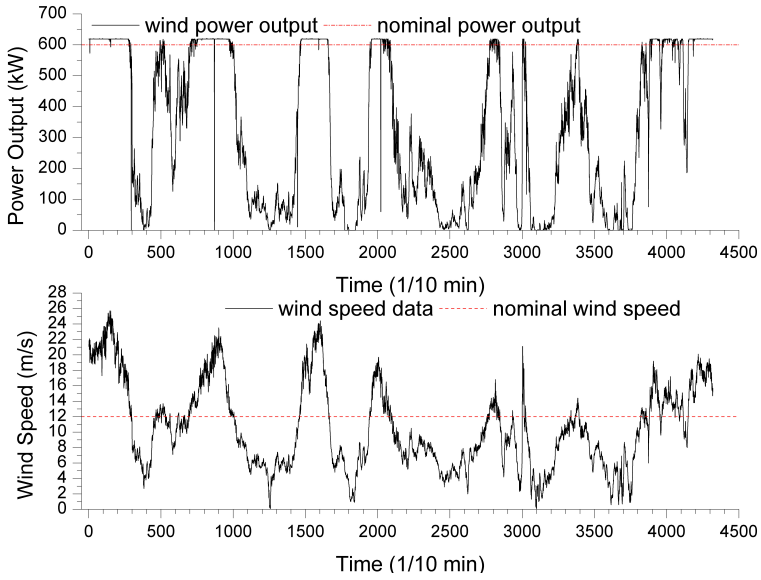

Fig. 15 Chung Tung WPS wind speed and power output in November 2006

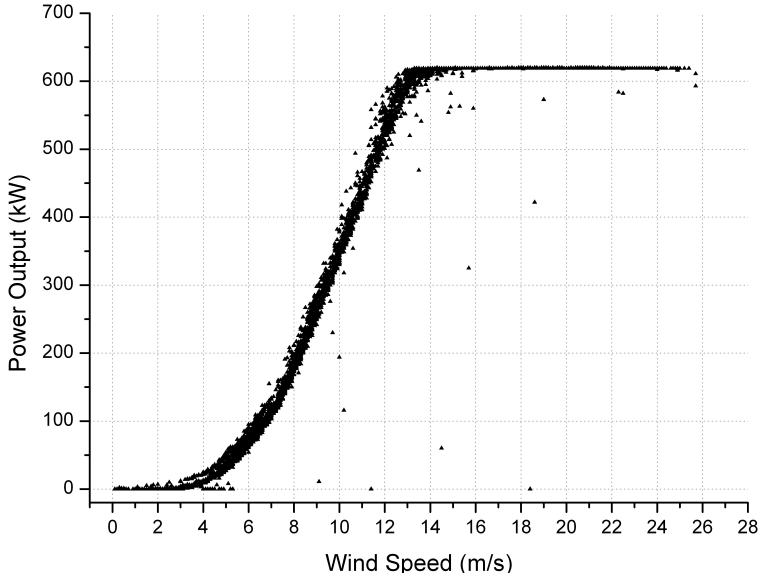

Fig. 16 Scatter plot of power output vs. wind speed in November 2006

(A) Training result:

For extracting useful features of wind power model, some criteria are set to filter the data (see p. 21). The data are normalized by nominal wind speed (12 m/s) and nominal power output (600 kW). After that, dataset is fitted by sigmoidal curve of the Boltzmann function (Fig. 17). The coefficients for the Boltzmann function are A_1=0.0046, A_2=1.04911, x_0=0.79169, and Δx=0.13095. There will be 102 data

points in the set for wind speed from 0 to 25 and 25 to 0 in the lag of 0.5 as scattered.

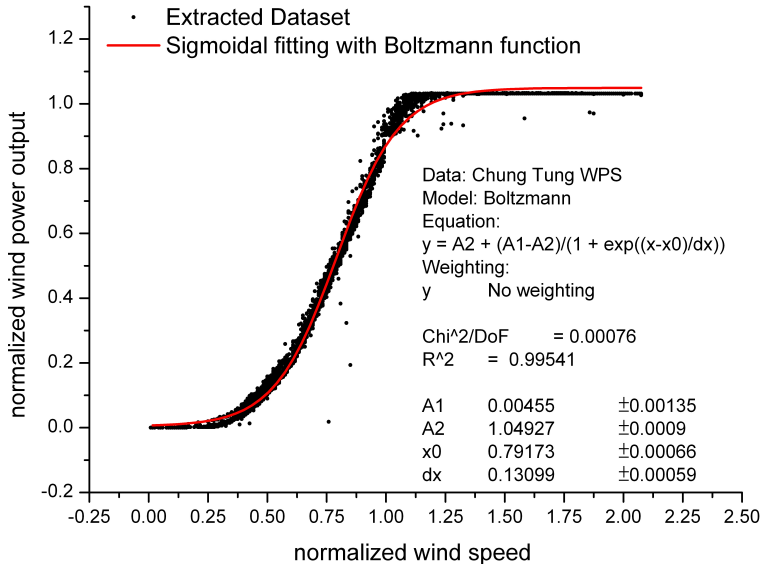

Fig. 17 Scatter points of normalized data and the sigmoidal fitting curve

After we preprocessed the training dataset for the GANN, the input data are stored in desired format to wait feeding into the model. After 100 epochs with 150 offspring to search the optimum, the final objective fitness of GA was 0.35765. By plotting the objective fitness values of each epoch (Fig. 18), we can see the good optimization ability of the GA.

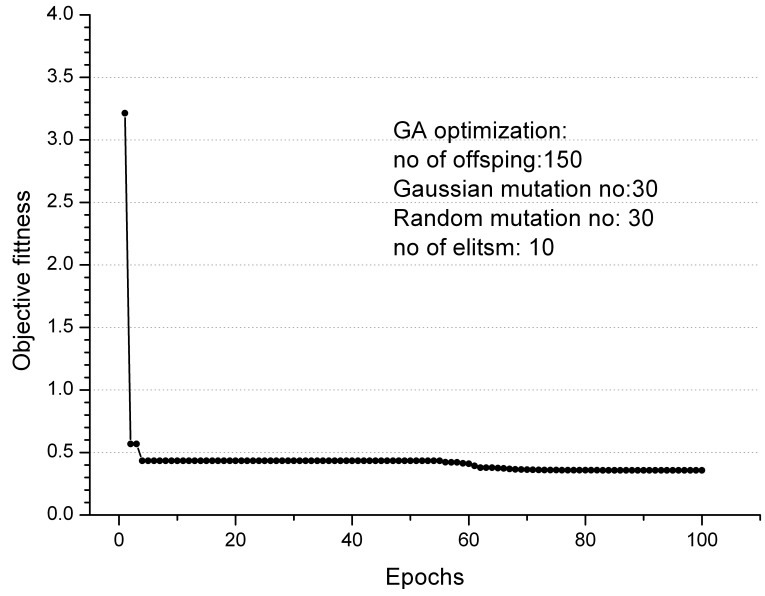

Fig. 18 Optimization process of GA

The output calculated using the optimum solution of the ANN architecture is shown in Fig. 19 with the training dataset. The power output of the trained GANN fits with the training data quite well. The deviation of each point is shown in Fig. 20 and the mean deviation is ±40.58 kW (6.7%) and these are plotted as the red, dashed lines. Note that some deviations become huge when the wind power drops to zero.

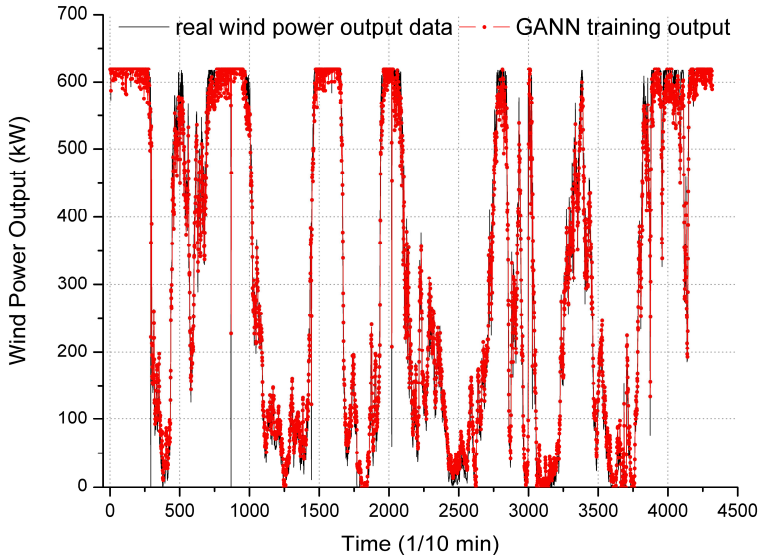

Fig. 19 Power output and training result of Chung Tung WPS

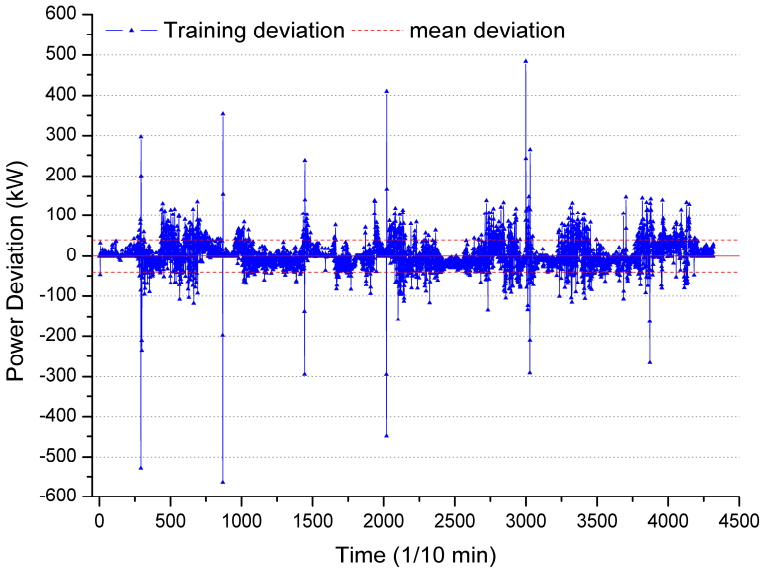

Fig. 20 Training deviation of Chung Tung WPS

(B) Forecasting result:

To demonstrate the forecasting of our model, there are 2736 data available in the operation records for December 2006 (Fig. 21). The dataset shows that there were not any record with wind speed equaling zero, but there are still some zero power output values. These values are because of the sudden shutdown during the wind gust that exceeded the cutoff limit. Also, there are few points located below the wind speed 4 m/s which is the cutin wind speed of the turbine (Fig. 22). Some sudden drops in power output are apparent, and there are only three periods with changing power output. Generally speaking, the power energy is abundant and stable in December in this site.

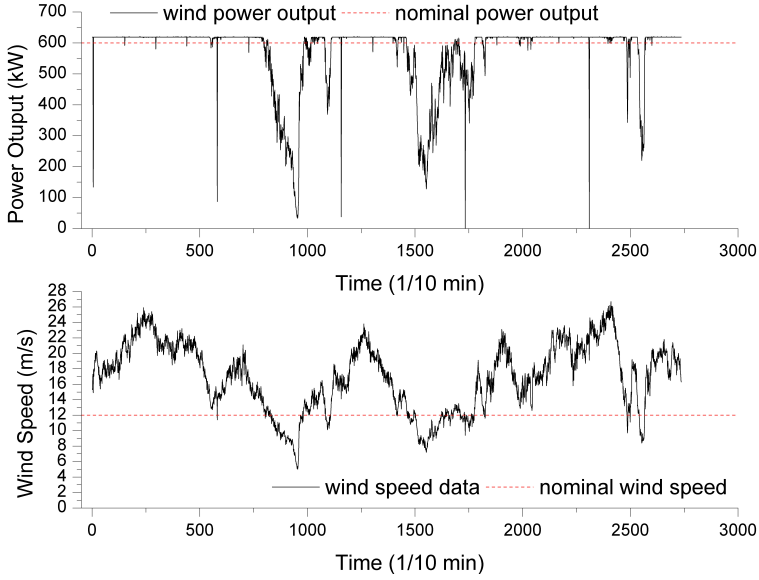

Fig. 21 Chung Tung WPS wind speed and power output in December 2006

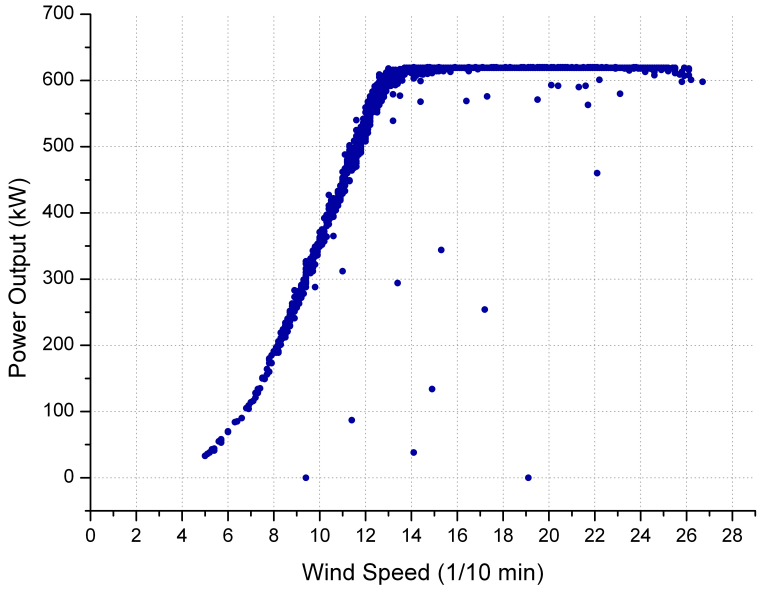

Fig. 22 Scatter points of power output vs. wind speed in December 2006

When the collected test dataset is fed into the trained GANN for forecasting, we can see that the forecasting power output follows the curvature with good extent (Fig. 23). The deviation of each point shows that the forecasting error is around ±39.12 kW (6.52%) except when the gust exceeds the cutout limit (Fig. 24). We can see that deviation is located between the acceptable range if there is rare power drop and sudden drop of wind speed. In normal sense, this is reasonable for GANN to forecast the power output with respect to the wind speed. If the gust can be a measured as an input of GANN, it will enforce the forecasting accuracy of this model.

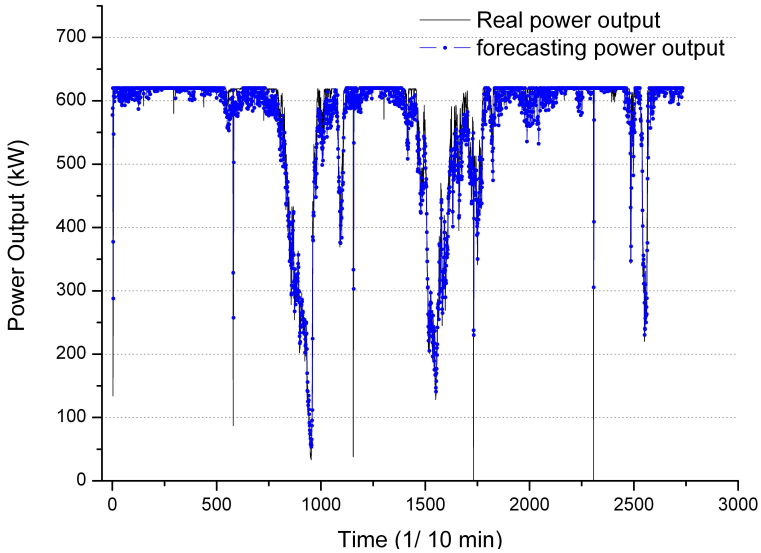

Fig. 23 Forecasting power output and the actual power output of Chung Tung WPS

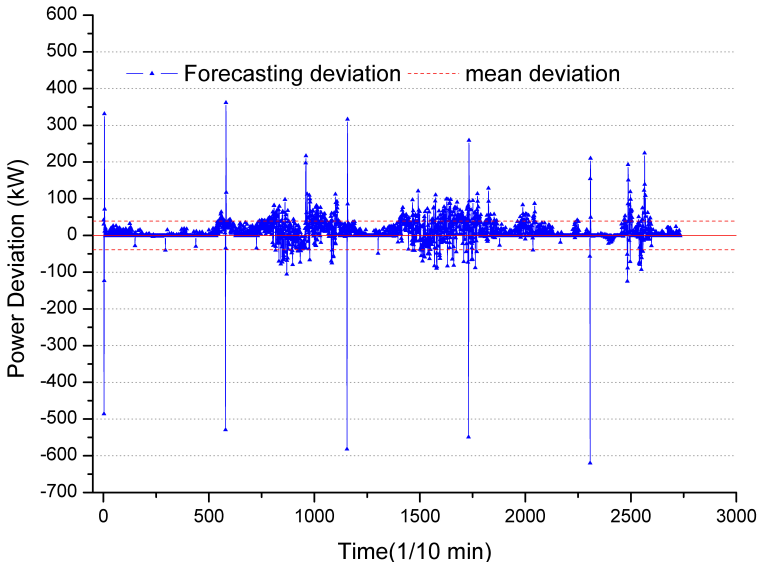

Fig. 24 Forecasting deviation of Chung Tung WPS

5.2 Heng Chun Wind Power station

There are some operation records of these three WTG between February 7th 2007 and March 14th 2007 and the operation record of #1 WTG was chosen for analysis. The duration of data is only about one month and the data was separated in half for training and testing. Training dataset (Fig. 25), February 7th – February 25th 2007, will be managed and normalized in data preprocessing which is mentioned in Sec. 5.1.(A). Nevertheless, the operation parameters of the wind turbine changed, as the nominal power output is now 1500 kW. Similarly, the test dataset, from February 25th – March 14th 2007, is input to check the forecasting ability of trained GANN.

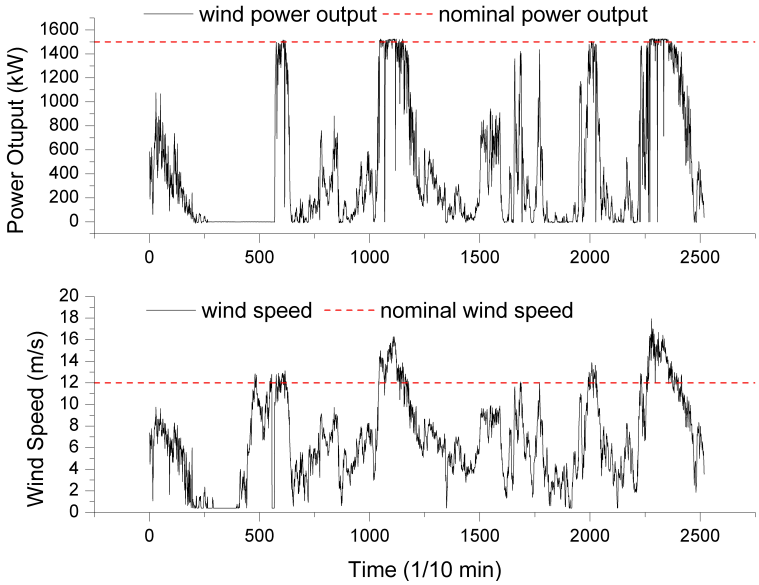

Fig. 25 Training wind speed and power output of Heng Chun WPS

In the scatter plot (Fig. 26), we can find that the pairs of wind speed and power output form the power curve of this wind turbine. Most of the wind speed values range from 0 to 19 m/s and wind power output grows with the wind speed even though it sometimes drops to zero because of the gust.

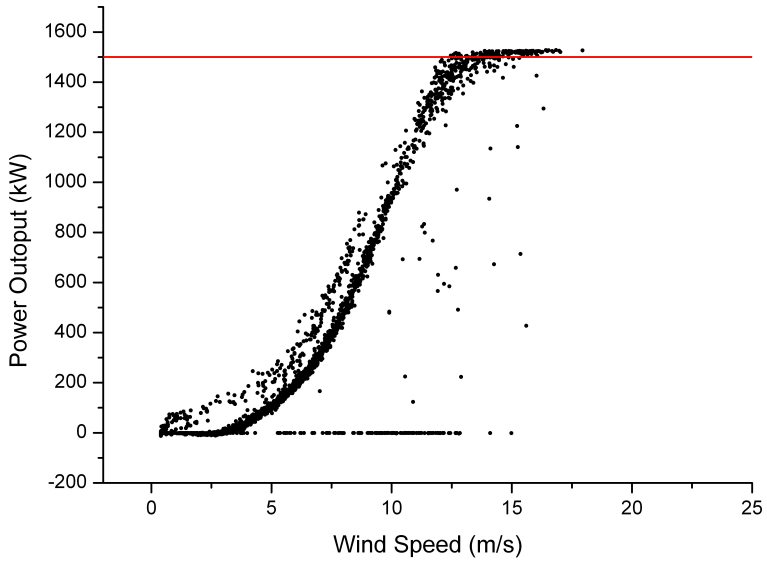

Fig. 26 Scatter points of power output vs. wind speed of Heng Chun WPS

(A) Training result:

By following the selection criteria, the operation data mentioned above is filtered and normalized (Fig. 27). Then the coefficients of sigmoidal fitting in Boltzmann function are A_1=-0.00611, A_2=1.06264, x_0=0.77149, and Δx=0.14388. The

training data are produced for wind speed from 0 to 25 and 25 to 0 in the lag of 0.5 on the fitting curve. There are totally 120 selected data to train GANN.

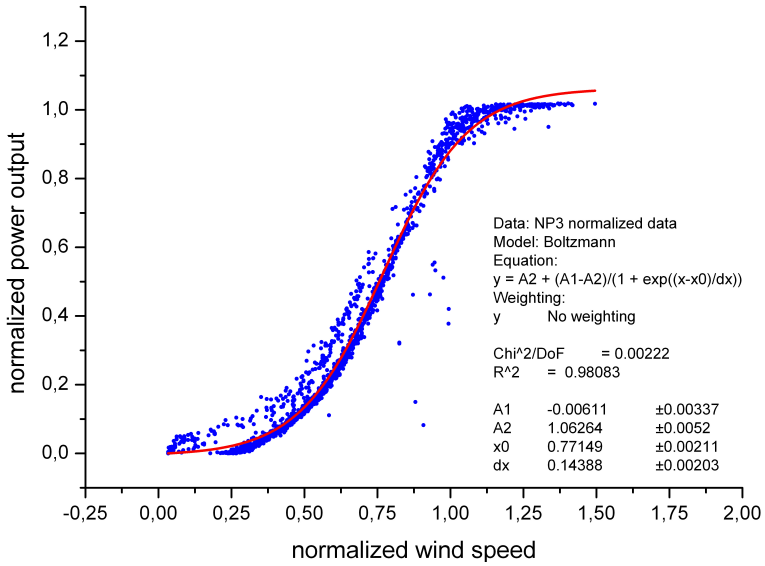

Fig. 27 Scatter points of normalized data and the sigmoidal fitting curve

The GANN model used is an ANN composed of five input neurons, twenty hidden neurons and one output neuron. The transfer function in hidden neurons is the simoidal function and the linear function is adopted in output neuron. The number of total parameters in ANN which includes weights and threshold is 141. The optimum of these parameters to minimize the error between output and target is found by genetic algorithm.

The optimization process of GANN is denoted in Fig. 28 with final objective fitness equaling 0.43. It is shown that the objective fitness quickly reaches 0.5 within 10 epochs and gradually optimizes to 0.43 at 100^{th} epoch.

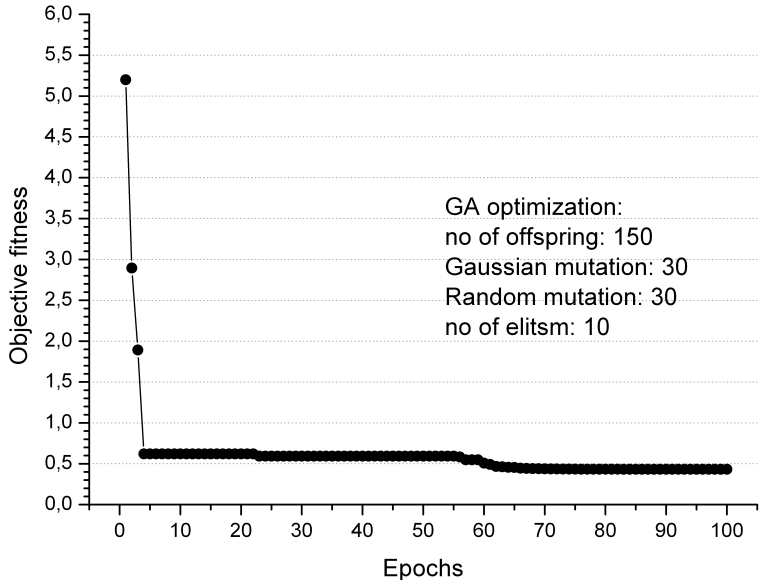

Fig. 28 Optimization process of GA

The training result is shown in Fig. 29 where the black continuous line is the real wind power output and red continuous line with dots is the training power output by our model. It is obvious that the forecasting output of GANN indeed follows the curvature of real power output in training data and error reaches maximum when sudden gusts sometimes come. However, the average error in training dataset is ±190.5 kW which is 12.7% of the nominal power output. The distribution of training deviation error is drawn in Fig. 30.

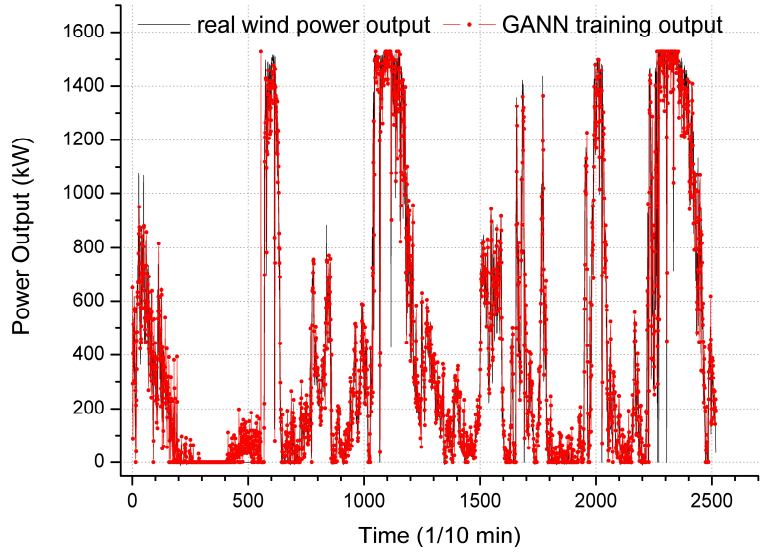

Fig. 29 Power output and training result of Heng Chun WPS

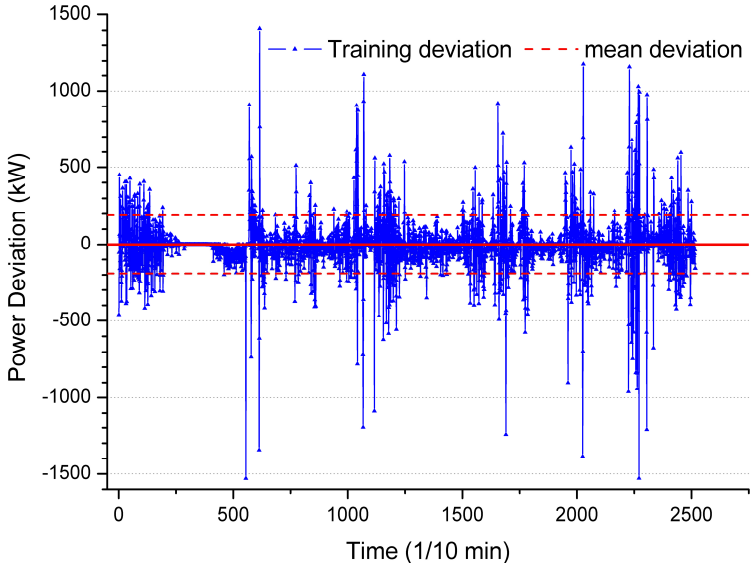

Fig. 30 Training power deviation of Heng Chun WPS

(B) Forecasting result:

To verify and check the forecasting ability of GANN, the testing dataset, February 25th – March 14th 2007, is collected as input. The wind speed and power output distribution with respect to time is shown in Fig. 31. We found it very clear that there were many power shutdowns in the duration. These points can be found in Fig. 32 where the wind speed is over nominal wind speed but the power output is 0.

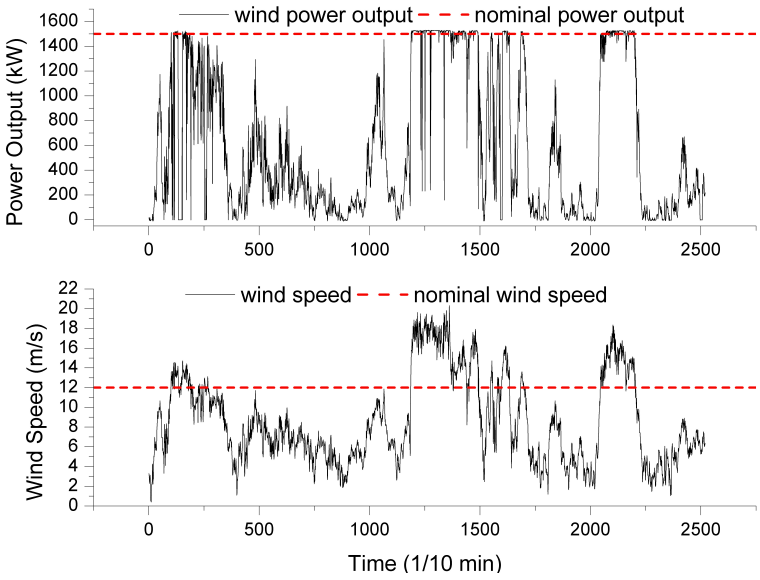

Fig. 31 Testing wind speed and power of Heng Chun WPS

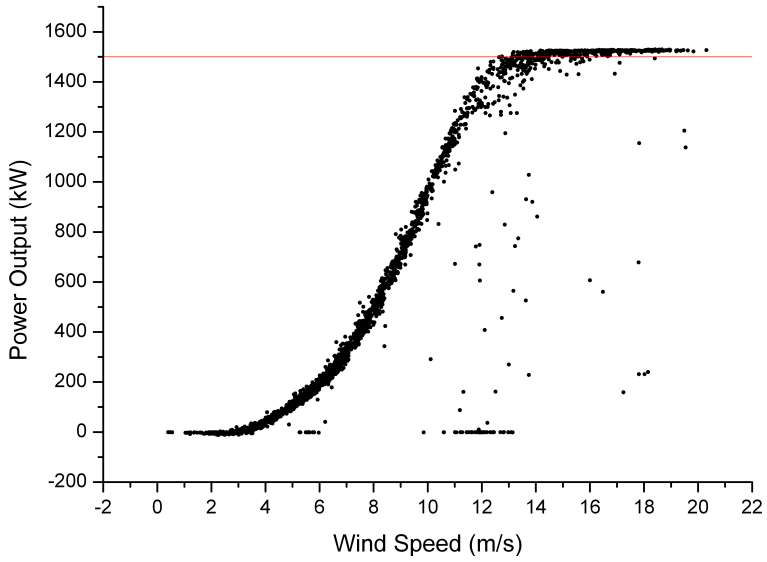

Fig. 32 Scatter points of power output vs. wind speed

The dataset is then inputted into the trained GANN. The result shows that output of GANN can properly forecast the power with changing wind speed (Fig. 33) even in a somewhat fluctuating state. Nevertheless, the average of the forecasting error is ±206.9 kW which is 13.7% of the nominal power output of GE 1.5se. The power deviation with time is shown in Fig. 34.

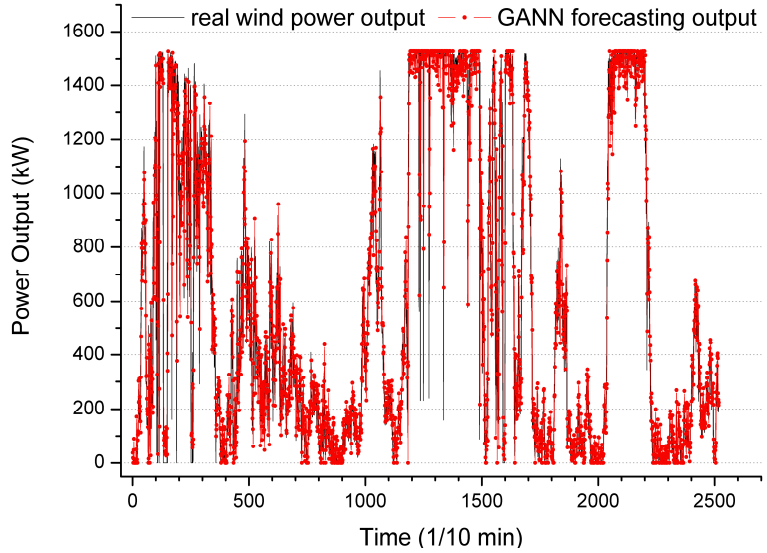

Fig. 33 Forecasting power output and the actual power output of Heng Chun WPS

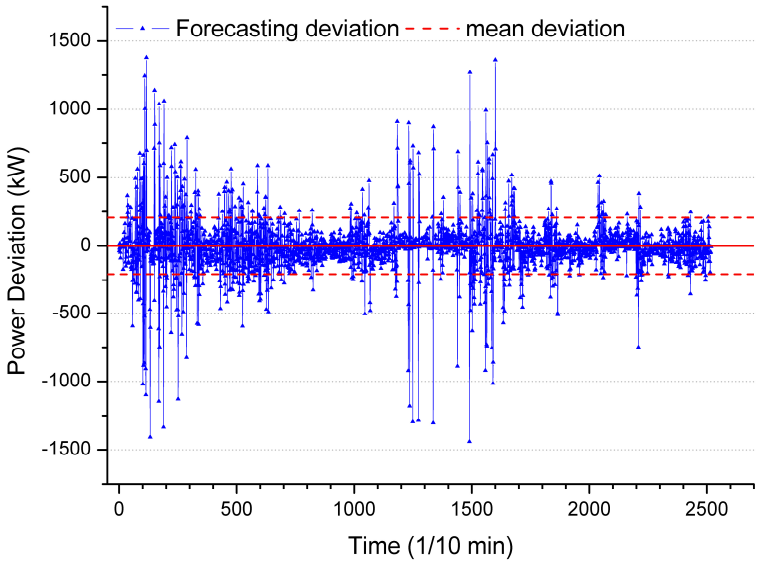

Fig. 34 Forecasting deviation of Heng Chun WPS

6. Discussion and Conclusion

The wind power output forecasting was implemented with artificial neural network and genetic algorithm in this work. Extracting characteristic dataset and sigmoidal fitting by the Boltzmann function were applied in training data processing. There are operation records from two commercial wind power stations in Taiwan used in the simulation. The forecasting results from these simulations are reported.

6.1 Discussion

In this work, we combined ANN with GA to map the wind speed and wind power output in order to forecast the future power production. By referring to the proposed method of using curve fitting, there are something different implemented here. One is that we extract useful data from operational record in order to properly find the mapping behavior. The other is fitting the normalized data with a different fitting method, *i.e.*, simoidal fitting of the Boltzmann function. These help the forecasting in efficiency and accuracy.

In the modeling process, GANN's mapping behavior of the Boltzmann fitting curve will save more time than using the whole dataset. Even the extracted dataset from the short-term operation record will usually have over hundreds pairs of data. It will cost a lot of time for GA in original dataset to search optimum if there is some governing rule between each.

This work uses short-term operation records of commercial wind turbines in Taiwan to forecast the future power output. The results show that this model can handle well the variation behavior between wind speed and power output. However, the result of the error and percentage error denoted that the sudden gust seriously affects the accuracy of forecasting which is not good in open electricity market. In both cases of modeling, it was found that some errors showed huge no matter what in training or forecasting case. After checking the power output and wind speed, we found that the power is near zero in shutdown when average wind speed per 10 min is still high. By consulting the site engineers, we found that such shutdown states are caused by the sudden gust. This could not be avoided in forecasting accuracy because it is the design regulation to protect wind turbines in

secure operation. We think the accuracy of forecasting can rise if we consider the gust as an effective input.

Forecasting time scale can be changed by different duration of training data. Although the forecasting is done using 10 minutes averaged data which is a short-tem case, such process in this work can be applied in longer duration such as 1 hour, 24 hours or monthly period. When the training model is done, the model can forecast the future power output well to some extent in open electricity market. By this forecasting mechanism, forecasting strategy of renewable energy such as wind energy and solar energy can be applied in the same way.

6.2 Conclusion

The forecasting accuracy of GANN is well while the power output is stable during steady wind speeds. The implemented results also show that GANN can forecast wind power output in the future and it maps the behavior of the power curvature in a good state. However, the forecasting error becomes large when the gust comes out during operation that is not mentioned in other researches before.

This work constructs the forecasting model and is applied in commercial wind turbine. This work gives a standard process to forecasting renewable energy. With some crucial parameters, the model can forecast power output well. Until now, there are some optimization method proposed like ARMAT, PSO (Particle Swarm Optimization), ACO (Ant Colony Optimization), AIS (Artificial Immune System), *etc.*. Many forecasting models can be also implemented to compare the efficiency and accuracy in the future.

7. References

[1] K. Daijin and K. Chulhyun, "Forecasting time series with genetic fuzzy predictor ensemble," *IEEE Transactions on Fuzzy Systems*, vol. 5, pp. 523-535, 1997.

[2] S. Kyung-Bin, B. Young-Sik, H. Dug Hun, and G. Jang, "Short-term load forecasting for the holidays using fuzzy linear regression method," *IEEE Transactions on Power Systems*, vol. 20, pp. 96-101, 2005.

[3] S. Kyung-Bin, H. Seong-Kwan, P. Jung-Wook, K. Dong-Jin, and K. Kyu-Ho, "Hybrid load forecasting method with analysis of temperature sensitivities," *IEEE Transactions on Power Systems*, vol. 21, pp. 869-876, 2006.

[4] J. Guajardo, J. Miranda, and R. Weber, "A hybrid forecasting methodology using feature selection and support vector regression," 2005.

[5] A. Sfetsos and C. Siriopoulos, "Time Series Forecasting of Averaged Data With Efficient Use of Information," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 35, pp. 738-745, 2005.

[6] A. M. Gonzalez, A. M. S. Roque, and J. Garcia-Gonzalez, "Modeling and forecasting electricity prices with input/output hidden Markov models," *IEEE Transactions on Power Systems*, vol. 20, pp. 13-24, 2005.

[7] M. Espinoza, C. Joye, R. Belmans, and B. DeMoor, "Short-Term Load Forecasting, Profile Identification, and Customer Segmentation: A Methodology Based on Periodic Time Series," *IEEE Transactions on Power Systems*, vol. 20, pp. 1622-1630, 2005.

[8] A. Sfetsos and C. Siriopoulos, "Time series forecasting with a hybrid clustering scheme and pattern recognition," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 34, pp. 399-405, 2004.

[9] H. Shyh-Jier and S. Kuang-Rong, "Short-term load forecasting via ARMA model identification including non-Gaussian process considerations," *IEEE Transactions on Power Systems*, vol. 18, pp. 673-679, 2003.

[10] L. Gwo-Ching and T. Ta-Peng, "Application of a fuzzy neural network combined with a chaos genetic algorithm and simulated annealing to short-term load forecasting," *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 330-340, 2006.

[11] N. Amjady and M. Ehsan, "Evaluation of power systems reliability by an artificial neural network," *IEEE Transactions on Power Systems*, vol. 14, pp. 287-292, 1999.

[12] H. S. Hippert, C. E. Pedreira, and R. C. Souza, "Neural networks for short-term load forecasting: a review and evaluation," *IEEE Transactions on Power Systems*, vol. 16, pp. 44-55, 2001.

[13] T. Senjyu, H. Takara, K. Uezato, and T. Funabashi, "One-hour-ahead load forecasting using neural network," *IEEE Transactions on Power Systems*, vol. 17, pp. 113-118, 2002.

[14] H. S. Hippert and C. E. Pedreira, "Estimating temperature profiles for short-term load forecasting: neural networks compared to linear models," *IEE Proceedings Generation, Transmission and Distribution*, vol. 151, pp. 543-547, 2004.

[15] D. Baczynski and M. Parol, "Influence of artificial neural network structure on quality of short-term electric energy consumption forecast," *IEE Proceedings-Generation, Transmission and Distribution*, vol. 151, pp. 241-245, 2004.
[16] T. Senjyu, P. Mandal, K. Uezato, and T. Funabashi, "Next day load curve forecasting using recurrent neural network structure," *IEE Proceedings-Generation, Transmission and Distribution*, vol. 151, pp. 388-394, 2004.
[17] S. K. K. Agbossou, "Nonlinear model identification of wind turbine with a neural network," *IEEE Transactions on energy conversion*, vol. 19, no. 3, pp. 607-612, 2004.
[18] T. Saksornchai, L. Wei-Jen, K. Methaprayoon, J. R. Liao, and R. J. Ross, "Improve the unit commitment scheduling by using the neural-network-based short-term load forecasting," *IEEE Transactions on Industry Applications*, vol. 41, pp. 169-179, 2005.
[19] T. G. Barbounis, J. B. Theocharis, M. C. Alexiadis, and P. S. Dokopoulos, "Long-term wind speed and power forecasting using local recurrent neural network models," *IEEE Transaction on Energy Conversion*, vol. 21, pp. 273-284, 2006.
[20] G. J. Tsekouras, N. D. Hatziargyriou, and E. N. Dialynas, "An optimized adaptive neural network for annual midterm energy forecasting," *IEEE Transactions on Power Systems*, vol. 21, pp. 385-391, 2006.
[21] S. H. Ling, F. H. F. Leung, H. K. Lam, L. Yim-Shu, and P. K. S. Tam, "A novel genetic-algorithm-based neural network for short-term load forecasting," *IEEE Transactions on Industrial Electronics*, vol. 50, pp. 793-799, 2003.
[22] J. Holland, "Adaption in Natural and Artificial Systems," *The University of Michigan Press, Ann Arbor*, 1975.
[23] M.-S. Kang, "Generation Cost Assessment of an Isolated Power System With a Fuzzy Wind Power Generation Model," *IEEE Transaction on energy conversion*, 2006.

Appendices

A. windforecast01.m

```
%This is the wind power output forecasting model by GANN (ANN+GA)
through regression
%analsis. The regression model should be input by manual analyze the
%outputfile "norSc.dat" by sigmoidal fitting.
%Setup parameters:
%1.input file names 2.wind turbine parameters 3.ANN's parameters
%Output files:
%1.'date+time+norSc.dat': normalized and extracted training dataset for
%sigmoidal fitting
%2.'date+time+GA_MSE.dat':the MSE record of GA optimization process.
%3.'date+time+trainResult.dat': the total results of training dataset
%4.'date+time+forecastResult.dat': the total results of forecast dataset
%Date:2007/04/15 by Lin Tzu Chao (polo1004@pchome.com.tw)
%=======================================================
clc;clear all;close all;format long;tic
%Open training and testing data
f1=fopen('NU3_1_traindata.dat');%training data
f2=fopen('NU3_1_testdata.dat');%testing data
%Parameters setup
cutin=4;%cut-in wind speed
cutout=25;%cut-out wind speed
nominalV=12;%nominal wind speed
nominalP=1500;%nominal power output
pair=3;
a=2*pair-1;h=20;o=1;%ANN's parameters
%Train--------------------------------
winddata_train=fscanf(f1,'%f',[2 inf]);
winddata_train=winddata_train';
fclose(f1);
[m1 n1]=size(winddata_train);%raw training wind data
maxP=max(winddata_train(:,2));
[M_train]=RegAna(winddata_train,cutin,cutout,nominalV,nominalP,maxP,pair);
[ANNPA]=ANN(M_train,a,h,o);%M_train is extracted and normalized wind data
disp('Training result:')
[H_train]=output(winddata_train,pair,ANNPA,nominalV,nominalP,maxP);
drawplot(H_train,1,m1,maxP);
%--------------
temp1=[H_train.target H_train.out H_train.error H_train.PerErr H_train.AbsAc
H_train.AriAc];
%Forecast-----------------------------
winddata_fc=fscanf(f2,'%f',[2 inf]);
winddata_fc=winddata_fc';
fclose(f2);
[m2 n2]=size(winddata_fc);
disp('Forecast result:')
```

```
[H_fc]=output(winddata_fc,pair,ANNPA,nominalV,nominalP,maxP);
drawplot(H_fc,2,m2,maxP);
%--------------
temp2=[H_fc.target H_fc.out H_fc.error H_fc.PerErr H_fc.AbsAc H_fc.AriAc];
%=====================================================
%
A=fix(clock);
tem1=[A(1:3) '_' A(4:5) '_' 'trainResult01.dat'];
S=sprintf('%d%d%d%c%d%d%c%s',tem1);
save(S,'temp1','-ascii');
tem2=[A(1:3) '_' A(4:5) '_' 'forecastResult01.dat'];
S=sprintf('%d%d%d%c%d%d%c%s',tem2);
save(S,'temp2','-ascii');
%tem3=[A(1:3) '_' A(4:5) '_' 'ANNPar01.mat'];
%S=sprintf('%d%d%d%c%d%d%c%s',tem3);
%save(S,'ANNPA');
%-------------------------------------------------
toc
```

B. RegAna.m

```
function
[M_train]=RegAna(winddata_train,cutin,cutout,nominalV,nominalP,maxP,pair)
%This is the function file for regression analysis about the wind data and it
manages the
% data as training data for GANN.
%Before regression, wind data will be extracted by the function file extractdata.
%The extracted data returned as SC and normalized as norSC. Normalized data
%is outputted as “norSC.dat”. It will be curve fitting by sigmoidal fitting by
%Boltzmann function and then keyin the coefficients in the popup windows.
[M,SC]=extractdata(winddata_train,cutin,cutout,nominalV,nominalP,maxP,pair);
%simoidal fitting -Regression analysis
%normailize extracted data
norSC=[SC(:,1)/nominalV SC(:,2)/nominalP];
A=fix(clock);
temp=[A(1:3) '_' A(4:5) '_' 'norSC.dat'];
S=sprintf('%d%d%d%c%d%d%c%s',temp);
save(S,'norSC','-ascii');
%simoidal fitting
%-----Input results of sigmoidal fitting
prompt = {'Enter A1:','Enter A2:','Enter x0:','Enter dx:'};
dlg_title = 'Please execute sigmoidal fitting and input coefficients';
options.Resize='on';
options.WindowStyle='normal';
options.Interpreter='tex';
num_lines = 1;
def = {'0.0046','1.04911','0.79169','0.13095'};
answer = inputdlg(prompt,dlg_title,num_lines,def,options);
```

```
answer=str2double(answer);
A1=answer(1);
A2=answer(2);
x0=answer(3);
dx=answer(4);
%-----------------------------
x1=(0:0.5:cutout)./nominalV;
x2=(cutout:-0.5:0)./nominalV;
x=[x1 x2]';
y=A2+(A1-A2)*1./(1+exp((x-x0)/dx));
temp=[x y];
%--------------------------------
i=1;[f g]=size(temp);
while i<=(f-pair+1)
   M_train(i,:)=[temp(i:(i+pair-1),1)' temp(i:(i+pair-1),2)'];
   i=i+1;
end
```

C. extractdata.m

```
function
[normalM,SC2]=extractdata(winddata,cutin,cutout,nominalV,nominalP,maxP,pair)
%This is the M file to extract meaningful data for the GANN.
%The input file is composed by hand into CSV.
%This is only suitable for duration of 30 min dataset
[m n]=size(winddata);
%check the size of data
if m<3 %for the dataset of 30 min at least
   disp('error data');
   return;
end
%-----------------------------------------------------
slope=nominalP/(nominalV-cutin);
ix1=find(winddata(:,1)>=0 & winddata(:,1)<cutout & winddata(:,2)>=0 &
winddata(:,2)<=maxP);%include
ix2=find(winddata(:,1)>=nominalV & winddata(:,1)<cutout &
winddata(:,2)<0.9*nominalP);%except
ix3=find(winddata(:,1)>=cutin & winddata(:,1)<nominalV &
winddata(:,2)==0);%except
%-------------------------------------------------------------------------
temp=zeros(m,1);
temp(ix1)=1;
temp(ix2)=0;
temp(ix3)=0;
ix5=find(temp==1);
temp=zeros(m+1,n);
temp(1:m+1,1)=-1;
temp(ix5,:)=winddata(ix5,:);
```

```
SC2=winddata(ix5,:);
%------------------------------
i=1;count=1;
[x y]=size(ix5);
while i<=(x-pair+1)
   if temp(ix5(i):ix5(i+pair-1),1) ~= -1
      %SC is the scatter dataset
      %if temp(ix5(i+pair-1)+1,1)== -1
         SC1( (count-1)*pair+1 : count*pair ,:)=temp(ix5(i):ix5(i+pair-1),:);
      %else
      %    SC( count,:)=temp(ix5(i),:);
      %end
      %Training dataset
      normalM(count,:)=[temp(ix5(i):ix5(i+pair-1),1)'/nominalV
temp(ix5(i):ix5(i+pair-1),2)'/nominalP];
      M1(count,:)=[temp(ix5(i):ix5(i+pair-1),1)' temp(ix5(i):ix5(i+pair-1),2)'];
      count=count+1;
   end
   i=i+1;
end
```

D. ANN.m

```
function [ANNPA]=ANN(M,a,h,o)
%This is main function of ANN.
%The parameters will be optimized by GA which is GA.m.
%The optimized parameters will be recorded in the MAT file.
%-------------------------------------------------------------
disp('GANN starts')
[m n]=size(M);
if (a+o)~=n
   disp('Error setup!');
   return;
end
%ANN training
%--------------------------------
    [ANNPA]=GA(M,a,h,o);
    %calculaue learning MSE
   sum_error=0;
   for i=1:m%--------------------------------
      A=M(i,1:a);
      Target(1:o)=M(i,(a+1):n);
      %**************************
      Hout=logsig(A*ANNPA.wih+ANNPA.bh);
      Out=Hout*ANNPA.who+ANNPA.bo;
      error=Target-Out;
      sum_error=sum_error+sum(error*error');
   end%-----------------------------------------------
```

```
   MSE=sum_error/o/m;
%-----------------------------------
MSE
```

E. GA.m

```
function [ANNPA]=GA(M,a,h,o)
% Genetic Algorithm
%The input data are total training data M, no of input neurons a, hidden neurons h,
%and output neuron o. The output ANNPA is a structural array contains the
parameters
% in ANN.
% The process of GA is explained in the content.
%Selection: Roulette wheel selection
%Recombination: real valued intermediate recombination
%-------------------
disp('GA starts')
var=(a+1)*h+(h+1)*o;%no of variable
k=150;%150; %numeber of population
n1=100;%100;%number of epochs
%no_parent=100; % no of parents
r=2;%mutation range [-1 1] for r=1
k1=4;%mutation precision k1E[4 5 ....20]
rho=1.1;
nmutationG=30;%no of Gaussian mutation
nelit=10;%no of Elitsm
nmutationR=30;%no of Random mutation of Breede Genetic Algorithm
%----------------------
t=1;
nmutation=nmutationR+nmutationG;
%chromosome production
p=1-2*rand(k,var);
%------------------------------
%tic
while t<=n1%optimization criteria
   %Decrease Random mutation
   if t>=50 & rho>=1e-5
      rho=rho/1.3;
   end
   %function evaluation
   for i=1:k
      %y(i)=fun01(p(i,:));%*******
      y(i)=FUN(M,p(i,:),a,h,o);
   end
   %-----------------
   %find nelit minimum
   [s,ix]=sort(y);
   minv(1:nelit)=s(1:nelit);
```

```
minp(1:nelit,:)=p(ix(1:nelit),:);
%-----------------
%Selection: Roulette wheel
%-----------------
%fr=sort(y,'descend');
[c d]=size(y);
ix2=c:-1:1;
fr=y(ix(ix2));
sel_pro=fr./sum(fr);
for i=1:k
    sel=rand;
    sump=0;
    j=1;
   while sump< sel
        sump=sump+sel_pro(j);
        j=j+1;
   end
    parent(i,:)=p(ix(j-1),:);
end
%-----------------
p=zeros(k,var);%reset the mating pool
%-----------------
for i=1:var
    %Recombination (Crossover)
    %real valued recombination: intermediate recombination
    for j=1:ceil((k-nmutation-nelit)/2)
        a2=rand*1.5-0.25;% t is [-0.25 1.25]
        p(2*j-1,i)=a2*parent(2*j-1,i)+(1-a2)*parent(2*j,i);
        p(2*j,i)=a2*parent(2*j,i)+(1-a2)*parent(2*j-1,i);
    end
   %-----------------
    %Elitism
    for j=1:nelit
        p((k-nmutation-j+1),i)=minp(j,i);
    end
   %-----------------
   %Mutation
   %Mutation according to Gaussian probability
   for j=1:nmutationG
        phi=1*(1-2*rand);%phi is [-1 1]
        p(j+k-nmutation,i)=rho*phi*gaussmf(phi, [0.5,0]) + parent(j,i);
   end
   %Mutation according to random
   for j=1:nmutationR
      s=1-2*rand;
      u=rand;
      a1=power(2,-u*k1);
      p(j+k-nmutationR,i)=parent(j,i)+s*r*a1;
```

```
      end
      %-----------------
    end
    %Reinsertion
    %-----------------
     minvalue(t)=minv(1);
     minv(1)
     minpoint=minp(1,:);
     %figure(1)
     %plot(minvalue,'b')
    t=t+1;
end
%toc
%minvalue(t-1)
%minpoint
%---------------
A=fix(clock);
temp=[A(1:3) '_' A(4:5) '_' 'GA_MSE.dat'];
S=sprintf('%d%d%d%c%d%d%c%s',temp);
tem=minvalue';
save(S,'tem','-ascii');
%save('GA_MSE.dat','minvalue','-ascii');
%fid=fopen('GA_OB.dat','w');
%fprintf(fid,'%6.4f \n',minvalue);
%fclose(fid);
%------------------------
ANNPA.wih=zeros(a,h);
ANNPA.who=zeros(h,o);
ANNPA.bh=zeros(1,h);
ANNPA.bo=zeros(1,o);
%-------------------------------------------------------------
for i=1:a
   ANNPA.wih(i,:)=minpoint( (i-1)*h+1 : i*h );
end
ANNPA.bh(1,:)=minpoint(a*h+1 : (a+1)*h );
for i=1:h
    ANNPA.who(i,:)=minpoint( (a+1)*h+(i-1)*o+1 : (a+1)*h+i*o );
end
ANNPA.bo(1,:)=minpoint( (a+1)*h+h*o+1 : (a+1)*h+(h+1)*o );
```

F. FUN.m

```
function sum_error=FUN(M,p,a,h,o)
% The function will evaluate the total square error of ANN for GA.
%M is the training dataset, p is the single set of parameters in ANN, and (a,h,o)
% are the no of neurons in ANN.
      %---------------------------
      Wih=zeros(a,h);
      Who=zeros(h,o);
      Bh=zeros(1,h);
```

```
    Bo=zeros(1,o);
    for i=1:a
       Wih(i,:)=p( (i-1)*h+1 : i*h );
    end
    Bh(1,:)=p(a*h+1 : (a+1)*h );
    for i=1:h
       Who(i,:)=p( (a+1)*h+(i-1)*o+1 : (a+1)*h+i*o );
    end
    Bo(1,:)=p( (a+1)*h+h*o+1 : (a+1)*h+(h+1)*o );
    %calculaue learning MSE
    [m n]=size(M);
    sum_error=0;
    for i=1:m
       A(1:a)=M(i,1:a);
       Target(1:o)=M(i,(a+1):n);
       %**************************
       Hout=logsig(A*Wih+Bh);
       Out=Hout*Who+Bo;
       error=Target-Out;
       sum_error=sum_error+error*error';
       %**************************
    end
```

G. output.m

```
function [H]=output(data,pair,ANNPA,nominalV,nominalP,maxP)
%This is for calculating the result of output of ANN
%data is the raw input data for ANN
%pair is the no of pairs of data to be input
%ANNPA is the structural array of parameters in ANN.
%nominalV, nominal P, and maxP are the operational parameters of wind turbine.
[r s]=size(data);
[a h]=size(ANNPA.wih);
[h o]=size(ANNPA.who);
if r<pair
   disp('Error setup')
   return;
end
%------------------------------
sum=0;sumE=zeros(1,o);sumB=zeros(1,o);
for i=1:(r-pair+1)
   %-----------------
   temp=[data(i:i+pair-1,1)' data(i:i+pair-1,2)' ];
   temp2=[temp(1:pair)/nominalV temp(pair+1:2*pair)/nominalP];
   input=temp2(1:a);
   target=temp(a+1:a+o);
   %------------------
   hout=logsig(input*ANNPA.wih+ANNPA.bh);
```

```
out=nominalP*(hout*ANNPA.who+ANNPA.bo);
%For modifying the power output
ix1=find(out <0);
out(ix1)=0;
ix2=find(out > maxP);
out(ix2)=maxP;
%
error=target-out;
sum=sum+error*error';
%
sumE=sumE+abs( error );
sumB=sumB+target;
%---------------------------------
%Percentage error and Absolute accuracy
%It is for vector outputs
%--------------------------
if o==1
   %PerErr
   if target<=0
      error=0-out;
   else
      error=target-out;
   end
   %------------------
   if error ==0
      PerErr=0;
   elseif target<=0 & error~=0
      PerErr=100;
   else
      PerErr=abs(error)/target*100;
      if PerErr>100
         PerErr=100;
      end
   end
   %AriAc
   if target<=0 & out==0
      AriAc=100;
   elseif target>0 & out~=0
      AriAc=target/out*100;
      if AriAc>100
         AriAc=0;
      end
   else
      AriAc=0;
   end
   %------------------
   AbsAc=100-PerErr;
```

```
    %--------------------------
    elseif o>1
       PerErr=zeros(1,o);
       ix1=find(target==0);
       ix2=find(error==0);
       %-----------------
       [c d]=size(ix1);
       if c >0
          PerErr(ix1)=100;
          if size(ix2) >0
             for j=1:c
                ix3=find(ix2==ix1(j));
                PerErr(ix2(ix3))=0;
             end
          end
       end
       %-----------------
       ix5=find(target~=0);
       PerErr(ix5)=abs(error(ix5)./target(ix5))*100;
       ix6=find(PerErr>100);
       PerErr(ix6)=100;
       AbsAc=100-PerErr;
       %------------------------
       AriAc=zeros(1,o);
       ix7=find(out==0);
       if c>0 & size(ix7)>0
          for j=1:c
             ix8=find(ix7==ix1(j));
             AriAc(ix7(ix8))=100;
          end
       else
          ix10=find(target~=0);
          AriAc(ix10)=out(ix10)./target(ix10)*100;
          ix11=find(AriAc>100);
          AriAc(ix11)=0;
       end
    end
    %--------------------------------
    %PerErr:Percentage Error
    %AbsAc:Absolute Accuracy
    %AriAc:Arithmetic Accuracy
    H.target(i,:)=target;
    H.out(i,:)=out;
    H.error(i,:)=error;
    H.PerErr(i,:)=PerErr;
    H.AbsAc(i,:)=AbsAc;
    H.AriAc(i,:)=AriAc;
end
```

```
MSE=sum/r
Error=sqrt(MSE)
MAPE=sumE/sumB
```

H. drawplot.m

```
function drawplot(H,no,m,maxP)
%The function will plot the results of performance by GANN model.
figure(no)
subplot(3,1,1)
plot(H.target,'--b*')
hold on
plot(H.out,':rs')
grid on
axis([0 m 0 maxP+50])
xlabel('Time (1/10 min)')
ylabel('Power output (kW)')
%-----------------
subplot(3,1,2)
plot(H.error,':rx')
grid on
axis([0 m -maxP maxP])
xlabel('Time (1/10 min)')
ylabel('Error (kW)')
%-----------------
subplot(3,1,3)
plot(H.PerErr,'-ko')
grid on
axis([0 m 0 100])
xlabel('Time (1/10 min)')
ylabel('Percentage Error (%)')
```