ANN MODELING AND OPTIMIZATION OF POWER OUTPUT FROM HORIZONTAL AXIS WIND TURBINE

Thesis

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

RASHMI

(ME13P04)

Under the Guidance of

Dr. SATHYABHAMA A

Associate Professor

and

Dr. SRINIVASA PAI P

Professor, NMAM Institute of Technology, Nitte

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA

SURATHKAL, MANGALURU - 575025, INDIA

March 2019

DECLARATION

by the Ph D Research Scholar

I hereby ***declare*** that the Research Thesis entitled **”ANN modeling and optimization of power output from horizontal axis wind turbine”** which is being submitted to the ***National Institute of Technology Karnataka, Surathkal*** in partial fulfillment of the requirements for the award of the Degree of **Doctor of Philosophy** in **Mechanical Engineering** is a *bona fide report of the research work carried out by me*. The material contained in this thesis has not been submitted to any University or Institution for the award of any degree.

Name of the Research Scholar: **Rashmi**
Register No.: **ME13P04**
Signature of Research Scholar:

Department of Mechanical Engineering

Place: NITK - Surathkal
Date:

CERTIFICATE

This is to *certify* that the Research Thesis entitled **"ANN modeling and optimization of power output from horizontal axis wind turbine"**, submitted by **RASHMI.** (**Register Number: 135051ME13P04)** as the record of the research work carried out by her, is *accepted* as the *Research Thesis submission* in partial fulfillment of the requirements for the award of degree of ***Doctor of Philosophy***.

Dr. Sathyabhama A
Research Guide
Date:

Dr. Srinivasa Pai P
Research Co-Guide
Date:

Dr. Shrikantha S. Rao
Chairman-DRPC
Date:

This thesis is
dedicated to
My Teachers, Family and Friends

ACKNOWLEDGEMENTS

I extend my heartfelt gratitude to **Dr. A. Sathyabhama**, Associate Professor, Department of Mechanical Engineering, N.I.T.K, Surathkal, for being my guide and constant source of encouragement throughout this research work. It gives me great pleasure to express my sincere thanks to **Dr. Srinivasa Pai P**, Professor, Department of Mechanical Engineering, N.M.A.M Institute of Technology, Nitte, for being my co-guide. I thank for his guidance and timely advices which made this study possible.

I extend my gratitude to **Dr. Shrikantha S Rao**, Professor and Head, Department of Mechanical Engineering, N.I.T.K, Surathkal, for his support. My sincere thanks also goes to **Dr. Shashikanth Karinka** Professor and Head, Department of Mechanical Engineering, N.M.A.M Institute of Technology, Nitte, for his suggestions and encouragement. I place on record a special word of gratitude to **Dr. Niranjan N Chiplunkar**, Principal N.M.A.M Institute of Technology, Nitte for encouraging me to do PhD.

Thanks are also due to **Mr. Adarsh Rai**, Assistant Professor, Department of Mechanical Engineering, N.M.A.M Institute of Technology, Nitte, for his help in checking and verifying the codes of my work. I express my gratitude to **Mr. Suhas B G**,**Mr. Srijith B K** and other fellow research scholars at N.I.T.K, Surathkal, and my colleagues of N.M.A.M Institute of Technology, Nitte for their constant support.

I acknowledge with gratitude, the love and patience of my husband **Mr. Prasad Shetty** and son **Chinmay P Shetty**, which gave me strength to complete this research work. I thank all other individuals who have directly or indirectly helped me to complete my research work.

(Rashmi)

ABSTRACT

Integration of wind energy with energy with existing power sources has been restricted due to its intermittent and stochastic nature. Hence, there is a great need to develop an accurate and reliable site-specific prediction model. Forecasting of wind speed which is an important parameter affecting turbine power output, will help the wind energy industry in proper planning, scheduling and controlling. Artificial Neural Network (ANN) has proved its capability in mapping such complex non-linear input-output relations. The main objective of the wind energy industry, is to reduce the cost and increase the power generation by optimizing the controllable parameters affecting the turbine power output. The metaheuristic optimization algorithms, which are robust to dynamic changes are proved to be successful in solving such complex real-world problems.

This research work has been carried out in three different phases namely wind power prediction, wind power optimization and wind speed forecasting (WSF).The data for this research work has been collected from the Supervisory Control and Data Acquisition System (SCADA) of 1.5 MW, pitch regulated, three bladed, horizontal axis wind turbine, located in a large wind farm present in central dry zone of Karnataka, India.

In the present study, different conventional and ANN models have been used to predict the power output of a turbine. ANN models have been developed based on batch learning and Online Sequential Extreme Learning Machine (OSELM) algorithms, by considering carefully selected variables affecting power output, namely wind speed, wind direction, blade pitch angle, density and rotor speed. Maximizing the power output of the wind turbine by optimizing the only controllable parameter namely blade pitch angle has been achieved using three different metaheuristic optimization algorithms. A

hybrid ANN multistep WSF model, which is a combination of OSELM, Cuckoo Search (CS) and Optimized Variational Mode Decomposition (OVMD) method, hence named OVMD-CS-OSELM has been proposed in the present study. The performance of this hybrid model has been then compared with the benchmark models.

From this study it has been found that, the models based on Extreme Learning Machine (ELM) converge extremely faster with better generalization performance and generate a compact network structure compared to Backpropagation learning. Out of the fifteen models based on batch learning, the fully optimized RBF model with ELM learning resulted in good performance with Root Mean Square Error (RMSE) value of 1.73%. The detailed study of OSELM algorithm showed a RMSE value of 1.96%, which is slightly higher than the fully optimized RBF model. However,for the present application due to the online nature of the wind data, OSELM algorithm is highly preferable.

CS optimization algorithm is found to be suitable in optimizing the blade pitch angle of the turbine and accordingly the optimization of the power output, due to its fast convergence and a highest Mean relative PG value of 17.329%.

In comparison with benchmark models, the proposed WSF model showed clear benefits of OSELM over ELM, OVMD over Emperical Mode Decomposition and CS over Partial Autocorrelation function for modeling, data pre-processing and input feature selection, with percentage improvements in Mean Absolute Percentage Error (MAPE) of 3.35%, 48.19% and 12.05% respectively for 1-step ahead forecasting. The proposed model has been validated using a standard database, which is from a meteorological station located in Portugal, thereby establishing its use in WSF.

This research work thus proposes efficient models based on ANN for wind power prediction, optimization and WSF, which is useful in proper planning, integration and scheduling in wind energy industry, thereby making it more competitive and a promising renewable energy source.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

A	Area of the rotor, m^2
$C_1\ and\ C_2$	Acceleration factors
C_p	Power coefficient
fit_i	Fitness value
$gbest$	Global best position
Lb	Lower boundary value
N_{test}	Total number of test patterns
O_k^{μ}	Output of the neural network
P	Power captured by the rotor of a wind turbine, kW
P_{Actual}	Actual power, kW
$P_{Optimized}$	Optimal power, kW
PE_{MAPE}	Percentage improvement in MAPE
$pbest$	Local best position
R	Radius of the rotor, m
s_t^i	Position of the particle at iteration i
Ub	Upper boundary value
u_i^t	Velocity of the particle at iteration i
u_{kj}	Membership value
v	Wind speed, m/s
v_c	Cut in wind speed, m/s
v_r	Rated wind speed, m/s
v_f	Cut-out wind speed, m/s
V_e	Effective wind speed perpendicular to the rotor plane, m/s
V_j^{μ}	Hidden layer output

w_{ji}	Weights between input and hidden layer
w_{kj}	Weights between hidden and output layer
$X^{'}$	Normalized data
X	Actual data
$\tilde{X}$	Predicted data
X_{max}	Maximum value in the data set
X_{min}	Minimum value in the data set

Greek Letters

α	Momentum coefficient
β	Blade pitch angle, deg
η	Learning rate
λ	Tip speed ratio
Ω_r	Rotor speed, radians/s
ρ	Air density, kg/m^3
σ	Width of the RBF unit
τ	Update parameter

Abbreviations

ABC	Artificial Bee Colony
AI	Artificial Intelligence
ANN	Artificial Neural Network
ANFIS	Adaptive Neuro Fuzzy Inference System
AR	Autoregressive
ARMA	Autoregressive Moving Average
ARIMA	Autoregressive Integrated Moving Average
BMA	Bayesian Model Averaging
BP	Backpropagation
CFCM	Conditional Fuzzy C-means
CRO	Coral Reefs Optimization Algorithm
CS	Cuckoo Search

DE	Differential Evolution
ELM	Extreme Learning Machine
EMD	Empirical Mode Decomposition
FCM	Fuzzy C-means
GA	Genetic Algorithm
IMF	Intrinsic Mode Functions
LSSVM	Least Square SVM
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MLP	Multilayer perceptron
NWP	Numerical Weather Prediction
OSELM	Online Sequential Extreme Learning Machine
OVMD	Optimized Variational Mode Decomposition
PACF	Partial Auto Correlation Function
PCA	Principal Component Analysis
PR	Polynomial Regression
PSO	Particle Swarm Optimization
PSO-FCM	Particle Swarm Optimization based FCM
RBF	Radial Basis Function
REI	Residual Evaluation Index
RMSE	Root Mean Square Error
RSM	Response Surface Methodology
SCADA	Supervisory Control and Data Acquisition
SARIMA	Seasonal Autoregressive Integrated Moving Average
SVM	Support Vector Machine
VMD	Variational Mode Decomposition
WT	Wavelet Transform
WSF	Wind Speed Forecasting

CHAPTER 1

INTRODUCTION

1.1 WIND POWER AND ITS SIGNIFICANCE

Rising demand for energy in recent decades for agriculture, industrial and domestic purposes coupled with increasing pollution, global warming and its ill effects have led to a strong focus on renewable energy. Wind, a clean and sustainable energy, is emerging as most promising and competitive energy source, with minimal negative environmental impact. This fast-growing energy sector is proving it′s reliability with competitive economic and environmental payback times. It is expected to fulfill 12% of world′s power demand by 2020 (Fan *et al.* 2009). In this scenario, India is providing substantial business opportunities and policy support to investors and has emerged as world′s fifth largest wind power market (Khare *et al.* 2013). The wind power capacity of India is growing in a faster pace and is expected to reach 60 GW by 2022, thus solving energy crisis.

Today India is the fourth largest wind energy market in the world, with the total installed capacity of 34.293 GW, mainly spread across south, north and western regions of the country (Council 2016), (Progress 2017). Tamilnadu and Karnataka are the two states in south which has good wind potential compared to others. Karnataka is the fourth largest wind energy market in India with a installed capacity of 4509 kW as of March 2018 (Bagul *et al.* 2018). North central dry zone is found to be having good wind potential and is ideally suited for production of wind power across the ten agroclimatic zones of Karnataka as shown in Figure 1.1 (Ramachandra and Shruthi 2003).

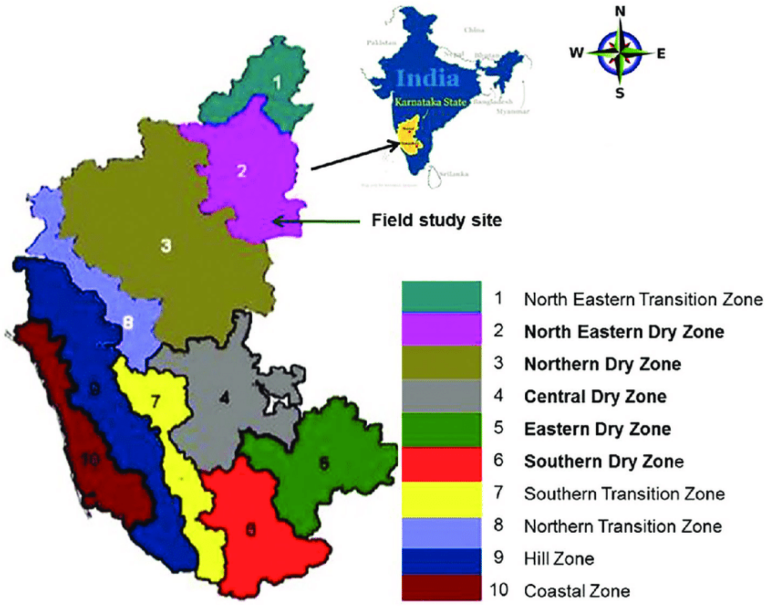


Figure 1.1: Agroclimatic zones of Karnataka

Source: (Soriano *et al.* 2018)

1.2 WIND PREDICTION AND FORECASTING

Wind turbine is the machine that converts the kinetic energy in the wind to electricity. Wind turbine power production depends on the interaction between wind and the rotor. The relative motion of the aerofoil and the surrounding fluid produces a distribution of forces over the blade surface. These forces are resolved into two forces and a moment as lift, drag, and pitching moment as shown in Figure 1.2.

Basically, wind turbines are classified as horizontal axis and vertical axis turbines. Most grid-connected commercial wind turbines today are of horizontal axis type because of their higher aerodynamic yield in comparison to the vertical axis turbines. Most modern wind turbines use three-bladed designs because an even number of blades causes stability problems for a machine. This study hence focuses on a horizontal axis three bladed wind turbine. Parts of the horizontal axis wind turbine is as shown in Figure 1.3. The main parts of the wind turbine are nacelle, rotor and tower. The nacelle consists of the outer covering to protect the machinery, a drive train connected with the

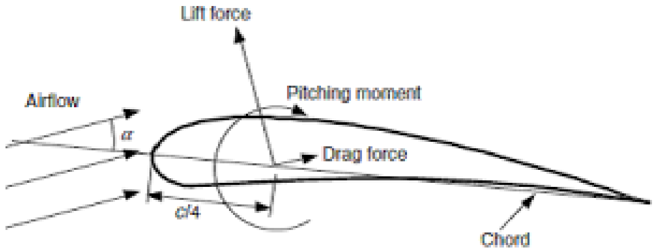


Figure 1.2: Forces and moments on an airfoil section

Source: (Manwell *et al.* 2010)

generator to increase the speed of the shaft and generate electricity respectively, sensors to measure various properties such as wind speed, direction, temperature etc. and yaw mechanism to rotate the turbine and keep it facing towards the prevailing wind. The rotor consists of the blades made up of glass reinforced fiber, extenders, hub which holds all the blades and houses the pitch drive to rotate the blades along its axis and this process is known as pitching of the blade. The tower is a hollow structure made up of steel or concrete, or a lattice structure, mounted on a strong foundation.

Though wind is a clean, abundant and sustainable source of energy, it is uncontrollable and volatile in nature. Wind is subjected to variability in different time scales ranging from very short term (minute by minute), daily and seasonal. The order of variation is different for every case. The randomness, intermittency and stochasticity of the wind and other meteorological variables, directly affect the power output of the wind turbine. Thus creating difficulty in operation, management, maintenance and scheduling of the power systems.

This is a major hindrance in growth of wind energy, as it reduces power penetration and integration. The conventional power generation on the other hand has the flexibility in proper scheduling and controlling. Hence, to tackle these problems and develop wind as a competitive and reliable energy source with large scale integration with the power grid, an accurate prediction and forecasting is crucial. The wind power modeling and prediction helps in estimating the power output of the wind turbine with respect to the

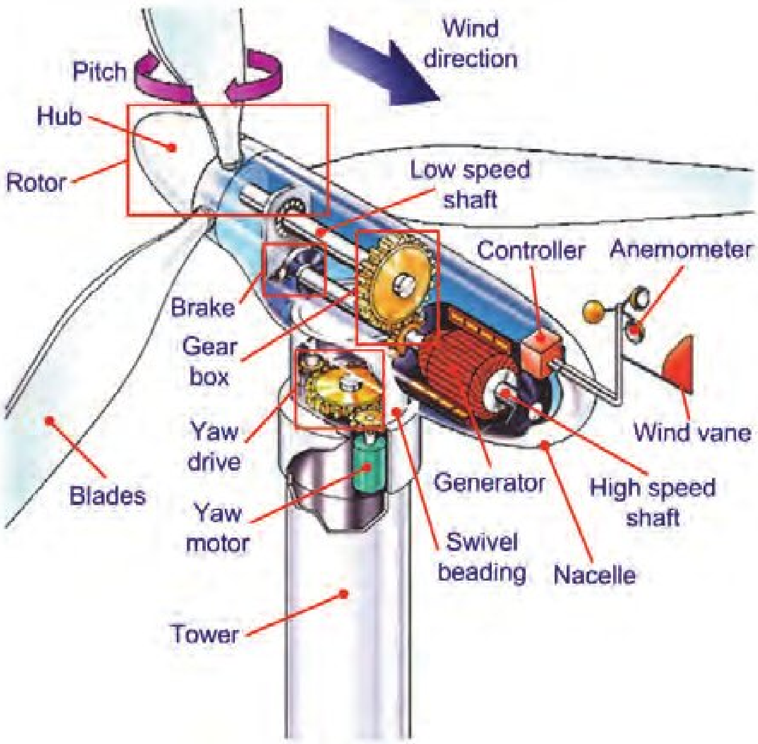


Figure 1.3: Parts of a horizontal axis wind turbine

Source: (Molina and Alvarez 2011)

given input conditions. This helps in monitoring the power production performance of the wind turbine. While forecasting helps in estimating the probable power that can be produced in future time steps. This helps in proper planning and scheduling.

1.3 SIGNIFICANCE OF MODELING TECHNIQUES USED IN POWER PREDICTION/FORECASTING

An accurate and reliable prediction, forecasting model can help overcome the barriers and assist in taking better decisions, leading to better performance monitoring and grid planning. This certainly has significant impact on reducing the loss and fetch higher economic benefits (Zhang *et al.* 2014).

Theoretical energy captured by a wind turbine is given in Equation 1.1

$$P = \frac{1}{2}\rho\pi R^2 C_p(\lambda, \beta)v^3 \tag{1.1}$$

where, P is the power captured by the wind turbine in kW

ρ is the air density in kg/m^3

R is the radius of the rotor in m

C_p is the power coefficient, β is the blade pitch angle in degree, λ is the tip speed ratio, v is the wind speed in m/s,

The tip speed ratio λ is defined as the ratio of the tangential velocity of the blade tip and the effective wind speed. λ can be determined from Equation 1.2.

$$\lambda = \frac{R\Omega_r}{V_e} \tag{1.2}$$

where, R is the rotor radius in meters,

Ω_r is the rotor speed in radians/s

V_e is the effective wind speed perpendicular to the rotor plane in m/s.

The wind is assumed to be blowing in a perpendicular direction to the rotor and hence is not considered in the power equation, but in practice it is not true. Thus, the parameters affecting power generation in the wind turbine are wind speed, air density, blade pitch angle, rotor speed and wind direction.

Researchers have applied a variety of approaches in developing models for predicting the power output of wind turbines. These models can be broadly classified as models based on fundamental equations of power available in wind and models based on the concept of power curve of wind turbine (Thapar *et al.* 2011). The later can be further classified into parametric and nonparametric techniques (Shokrzadeh *et al.* 2014). The parametric techniques are based on mathematical models such as the Linearized Segmented model, the Polynomial power curve, the Maximum Principle method, the Dynamic power curve, the Probabilistic model, the Ideal power curve, the 4-parameter Logistic Function, and the 5-parameter Logistic Function etc.

Nonparametric techniques, unlike parametric techniques, do not impose any prespecified model. The estimation of the power curve in this case is as close as possible to the available data subject to the smoothness of the fit. Nonparametric models include the Cupola Power Curve, the Cubic Spline Interpolation, the Neural Networks, the Fuzzy methods, the Response Surface Methodology (RSM) and the data mining algorithms (Random Forest, k-nearest neighbor), etc (Shokrzadeh *et al.* 2014), (Lydia *et al.* 2014), (Marvuglia and Messineo 2012), (Ouyang *et al.* 2017).

Equation 1.1 can be used to calculate the wind power by considering C_p as 0.593 which is the theoretical maximum value. The chances of error in using Equation 1.1 to predict the power is twofold due to difficulty in achieving theoretical maximum value of C_p and assumption of 100% mechanical efficiency of the turbine.

The power curve on the other hand is derived usually under standard conditions, hence does not depict the actual working condition of the turbine (Commission *et al.* 1998). Hence, though wind speed is the important variable affecting the power output of the wind turbine, the effect of other variables discussed above cannot be neglected. Hence a strong need for developing a site-specific model to predict the power of a wind turbine arise by considering all the parameters affecting its performance. Artificial Neural Network (ANN) is found to be most suitable and a widely used technique in developing such a model.

As it can be observed from Equation 1.1, wind speed is the main parameter affecting the power output of a turbine. Hence forecasting of the wind speed can significantly contribute in estimating wind power. Developing an accurate wind speed forecasting (WSF) model is a challenge because it is affected by many factors such as rotation of earth, local terrain, pressure and temperature gradient etc. (Sfetsos 2002).

The WSF models can be categorized into i) Physical ii) Statistical iii) Artificial intelligence and iv) Hybrid models. Physical methods use physical laws, meteorological and physical data such as temperature, humidity, pressure, roughness, terrain etc. to predict the local wind speed. The availability of data and computation involved in processing the huge amount of data are the limitations of this approach (Zhang *et al.* 2017). The statistical models use the historical data and develop relationships between the power generated and the explanatory variables on training. Hence are called as data driven models. The time series models namely, Autoregressive (AR), Moving Average (MA), Autoregressive Moving Average (ARMA) and Autoregressive Integrated Moving Average (ARIMA) models are most widely used statistical models. They assume that the time series data follow the normal distribution (Erdem and Shi 2011), (Schlink and Tetzlaff 1998). Statistical models work well for linear and stationary data. These methods fail to capture the nonlinear and non-stationary variation of wind speed.

Artificial Intelligence (AI) methods such as ANN, Fuzzy Logic and Support Vector Machines find extensive application in wind forecasting due to their ability to tackle the nonlinear problems (Zhang *et al.* 2017), (Mohandes *et al.* 2004). Out of many

AI methods, ANN has been widely used and exploited by the researchers due to their ability to approximate complex relationships.

A number of ANN architectures namely Multilayer perceptron (MLP) neural network (Liu *et al.* 2015*a*),(Chang *et al.* 2017), Radial Basis Function (RBF) neural network (Li and Shi 2010*b*), (Wang *et al.* 2016), Support Vector Machine (Mohandes *et al.* 2004), Elman neural network (Liu *et al.* 2015*c*) and Extreme Learning Machine (ELM) (Zhang *et al.* 2017), (Salcedo-Sanz *et al.* 2014) are used successfully in WSF.

An increased effort in combining two or more techniques to overcome the disadvantages of the single technique is found in the recent literatures. It was observed that, the forecasting accuracy of these models are enhanced by integrating the advantages of two or more models.

Types of hybridization found in the field of WSF can be listed as follows:

i) Combining different modeling techniques (Liu *et al.* 2012*b*),(Cadenas and Rivera 2010). Combining statistical method with ANN, ARIMA-ANN is found in (Cadenas and Rivera 2010), Seasonal ARIMA (SARIMA) - Least Square SVM (LSSVM) are combined in (Guo *et al.* 2011).

ii) Combining data pre-processing techniques with the modeling method (Liu *et al.* 2013),(Liu *et al.* 2012*a*),(Ren *et al.* 2015). Many data pre-processing techniques such as Wavelet Trasform (WT) (Liu *et al.* 2014), Empirical Mode Decomposition (EMD)(Zhang *et al.* 2016*a*) and its variants, Variational Mode Decomposition (VMD)(Wang *et al.* 2017) and its variants are used.

iii) Combining different models based on weights (Li and Shi 2010*a*),(Li *et al.* 2011),(Sánchez 2008). Bayesian Model Averaging (BMA) is found to be popular and widely used.

Some efforts in combining the optimization algorithms are found in the literature for

i) Optimizing the feature selection (Salcedo-Sanz *et al.* 2015),(Zhang *et al.* 2017). Use of Coral Reefs Optimization algorithm (CRO) is used in (Salcedo-Sanz *et al.* 2015), (Salcedo-Sanz *et al.* 2014) for selecting the best meteorological variables as inputs to the wind speed forecasting model.

ii) Optimizing the model parameters (Liu *et al.* 2015*a*),(Niu *et al.* 2018). Many metaheuristic algorithms namely Bat Algorithm (Niu *et al.* 2018), Genetic Algorithm (Liu *et al.* 2015*a*) and Cuckoo Search (CS) Algorithm (Zhao *et al.* 2015) have been found to be used for optimizing the model parameters.

Hence, a proper utilization of optimization algorithms to optimize the various factors discussed above along with data pre-processing in developing a WSF model will certainly help in building an efficient model.

1.4 ARTIFICIAL NEURAL NETWORK

Artificial Neural Network is a computation paradigm inspired by the functioning of human brain. It is made up of interconnection of nonlinear neurons, which can process the information similar to biological neurons. Hence, they have shown tangible advantages over other classical parametric techniques with regard to learning abilities, generalization aspects, robustness and fault tolerance. Thus, it finds extensive use in research fields such as pattern recognition, signal processing, forecasting, robotics etc. (Paliwal and Kumar 2009), (Kalogirou 2001).

ANN generally consists of three layers: input, hidden and output layer which are interconnected as shown in Figure 1.4 The input layer receives the information from external world and presents it to the network. The hidden layer consists of number of neurons having activation function. This layer processes the information obtained from the previous layer. A neural network model can have single or multiple hidden layers depending on the complexity of the problem. The output layer collects the information

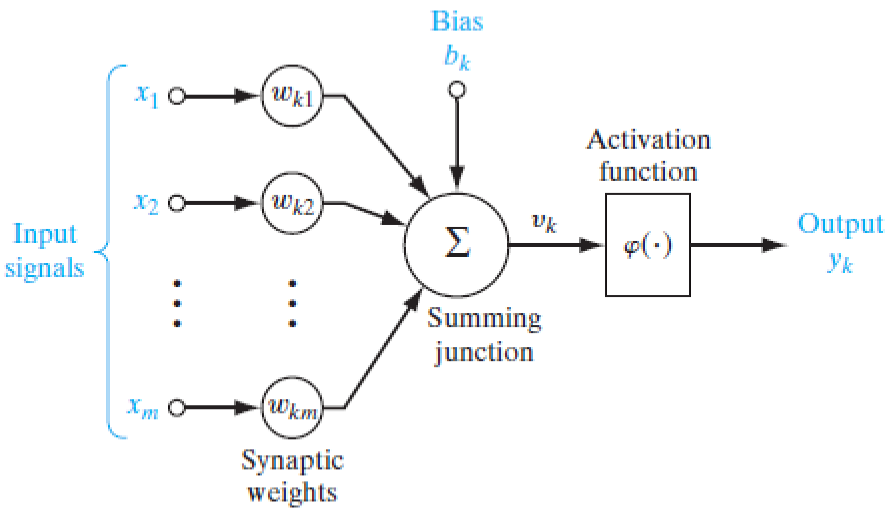


Figure 1.4: General architecture of a neuron

Source: (Haykin and Network 2004)

from the hidden layer and gives the network output. The neurons in each of the layers are interconnected by synaptic weights and biases.

The property of primary significance in ANN, which makes it suitable to be applied in a wide variety of fields is its ability to learn from the environment. The learning in neural network is broadly classified as:

a) Supervised learning b) Unsupervised learning

Figure 1.5 illustrates the working of supervised learning. In this type of learning, input and desired output are presented to the network during training. The initial values of weights and biases are randomly generated. These inputs are used by the ANN model to predict the suitable output. The network output and desired output are compared. The difference between the two, called as error signal, is fed back. The weights and bias are modified under the influence of this error signal.

In the unsupervised learning, the classification rules are developed from the input information presented to the network. These rules are responsible to change the network weights, thereby grouping the input vectors. This results in producing similar network

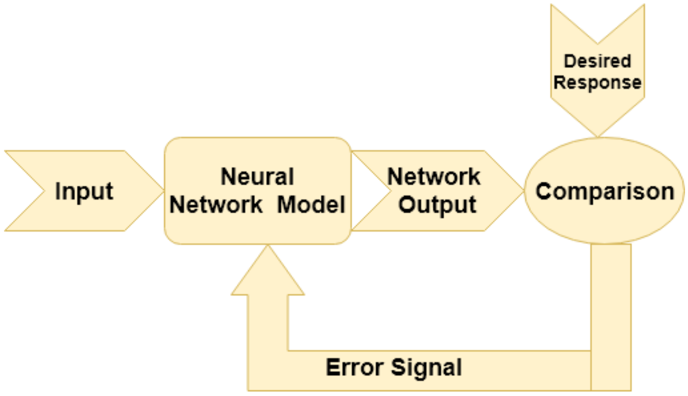


Figure 1.5: Principle of supervised learning

outputs to similar kind of inputs (Haykin and Network 2004).

The feed forward neural networks are the simplest type of neural network in which information flows only in forward direction from input nodes through the subsequent layers without any feedback connections. MLP neural network and RBF neural network are the two important classes of feedforward neural networks.These traditional ANN models use gradient descent learning process, which is generally slow and needs iterative tuning of the network parameters. They also have a tendency to converge to local minima.

Extreme Learning Machine (ELM) algorithm proposed by G.B Haung (Huang *et al.* 2006) is a fast learning algorithm, with better generalization performance, less simulation parameters and avoids the difficulties of local minima problem. Thus, it is most suitable for large scale computing and real time applications (Huang *et al.* 2015). The learning in ELM is a one pass operation which makes it extremely faster, eliminating the iterative procedure associated with gradient descent learning algorithm (Wan *et al.* 2014). Hence finds application in numerous areas of research including wind power prediction (Cancelliere *et al.* 2013), (Zheng-zhong *et al.* 2014). The learning in ELM takes place in one epoch, hence is beneficial in drastic reduction of computational time especially for large data sets such as in wind power prediction application.

From the literature, it can be observed that most of the efforts in wind power modeling and prediction are based on use of batch learning algorithms like BP or ELM by using off-line data. They require the entire dataset to train the model at once and further need a large data set. Such models may not be able to capture the dynamic characteristics of the system. But the data acquired from the wind turbine is online in nature, hence online sequential learning is very much preferred over batch learning, since it avoids training the model, whenever a new set of observation is received. N.Y. Liang et al. introduced Online Sequential ELM (OSELM) with additive or RBF hidden nodes in a unified framework which can learn data 1-by-1 or chunk-by-chunk (a block of data) with fixed or varying chunk size (Liang *et al.* 2006). Hence, OSELM can be used effectively in wind power modeling applications by considering newly arriving data for updating the network parameters online and make it amenable for online learning and avoid efforts in storing large amount of data (Guo *et al.* 2014).

1.5 OPTIMIZATION OF PARAMETERS AFFECTING WIND POWER

Optimization is a suitable tool in renewable energy systems to solve complex design, planning and control problems. Maximizing the power generation at minimum cost is the main objective of the energy industry. High operation and maintenance costs of wind turbine is a barrier in rapid expansion of the wind power market. Lot of efforts by researchers can be found in making the wind energy more competitive over others by maximizing the power production and reducing the cost. The various ways in which this objective is being achieved or attempted to achieve include, optimizing the rotor design and blade geometry (Kenway and Martins 2008), proper site selection (Nigim and Parker 2007), proper positioning of wind turbine and optimizing the layout design of the wind farm (Mittal 2010), (Grady *et al.* 2005), fault detection and condition monitoring (Hameed *et al.* 2009), reliability analysis (Herbert *et al.* 2010) etc. Researchers have used many optimization techniques to achieve these goals.

The objective of maximizing the wind power can also be achieved by optimizing the controllable parameters affecting the power output of the wind turbine. From Section 1.3 it is clear that, the parameters affecting power generation in the wind turbine are wind speed, air density, blade pitch angle, rotor speed and wind direction. Out of these, the only controllable parameter is the blade pitch angle, which can be optimized to maximize the wind power. Defining the constraints in solving an optimization problem is very important. There are practical and theoretical constraints on power production by a wind turbine. The practical constraint being the design capacity of the turbine under consideration. The theoretical constraint is given by the Betz law ie. the theoretical maximum power that a wind turbine can harvest is 59% of the kinetic energy in the wind (Manwell *et al.* 2010).

Traditional optimization algorithms namely linear programming, quadratic programming, Lagrangian relaxation and NelderMead Simplex are simple. In real world, when the problem is non-linear and non-differential in nature, optimum solution may be extremely difficult to be achieved. In such situations the classical optimization algorithms fail to solve the problem. In such situations the metaheuristic algorithms such as Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC) and Cuckoo Search (CS) come into picture. These algorithms are inspired by natural processes and are stochastic in nature. Where the process of searching the solution by trial and error is termed as heuristic, while the algorithms with higher level of heuristics are termed as metaheuristic. They are robust to dynamic changes, have broad applicability and are capable of solving non-linear problems with high complexity. The solution obtained by these methods may not be global optimum, but are sub-optimal in most of the cases (Civicioglu and Besdok 2013).

The metaheuristic algorithms explore the search space by generating diverse solutions. This refers to the global search. Intensification or exploitation is the other major component of these algorithms. This is the process of searching in a regionally focused manner. A good combination of exploration and exploitation leads to the selection of best solution (Koziel and Yang 2011).

1.6 PROBLEM DEFINITION

In wind power modeling studies, there have been limited efforts in proper identification of the input parameters affecting the wind power output and also on comparing different modeling methods, such as based on wind power equation, based on power curve, RSM and ANN. These studies have failed to reveal the significance of selecting the proper input variables in addition to wind speed.

MLP neural network is the most widely used ANN technique in wind power modeling. There have been very few efforts in using RBF neural network in wind power modeling applications. There has been less focus on selecting the location and the number of centers along with selection of width of the RBF units of RBF neural network model development in wind modeling application. Use of meta-heuristic algorithms in selection of the centers and width of the RBF units of the neural network, which reduces the time taken for model development and knowledge required in manual setting of the simulation parameters have been less explored. The Backpropagation (BP) learning algorithm is found to be used most widely and use of ELM which is fast and accurate in comparison to BP is observed to be limited in wind power modeling. The comparison of various ANN models with respect to different activation functions, learning algorithms, center fixing strategies in RBF neural network for wind power modeling application is very rarely found in literature. There has been very limited study on use of online learning techniques, which will reduce the need for storing the huge amount of data in addition to capturing the dynamic nature of the wind data.Wind data is online in nature arriving sequentially in 1-by-1 or chunk-by-chunk manner.

A study on selection of proper approach for developing an accurate objective function as well as performance comparison of different meta-heuristic algorithms with PSO, a most widely used optimization algorithm in wind power optimization, has not been attempted.

Attempts of hybridizing in development of wind forecasting models are observed in terms of combining with data pre-processing techniques, optimizing the network parameters or optimizing the input feature selection. Meta-heuristic algorithms can be successfully used in optimizing the network parameters and input feature selection. There is a need to compare a feature selection method with widely used linear method namely Partial Auto Correlation Function (PACF) to understand its benefits. OSELM, which is a popular online learning algorithm, with its model parameters namely chunk size, number of hidden nodes, weights and biases optimized using a suitable meta heuristic algorithm, can definitely improve the model performance and drastically reduce the effort in setting these parameters.

1.7 OBJECTIVES

- To develop a suitable RBF neural network model by using various training strategies in fixing number and position of centers and width of the hidden neurons to predict the wind power output, compare its performance with the widely used MLP neural network. Compare the performance of these models by using different learning algorithms namely BP and ELM with conventional models.
- To optimize the controllable parameters affecting the wind power output, by using optimization algorithms like, Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC) and Cuckoo Search (CS) and compare their performances.
- To develop an efficient neural network model for wind speed forecasting, thereby supplementing efficient wind power modeling and prediction.

Figure 1.6 presents the overall research methodology followed in this work to fulfill all the research objectives namely modeling, optimization of wind power and forecasting of wind speed.

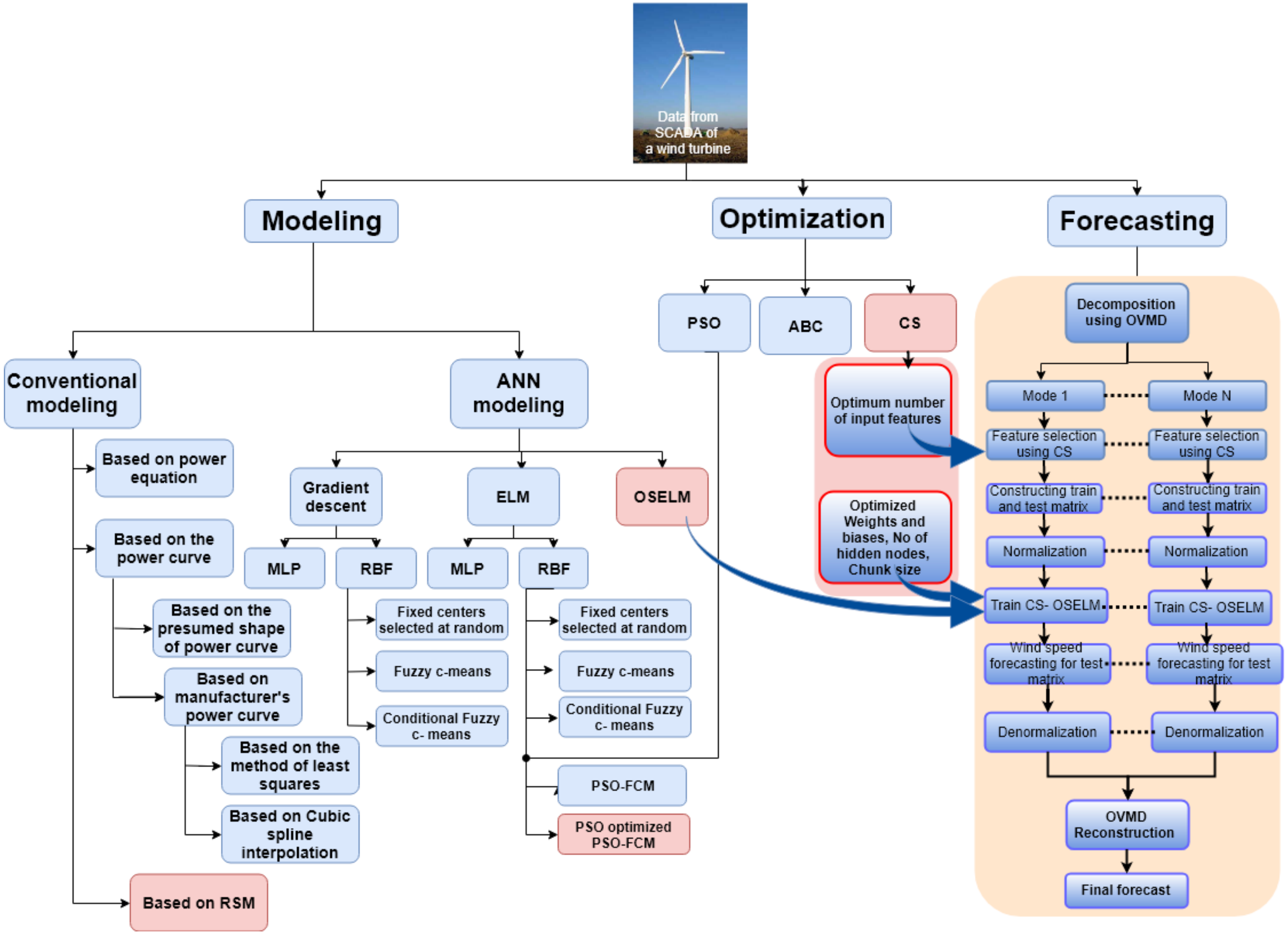


Figure 1.6: Methodology

1.8 ORGANIZATION OF THE CHAPTERS

This thesis consists of seven chapters namely, introduction, literature review, conventional modeling techniques for power prediction, ANN based modeling techniques, optimization of wind power output, ANN based wind speed forecasting and conclusions. The scope for future work, references and appendices are provided at the end.

The second chapter reviews the literature available on wind power modeling, wind power optimization and wind speed /power forecasting. The observations from the literature review is summarized at the end of this chapter.

The third and fourth chapters focus on various conventional and ANN based modeling techniques for wind power prediction, including the details of model development, respective results and comparison of the performance of conventional and ANN models for wind power modeling.

The fifth chapter discusses the details of various optimization techniques and development of suitable objective function for the same and the corresponding results. The sixth chapter gives the details of the proposed model for wind speed forecasting and the benchmark models used for comparison of results. The seventh chapter presents the conclusion from the present study and provides the scope for future work.

CHAPTER 2

LITERATURE REVIEW

The literature review has been presented in three sections. The literature on wind power modeling is provided in the first section, literature on wind power optimization is provided in the second section followed by the literature on wind speed forecasting in the third section. The first section is further divided based on the number of input parameters considered to develop the model. At the end of this chapter, salient observations made from this review is presented to support the problem definition.

2.1 LITERATURE REVIEW ON WIND POWER MODELING

In this section, the literature related to wind power modeling has been provided under two subsections based on the number of input parameters selected. The first subsection covers the modeling studies considering only wind speed as the input parameter as it the most important factor affecting the power output of a wind turbine and studies on modeling the power curve, which gives the relationship between power and wind speed. The literature on wind power modeling considering multiple input variables is provided in the second subsection.

2.1.1 Considering single (wind speed) input variable

Li *et al.* (2001*a*) compared regression and Artificial Neural Network (ANN) models for wind turbine power curve estimation. The regression model was found to be function dependent. It was observed that the ANN model performed better than the regression model under too many influencing factors affecting the wind power generation.

Üstüntaş and Şahin (2008) used least square methodology to model wind turbine power curve and compared it with clustering center fuzzy logic modeling. It was seen that, the wind turbine power curve could be modeled well by the clustering center fuzzy logic with low Root Mean Square Error (RMSE) in comparison to classical least square methodology. The error reduction with decrease in cluster neighborhood distance was observed.

Kusiak *et al.* (2009*b*) integrated evolutionary and data mining for developing the models for predicting wind farm power output by using wind speed as the input. They found that k-nearest neighbor (k-NN) model with Principal Component Analysis (PCA) performed better than other models. The outliers were filtered according to control charts and residual approach.

Kusiak *et al.* (2009*c*) in a similar study, developed two parametric techniques namely, least squares method and maximum likelihood method to model the wind turbine power curve and compared their performance with non-parametric models. Out of the five non-parametric models developed, k-NN model outperformed the others and was selected for comparison with the parametric models. It was proved that the non-parametric models are less computationally intensive and less sensitive to the outliers than the parametric models.

Thapar *et al.* (2011) presented a comparative study of various mathematical methods of modeling wind turbine power curve. It was observed that the use of fundamental equations of wind power fails to replicate the power output of the actual wind turbine. Methods that follow the presumed shape of wind turbine power curve were simple but inaccurate. The curve fitting techniques namely cubic spline interpolation and method of least squares were successful in proper replication of the manufacturers power curve of the wind turbine.

Marvuglia and Messineo (2012) compared three data driven models namely Generalized Mapping Regressor, Multilayer perceptron (MLP) neural network and General Regression neural network to estimate the power curve of the wind farm. To monitor the abnormal situations of wind farm functioning, Residual control chart technique was used. It was observed that the proposed approach performed well on suitable data preprocessing.

Carrillo *et al.* (2013) carried out a review on commonly used equations to model the power curve of wind turbine. Data collected from 200 wind turbines of capacity ranging between 225-7500 kW was used for the analysis. From the study it was found that, exponential, cubic and approximate cubic equations provided best approximation of the power curve. Polynomial equations showed worst results due to their parameter sensitivity.

Lydia *et al.* (2013) compared the performance of parametric and non-parametric techniques for modeling wind turbine power curve. Parametric models are a set of mathematical expressions. The optimization algorithms namely Genetic Algorithm (GA), Differential Evolution (DE), Evolutionary Programming (EP) and Particle Swarm Optimization (PSO) were used to determine the coefficients in parametric models. Non-parametric models namely Fuzzy C-means clustering, neural network and data mining algorithms were used. It was observed that five-parameter logistic function with DE algorithm is the best parametric and neural network is the best non-parametric model respectively.

Shokrzadeh *et al.* (2014) studied parametric and nonparametric methods to model wind turbine power curve. They studied Polynomial Regression (PR), locally weighted PR method, Spline Regression method and Penalized Spline Regression model. The performance of the models were analyzed by using the wind power data from four wind turbines and found that Penalized Spline Regression performed better compared to other techniques.

Goudarzi *et al.* (2014) compared parametric and non-parametric modeling techniques for wind turbine power curve modeling. The data provided by the manufacturers power curve was used to develop the models. GA was used to optimize the coefficients of the mathematical models and the results were compared with that of the ANN model. ANN model showed superior performance over other methods.

Pelletier *et al.* (2016) developed MLP neural network model with two hidden layers, for modeling of wind turbine power curve. Six input parameters namely, nacelle wind speed, turbulence intensity, air density, wind shear, yaw error and wind direction were considered. The results proved the capability of the ANN technique to properly model the power curve in comparison with non-parametric, parametric and discrete methods.

Ouyang *et al.* (2017) proposed data mining and centers of data partitioning approach to model power curve of a wind turbine. Power curve models were built with Support Vector Machine (SVM) algorithm. Best performance of the model was observed with 20 partitions. Results demonstrated that, data partitioning leads to good performance of the model with reasonable computational cost.

2.1.2 Considering multiple input variables

Li *et al.* (2001*b*) developed an ANN model to predict the power of wind turbine by using ten-minute average data of wind direction and wind velocity collected from two meteorological towers. It was found that, ANN model with four inputs performed better than traditional model with single input.

Mabel and Fernandez (2008) developed ANN model to predict wind farm power generation. Three input variables namely, wind speed, relative humidity and generation hours were used. High generation hours and wind speed values above rated were observed to be the important variables contributing to the power generation.

Mabel and Fernandez (2009) in a similar work developed neural network model to estimate the energy yield of the wind farm. They found that 3-5-1 is the best neural network configuration which resulted in mean square error of 7.6 x 10^{-3}.

Tu *et al.* (2010) used MLP neural network to estimate the monthly energy output of a wind farm and concluded that ANN is an efficient technique for this application. On investigating the most suitable training interval for estimating the monthly capacity factor, it was found that half yearly interval provided accurate results in comparison to seasonal and monthly intervals.

Lapira *et al.* (2012) developed three modeling methods namely, Self-Organizing Map (SOM), Gaussian Mixture Model (GMM) and neural network. Only wind speed was considered as an input variable in case of GMM and SOM models. Eight input parameters including wind speed was considered for ANN model and it was found that ANN resulted in better performance.

Liu *et al.* (2012*c*) applied complex-valued recurrent neural network to develop wind power prediction model. Wind speed and wind direction were considered as the input parameters. Probabilistic Neural Network (PNN) was used for filtering the data before using it for training the Recurrent neural network. It was found that considering wind direction as the input parameter along with wind speed improved the performance of the model.

Tu *et al.* (2012) developed neural network model to estimate the monthly energy output of a wind farm. For investigating the adequate training length, they used four training intervals and four training periods. It was found that training length of 12 months and half yearly training period gave the best prediction of monthly capacity factor.

Cancelliere *et al.* (2013) analyzed the performance of two MLP neural network models trained using Backpropagation (BP) and Extreme Learning Machine (ELM) learning algorithms respectively to predict the power produced by wind turbine. The data collected from a wind farm in southern Italy was used after removal of outliers. Three input parameters namely, wind speed, air density and wind direction were considered. It was observed that, the model with ELM learning resulted in lower error also proper data pre-processing reduced the error and instability of the model.

Gautam and Venayagamoorthy (2013) used Elman Recurrent neural network model to predict performance of the wind turbine generator. They used feedback from the hidden layer as the additional input along with wind speed. PSO was used to learn the weights of the Elman neural network. It was concluded that Recurrent neural network can capture the wind turbine characteristics in a better way compared to the traditional approach.

Schlechtingen *et al.* (2013) compared the performance of three data mining models namely, Cluster Center Fuzzy Logic, k-nearest neighbor and neural network model with Adaptive Neuro-Fuzzy-Inference System (ANFIS) to predict wind turbine power output. The study considered three input parameters namely wind speed, wind direction and ambient temperature. It was found that ANFIS model performed best and the error rate decreased on considering two additional inputs along with wind speed.

Petković and Shamshirband (2015) analyzed the factors influencing the power production of a wind turbine. ANFIS was used to select the variables predominantly affecting the wind energy conversion. They found that four parameters affecting the power coefficient of the wind turbine are wind speed, rotor speed, rotor radius and blade pitch angle. Blade pitch angle was the most influential among all.

2.2 LITERATURE REVIEW ON WIND POWER OPTIMIZATION

Kusiak *et al.* (2009*a*) optimized two problems related to wind turbine control namely, maximizing the power output and minimizing the load variability by using evolutionary strategy algorithm. ANN was used to develop the objective function by considering the controllable parameters namely, generator torque, pitch angle and non-controllable parameters namely, wind speed, rotor speed and power output. It was observed that, reduction in generator torque and rotor speed variability was beneficial in increasing the lifetime of the turbine components.

Kongnam and Nuchprayoon (2010) applied PSO algorithm to solve the control problem of a wind turbine, to maximize power output by determining the optimal values of rotor speed and tip-speed ratio. It was found that PSO solved the optimization problem efficiently. Further, the energy yield of variable speed wind turbine is more than fixed speed wind turbine and mean wind speed is the major factor influencing the wind power.

Kusiak *et al.* (2010*a*) in another work proposed a multi-objective wind turbine performance optimization model. Three different objectives, vibration of drive train, vibration of tower and wind power output were used for the evaluation of performance of the wind turbine. Neural network was found to be accurate in predicting the power produced by the wind turbine and vibration. Potential gain in wind turbine performance was observed by optimizing the controllable parameters using evolutionary strategy algorithm.

Kusiak *et al.* (2010*b*) in further studies optimized yaw angle and blade pitch angle for maximization of wind turbine power output by using evolutionary strategy algorithm. Data mining algorithms were used to obtain the functional mapping between controllable, non-controllable variables and the power output of a wind turbine. The study demonstrated improvement in turbine power output by optimizing the two controllable parameters.

Kusiak and Zhang (2011) presented adaptive control scheme for the wind turbine by considering weighted combination of two objective functions, generator torque ramp rate minimization and power maximization. The objective weights were adjusted depending on the predicted power production and demand. The optimization problem was solved by Particle Swarm Fuzzy algorithm. The feasibility of the proposed control scheme was demonstrated through industrial case study.

Kusiak and Zhang (2012) introduced a control scheme for mitigating wind turbine vibration and power generation optimization. Three data-driven models namely tower vibration model, drive train model and power generation model were developed. The first two models represented the turbine vibration and the later, the process of power generation in the turbine. The two controllable parameters namely blade pitch angle and generator torque were optimized using PSO algorithm. It was observed that, though increase in power generation can be achieved with higher generation torque, it leads to higher acceleration of the tower and the drive train. Hence optimization strategy with lower generation torque was recommended.

2.3 LITERATURE REVIEW ON WIND SPEED AND POWER FORECASTING

Salcedo-Sanz *et al.* (2014) proposed a hybrid short term wind speed forecasting (WSF) model based on ELM. The input features were selected by Coral Reefs Optimization algorithm (CRO) from the meteorological parameters forecasted by a physical weather forecast model. It was proved that, features selected using CRO to train ELM model resulted in accurate wind speed forecasting compared to other optimization algorithms and neural network model combinations used in the study.

Liu *et al.* (2015*b*) presented four hybrid models by merging signal decomposing algorithms namely, Wavelet Decomposition (WD), Wavelet Packet Decomposition, Empirical Mode Decomposition (EMD), Fast Ensemble EMD (FEEMD) with ELM. From the study they observed that, ELM is suitable for WSF. Hybrid models with signal decomposition perform better than single ELM, the performance of Wavelet Packet Decomposition is good in 1-step and 2-step forecasting, Fast Ensemble EMD show good prediction accuracy in 3-step forecasting and Wavelet Packet Decomposition performs better than WD and EMD.

Liu *et al.* (2015*a*) combined FEEMD, Mind Evolutionary Algorithm (MEA), MLP neural network and GA to develop two hybrid WSF models each using MEA and GA for optimizing the parameters of MLP neural network respectively. It was found that FEEMD algorithm was important in improving the performance of MLP neural network in comparison to MEA and GA. MEA showed better performance in parameter optimization than GA, as the later suffered from local optimization problems.

Ramasamy *et al.* (2015) developed WSF model based on ANN to predict the mean daily wind speed at 11 locations of Himachal Pradesh, India. The study aimed to identify the potential sites for wind energy harvesting. The inputs considered for ANN model were, air pressure, altitude, temperature and solar radiation. The predicted wind speed was in the range of 1.27 to 3.78 m/s. It was found that micro wind turbine with generation capacity ranging between 773.61 W to 5329.76 W can be installed to fulfill the electricity requirement for lighting applications.

Salcedo-Sanz *et al.* (2015) in a further study introduced hybrid optimization algorithm by combining CRO and Harmony Search (HS), to get the most suitable meteorological parameters as the inputs to short term WSF model based on ELM. It was found that the model with hybrid optimization algorithm was more efficient in feature selection than CRO and HS, when used individually.

Wang *et al.* (2015) proposed a hybrid WSF model by combining ELM, Seasonal Auto Regressive Integrated Moving Average (ARIMA) and the Ljung-Box Q-test (LBQ). Results of the proposed model exhibited good forecasting ability, when compared with other most widely used models such as BP, ELM, ARIMA and Seasonal ARIMA (SARIMA). The hybrid model possessed combined advantages of ELM and SARIMA and hence resulted in high learning speed, tuning free operation and required a small dataset for training.

Li *et al.* (2016) introduced a short-term wind power prediction model based on error correction and ELM. ELM was first used for short-term forecasting of wind power then, on processing the forecasting error based on the persistence approach, the ultra-short-term forecasting was achieved. It was noted that, there was drastic change in the error on varying the number of hidden nodes. Improvement in accuracy of ultra-short-term forecasting was observed on error correction.

Wang *et al.* (2016) introduced a hybrid GA-BP neural network in combination with Ensemble EMD (EEMD) for wind speed forecasting. EEMD being an improved method of EMD, was effective in overcoming the mode mixing problem. GA was used to optimize the weights of BP neural network. From the study it was found that, ANN is sensitive to parameter settings and hence appropriate settings of the parameter has definite impact on the forecasting accuracy.

Zhang *et al.* (2016*a*) proposed a hybrid short-term WSF model by combining EMD and feature selection with ANN or SVM. No significant difference in performance of SVM and ANN was observed in the study. Significant performance improvement was observed on combining EMD with SVM or ANN. Feature selection showed two-fold advantages, significant improvement in performance of the models and reduced computational efforts.

Chang *et al.* (2017) developed an improved RBF neural network model with error feedback for wind speed and power forecasting. An additional shape factor introduced in the Gaussian activation function of the neural network and a method for effective initialization of width and center values were proposed. The forecasting accuracy of the proposed model was compared with four other neural network models including BP, ANFIS and was proved to be better.

Feng *et al.* (2017) developed multi-model, which is data-driven, with an ensemble machine learning method consisting of two layers. The study focused on deep feature selection for appropriate selection of input features using PCA, Partial Autocorrelation Analysis, Autocorrelation Analysis, Granger Causality Test and Recursive Feature Elimination. Different models were used in Two layers and Grid method was used for optimal selection of parameters. It was found that, the blend of different algorithms performed better than the single and the Polynomial-kernel SVM was found to be best.

Both feature selection and hybridizing of the models significantly improved the forecasting accuracy.

Mi *et al.* (2017) proposed a hybrid multistep WSF model based on Wavelet Packet Decomposition (WPD), Wavelet Denoising, EMD, ARIMA, ELM and Outlier Correction method. Denoising and decomposition steps were used for data pre-processing in terms of reducing the noise and non-stationary characteristics of the wind speed data. Outlier correction method was helpful in overcoming the effect of overfitting in ARIMA-ELM model. The proposed model was observed to be accurate in comparison to most widely used WSF models namely, BP, ELM, ARIMA, Elman neural network and WPD-ELM.

Peng *et al.* (2017) presented a hybrid multistep WSF model, which is a combination of On-line Sequential Outlier Robust ELM (OS-ORELM) and Time-varying mixture copula function. To further improve the performance of OS-ORELM, a forgetting mechanism and ordered aggregation techniques were used. Bernaola Galvan Algorithm (BGA) was used as a data pre-processing technique and Adaptive Variational Mode Decomposition (AVMD) was used for signal decomposition. To optimize the model, Modified Crisscross Optimization Algorithm with self-adaptive mutation was used. Superior performance of AVMD over EEMD based models and online models over offline models in multistep forecasting were observed. The proposed model was proved to be fast and efficient.

Wang *et al.* (2017) proposed a hybrid model for multistep ahead WSF based on Variational Mode Decomposition (VMD), Wavelet neural network optimized using GA and Phase Space Reconstruction (PSR). VMD was used to decompose the data into sub-series, PSR was used for input-output selection. The proposed model showed superior performance over persistence model and four other neural network models.

Zhang *et al.* (2017) developed a hybrid model based on ELM for short term forecasting of wind speed. Real-valued Backtracking Search Algorithm (RBSA) and Binary-Valued Backtracking Search Algorithm (BBSA) were used respectively for optimizing the weights and bias of the model and for input feature selection. The data denoising was done using Optimized Variational Mode Decomposition (OVMD). It was observed that the models that used OVMD denoised data resulted in better prediction accuracy compared to the noisy and EMD denoised data. Optimization of weights and features selection in combination with OVMD decomposition resulted in an effective WSF model.

Liu *et al.* (2018) proposed hybrid WSF model based on WD, Sample Entropy, VMD, Modified adaBoost. RT with BroydenFletcherGoldfarbShanno Quasi-Newton Back Propagation (BFGS), named as WD-SampEn-VMD-MadaBoost-BFGS-WF. It was found that the hybrid model performed better than the MAdaBoost-BFGS model, the WD-SampEn-VMD-BFGS model, the BFGS-WF model and the BFGS model. It was observed that, with increase in forecasting steps, the forecasting error accumulates and the mapping becomes complex.

Naik *et al.* (2018) presented a hybrid WSF model based on VMD, multi kernel regularized pseudo inverse neural network and Vaporization and Precipitation-based Water Cycle Optimization Algorithm (VAPWCA) for model parameter optimization. To reduce the mutual effects between different modes, VMD was used. EMD was used in the study to evaluate the superiority of VMD. It was proved that VMD based models performed better than that of EMD and the VAPWCA optimized hybrid model showed good forecasting precision over other hybrid models without optimization.

Niu *et al.* (2018) presented a hybrid multistep WSF model by combining Singular Spectrum Analysis (SSA), a Modified Bat Algorithm and BP neural network. SSA was used to select the meaningful features. To prove the capability of SSA, the performance was compared with widely used techniques such as WD, EMD and ensemble EMD. It was observed that the proposed model gave better forecasting accuracy compared to others.

2.4 OBSERVATIONS FROM THE LITERATURE REVIEW

1. Modeling of wind power based on fundamental equations are complex and inaccurate due to the theoretical assumptions made.
2. The wind turbine power output on site is different from that provided by the manufacturers power curve, due to difference in the operating conditions.
3. ANN performs better than parametric and other non-parametric models for wind speed/ power prediction applications.
4. ELM is proved to be fast and efficient with less tuning of parameters compared to BP for training ANN models.
5. Considering carefully selected input parameters in addition to wind speed, significantly improves the wind power modeling accuracy.
6. The performance of ANN is sensitive to proper choice of input features, setting of model parameters such as weights, biases, width and centers of the RBF unit.
7. Optimization of the controllable parameters of the wind turbine results in significant increase in the power production.
8. Many optimization algorithms have been used by the researchers to meet different objectives such as optimizing the controllable parameters of the wind turbine, optimizing the model parameters and optimizing the feature selection. Proper selection of optimization algorithm plays a role in achieving global optima.
9. Use of data pre-processing methods greatly improves the forecasting accuracy. VMD and its variants are very widely used and have been proved to be better than other methods.
10. Models based on online learning, perform better than that of offline models due to the non-linear, non-stationary characteristics of the wind. However there has been very limited efforts in this direction regarding the use of these models for wind speed or power forecasting.

CHAPTER 3

CONVENTIONAL MODELING TECHNIQUES FOR POWER PREDICTION

This chapter provides the details of data collection, pre-processing of the data and various conventional modeling techniques for wind power prediction. The chapter includes the details of model development followed by results and discussion.

3.1 DATA COLLECTION AND PREPROCESSING

A 10 min resolution data collected from Supervisory Control and Data Acquisition (SCADA) system of 1.5 MW, pitch regulated, three bladed, horizontal axis wind turbine, located in a large wind farm present in central dry zone of Karnataka has been used in this research. The sample data set has been provided in Table 3.1. The rated, cut-in and cut-out wind speeds of the turbine considered are 13, 4 and 20 m/s respectively. The six-month data (June-November 2013) considered for the study has been averaged to 1 hour interval.

The pre-processing of the data is an important step in model development, to enhance the prediction accuracy. Error in the data collected from SCADA may be due to failure of sensors and many other subsystems. Such erroneous data are identified and removed. A total of 2966 data sets has been used. Out of the total data of 2966, 85%, i.e., 2522 data have been used for training and the rest 15% for testing the developed Response Surface Methodology (RSM) and Artificial Neural Network (ANN) models.The data has been normalized between 0 and 1 using the Equation 3.1 to ensure that every input provides equal contribution in ANN model development.

Table 3.1: Sample data set collected from the SCADA of the wind turbine

Local Time	Out door Temp °C	Wind Direction Degree	Wind Speed m/s	Nacelle Temp °C	Blade pitch Angle Degree	Rotor Speed RPM	Active Power kW
8/1/2013 0:10	22	-0.1	11.8	26	4.9	16.3	1347.7
8/1/2013 0:20	22	-0.8	13	26	9.7	16.5	1491.1
8/1/2013 0:30	22	-0.5	12.8	26	7.7	16.4	1448.7
8/1/2013 0:40	22	-0.4	12.4	26.1	7.2	16.4	1482.4
8/1/2013 0:50	22	-0.5	12.5	26	8	16.5	1481.3

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{3.1}$$

3.2 CONVENTIONAL MODELING TECHNIQUES INVESTIGATED

The broad classification of the conventional modeling methods used in this study for wind power prediction are as follows:

- Models based on wind power equation
- Models based on the concept of power curve of wind turbine

  Models based on the concept of power curve that provides the relationship between the turbine power and wind speed are further classified as

  - Models based on a presumed shape of power curve
  - Models based on actual power curves supplied by the manufacturer
    - Method of least squares
    - Cubic spline interpolation

The above methods mainly consider wind speed as the input parameter. If more input parameters have to be considered, the non-parametric methods are useful. Response

Table 3.2: Details of the conventional models developed

Model	Description
Model-1	Model based on wind power equation
Model-2	Model based on presumed shape of power curve
Model-3	Model based on method of least squares
Model-4	Model based on cubic spline interpolation
Model-5	Model based on Response Surface Methodology

Surface Methodology (RSM) is one of such conventional technique considered in this study. The details of the models developed in the present study is provided in Table 3.2.

3.2.1 Model-1

The maximum theoretical power P_e available in the wind is given in Equation 3.2.

$$P_e = C_p(\frac{1}{2}\rho A v^3) \tag{3.2}$$

where ρ is air density in kg/m^3, A is the area of the rotor in m^2, v is the wind speed in m/s. The value of C_p, the power coefficient is considered as 0.593 which is called as Betz limit (Manwell *et al.* 2010).

3.2.2 Models based on the concept of power curve

The power curve of the turbine under study has been shown in Figure 3.1. The values of power corresponding to various wind speeds from manufacturers power curve is listed in Table 3.3. From the power curve it can be observed that, the power production in wind turbine takes place only for the value of wind speed higher than cut-in speed. Steady rise in power takes place till rated wind speed, thereafter it is maintained constant at rated power till the cut-out. The information contained in the power curve is used to develop power prediction models.

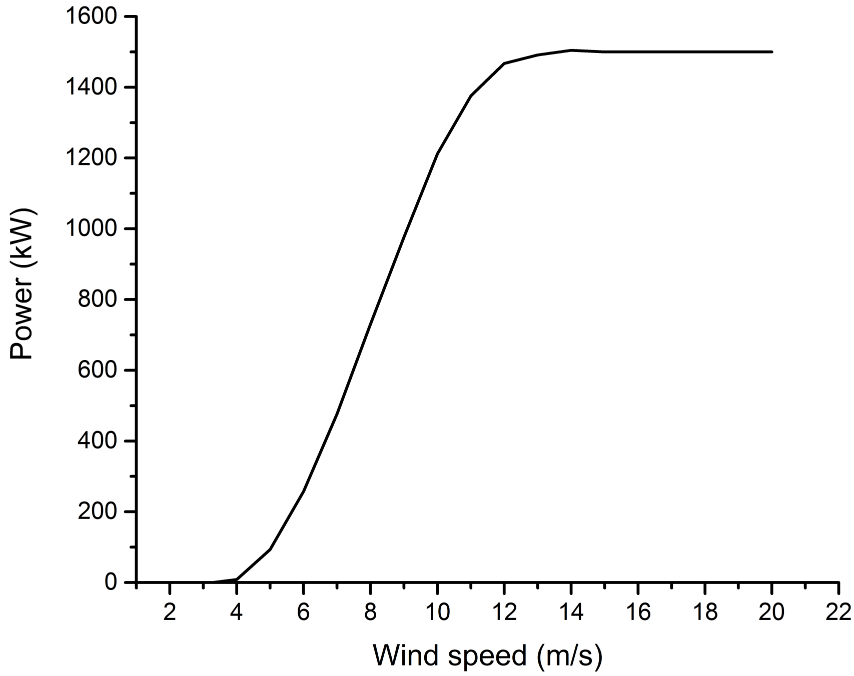


Figure 3.1: Power Curve of the wind turbine

Table 3.3: Power values for various wind speeds from manufacturer's power curve

Sl. No	Wind speed (m/s)	Power (kW)
1	5	100
2	6	250
3	7	480
4	8	730
5	9	980
6	10	1200
7	11	1400
8	12	1450
9	13	1500
10	14	1500
11	15	1500
12	16	1500

Model-2

In this model it is assumed that, for a typical wind turbine, the power generation starts at cut-in wind speed v_c, the power output increases linearly for wind speed between cut-in and rated wind speed v_r and then, a constant rated power is produced between rated wind speeds v_r to cut-out wind speed v_f. The set of characteristic equations in their general form are given in Equation 3.3.

$$P_e = \begin{cases} 0 & \text{for } v < v_c; \\ P_{er}\frac{v-v_c}{v_r-v_c} & \text{for } v_c \leq v \leq v_r; \\ P_{er} & \text{for } v_r \leq v \leq v_f; \\ 0 & \text{for } v > v_f. \end{cases} \tag{3.3}$$

where P_e is the power output of the turbine for a given wind speed v and P_{er} is the rated power of the wind turbine (Abouzahr and Ramakumar 1990).

Models based on actual power curve supplied by the manufacturer

In the present work, two curve fitting techniques namely, method of least squares and cubic spline interpolation have been investigated to fit the data available in the manufacture′s power curve of the wind turbine.

a)Model-3 It is a mathematical procedure of fitting a best curve for the given set of data points in such a way that the sum of squares of offset of the points from the curve are minimum. It has been proposed in the literature (Ai *et al.* 2003) to use Equation 3.4 to predict the power output of a wind turbine.

The equations developed in the present work based on this approach for the turbine under study are as given in Equation 3.5

b) Model-4 A cubic spline is a spline constructed of piecewise third-order polynomials which pass through a set of m control points. The characteristic equations in their general form can be expressed as Equation 3.6 (Diaf *et al.* 2007),(Hocaoğlu *et al.* 2009). The set of equations developed in the present study for the turbine under study using the above approach are as given in Equation 3.7.

$$P_e = \begin{cases} 0 & \text{for } v < v_c; \\ a_1 v^2 + b_1 v + c_1 & \text{for } v_c \leq v \leq v_1; \\ a_2 v^2 + b_2 v + c_2 & \text{for } v_1 \leq v \leq v_r; \\ a_3 v^2 + b_3 v + c_3 & \text{for } v_r \leq v \leq v_f; \\ 0 & \text{for } v > v_f. \end{cases} \tag{3.4}$$

where a_1,b_1,c_1, a_2,b_2,c_2, a_3,b_3 and c_3 are the coefficients

$$P_e = \begin{cases} 0 & \text{for } v < 4; \\ 35v^2 \ - 231v \ + 380 & \text{for } 4 \leq v \leq 7; \\ -30v^2 \ + 786v \ - 3648 & \text{for } 7 \leq v \leq 13; \\ 0v^2 \ + 0v \ + 1500 & \text{for } 13 \leq v \leq 20; \\ 0 & \text{for } v > 20. \end{cases} \tag{3.5}$$

$$P_e = \begin{cases} 0 & \text{for } v < v_c; \\ a_1v^3 + b_1v^2 + c_1v + d_1 & \text{for } v_c \leq v \leq v_1; \\ a_2v^3 + b_2v^2 + c_2v + d_2 & \text{for } v_1 \leq v \leq v_2; \\ . & \\ . & \\ . & \\ a_nv^3 + b_nv^2 + c_nv + d_n & \text{for } v_{n-1} \leq v \leq v_r; \\ P_r & \text{for } v_r \leq v < v_f. \end{cases} \tag{3.6}$$

where a_1, b_1, c_1 ,d_1,a_2, b_2, c_2 ,d_2, a_n, b_n, c_n and d_n are the polynomial coefficients of cubic spline interpolation functions, n is the number of cubic spline interpolation functions corresponding to n+ 1 values of data points.

$$P_e = \begin{cases} 0 & \text{for } v < 4; \\ 11.33v^3 - 101.97v^2 + 314.58v - 331.92 & \text{for } 4 \leq v \leq 5; \\ 3.27v^3 - 5.25v^2 - 72.287v + 183.865 & \text{for } 5 \leq v \leq 6; \\ 5.34v^3 - 36.12v^2 - 88.61v + 928.54 & \text{for } 6 \leq v \leq 7; \\ -20v^3 + 420v^2 - 2690v + 5590 & \text{for } 7 \leq v \leq 8; \\ 2v^3 - 42v^2 + 542v - 1942 & \text{for } 8 \leq v \leq 9; \\ -10v^3 + 246v^2 - 1762v + 4202 & \text{for } 9 \leq v \leq 10 \end{cases} \tag{3.7}$$

In non-parametric models, the physical or analytical input-output relationship is not based on any pre-defined assumptions as in parametric models. But the correlation between the input and output is derived from the data provided. Hence non-parametric models define the input-output relationship in terms of many parameters and are more flexible. One of such popular non-parametric model is, response surface methodology and is investigated in this study.

3.3 MODEL-5

In Response Surface Methodology (RSM), the mathematical relationship between independent input variables and the dependent output variable are obtained in terms of a second order model of the form $y = ax^2 + bx + c$

where, y is the predicted response; x is the input variable that influences the response variable. Wind speed, air density, blade pitch angle, rotor speed and wind direction have been used as the input parameters in developing Model-5 based on RSM in contrast to use of only wind speed in other models. The regression model development has been carried out using Statistica 12.0 (StatSoft) software (StatSoft 2001). 99% confidence level has been set and backward elimination method has been used in this work.

3.4 RESULTS AND DISCUSSION

In the present study, five conventional models have been developed to predict wind turbine power. The results have been discussed in this section with the help of randomly picked set of values as shown in Table 3.4, that cover cut-in to cut-out wind speed interval in the data collected from the SCADA of the wind turbine and that lies in the test set used in RSM technique. The direct comparison of predicted power by the model with actual power generated by the turbine, helps in checking the relevance of different models.

Root Mean Square Error (RMSE) which is defined as the square root of the mean of the squared difference between the actual and predicted power values is used as the performance metric to evaluate the performance of the five models. The RMSE is expressed in Equation 3.8.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(P_e\left(i\right) - P_a\left(i\right)\right)^2} \tag{3.8}$$

Table 3.4: Data selected for comparison of performances of the models

Sl.no	Wind speed (m/s)	Actual power (kW)
1	5.66	250.85
2	6.01	289.38
3	7	447.43
4	8.03	637.13
5	9	820.9
6	10	981.15
7	11.017	1201.06
8	12	1392.21
9	13	1473.73
10	14	1500.18
11	15.23	1487.85
12	16.11	1475.53

where P_e and P_a are estimated and actual power in kW respectively. N is the total number of data.

3.4.1 Model-1

This model uses Equation 3.2 and it results in wind power values too far from actual values. e.g: for a wind speed of 5.66 m/s, the equation results in wind power of 313924.62 kW in contrast to 250.85 kW of actual power produced by the turbine. This huge difference is due to the assumption of 100% mechanical to electrical conversion efficiency by the turbine and theoretical highest value of C_p of 0.593 used in this model. Both of the assumptions has led to erroneous results as these assumptions depict ideal condition and is impossible to be attained in practice. Some of the other models based on wind power equation takes into consideration the efficiencies (mechanical transmission and generator) and the power factor, which is a function of blade pitch angle, rotational speed of the turbine as well as angle of attack. Due to interdependency of the above-mentioned parameters and their variations based on the wind speed and other climatic conditions, these models become complex and erroneous (Thapar *et al.* 2011).

3.4.2 Model-2

A model based on linear power curve proposed by Abouzaher et al. has been used. The resulting power output values are comparable with actual as shown in Table 3.5. These models are not very accurate, since the characteristic equations evolved are more general and not specific to any turbine, hence does not replicate the performance of a specific turbine very clearly.

3.4.3 Model-3 and Model-4

The predicted values of power by Model-3 and Model-4 have been presented in Table 3.5. Though the techniques are efficient in fitting the curve as is evident from predicted power and values of power provided by manufacturer′s power curve in Tables 3.3 and 3.5, the RMSE values for both the techniques are quite high with respect to actual power values. The main reason for this is the difference between power values provided by manufacturer′s power curve and actual power produced by the turbine. This is because, the manufacturer′s power curve is derived under standard conditions. The power produced by the turbine under complex weather conditions is quite different from that of the standard. For example, for a wind speed of 10 m/s, the value of power in manufacturer′s power curve from Table 3.3 is 1200 kW but actual power from Table 3.4 is 981.15 kW. In addition to this there could be error due to the curve fitting method adopted in the models. Because a finite set of parameters are assumed in parametric models, these are not very flexible and are restricted in their nature.

3.4.4 Model-5

The analysis of response variable namely wind turbine power output can be done through surface plots obtained from RSM study. The typical three-dimensional (3D) surface plots for wind turbine power in terms of the process variables are shown in Figures 3.2-

3.5. Figure 3.2 illustrates the surface plot for power by varying two variables namely wind speed and pitch angle. It is clear from the figure that power increases with increase in both wind speed and pitch angle. This is because from the wind power Equation 1.1 it is clear that power is directly proportional to cubic power of wind speed. Power coefficient C_p of turbine is a function of blade pitch angle, hence pitch control is widely used to regulate the power output of a wind turbine. For a turbine, for a given wind speed, there is an optimum blade pitch angle at which the power output is maximum (Manwell *et al.* 2010). It can be established from Figure 3.4 that in the given region there is little variation in air density, hence it has no significant effect on power. From Figure 3.3 it can be seen that power produced is maximum when wind blows from North which is set as zero degree and decreases slightly in both directions away from North. It is clear from Figure 3.5 that power of a wind turbine increases with the rotor speed as the generator power is the function of rotor speed. The R^2 value obtained is 0.9954.

The resulting RSM equation to predict the power output of a turbine for the present study, considering second order model is given by Equation 3.9.

$$Y = 3356 + 310.8X_1 + 126.2X_2 + 1.34X_3 + 738.4X_4 866X_5 - 0.8122X_3^2 + 35.02X_5^2 \\ -7.7526X_1X_2 - 0.882X_1X_3 - 8.99X_1X_5 - 2.434X_2X_5 \tag{3.9}$$

where, X_1, X_2, X_3, X_4, X_5 are wind speed, blade pitch angle, wind direction, density and rotor speed respectively and Y is the wind turbine power output.

From the Table 3.5, it can be noted that the values of power predicted by RSM model are much closer to that of actual power. Since 2522 and 444 data collected from SCADA have been used for training and testing the RSM model, the mean RMSE for training and test data are 14.07 and 15.29 respectively for Model-5.

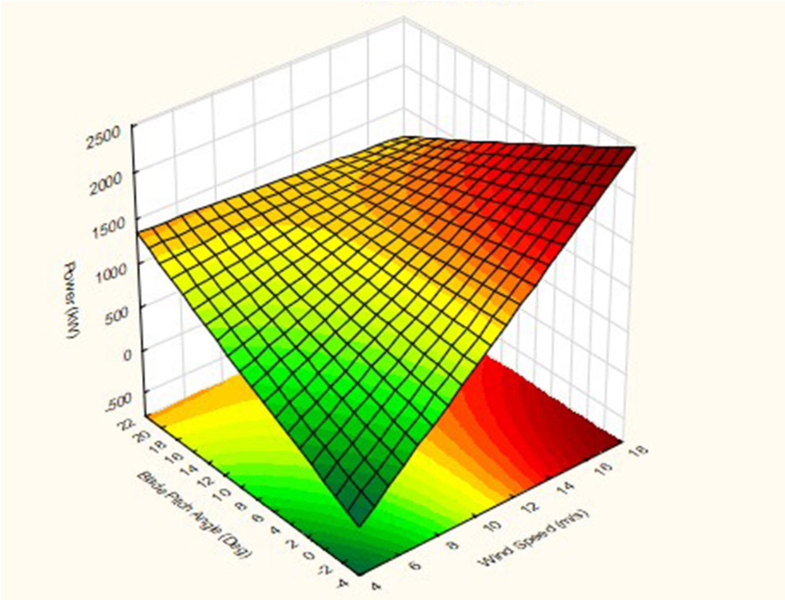


Figure 3.2: Estimated 3D response surface plot for wind turbine power(Power vs Blade pitch angle and Wind speed)

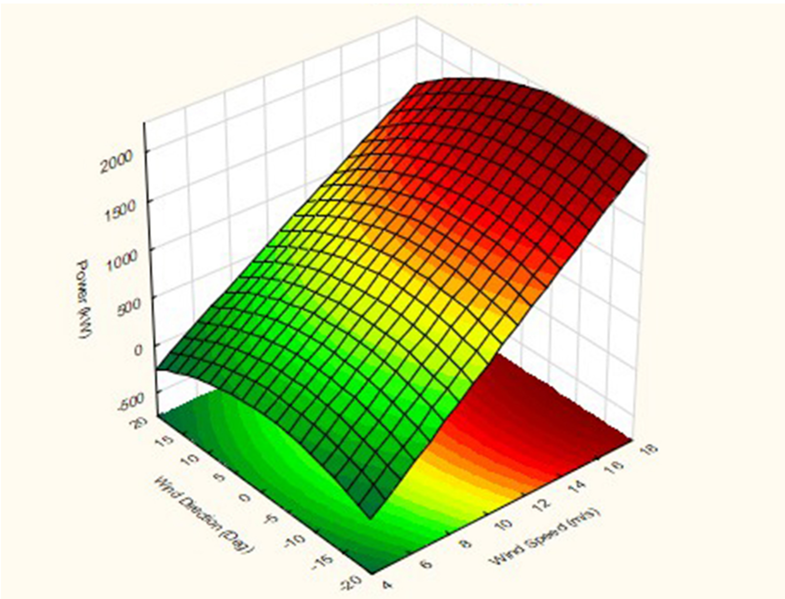


Figure 3.3: Estimated 3D response surface plot for wind turbine power(Power vs Wind direction and Wind speed)

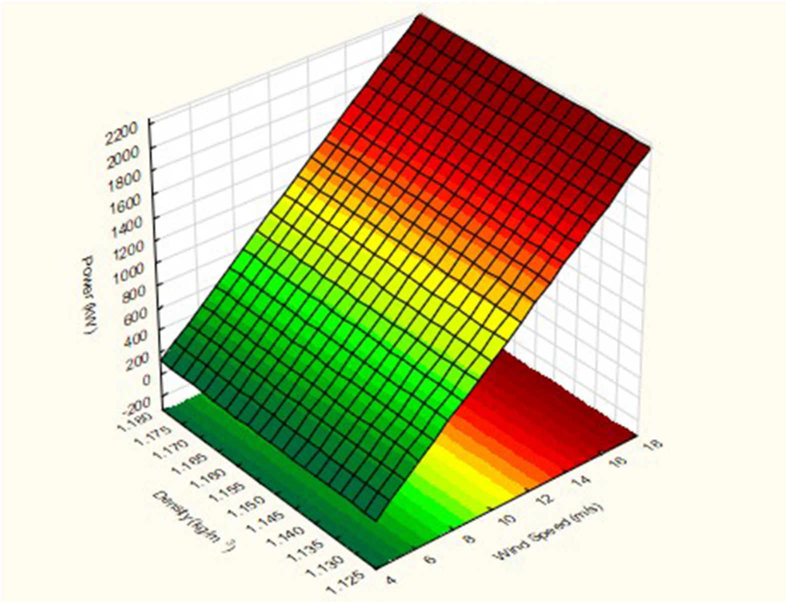


Figure 3.4: Estimated 3D response surface plot for wind turbine power(Power vs Density and Wind speed)

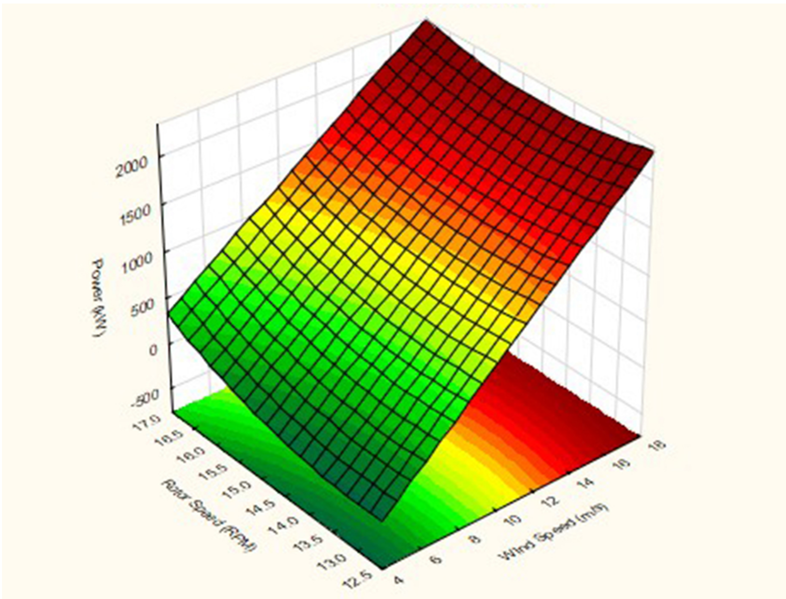


Figure 3.5: Estimated 3D response surface plot for wind turbine power(Power vs Rotor speed and Wind speed)

Table 3.5: Comparison of power predicted by different modeling methods

Sl. No	Wind speed m/s	Power (kW)	Model-2		Model- 3		Model- 4		Model- 5	
			Power (kW)	RMSE	Power (kW)	RMSE	Power (kW)	RMSE	Power (kW)	RMSE
1	5.66	250.85	250.05	0.565	200.61	35.52	200.73	35.43	222.28	20.19
2	6.01	289.38	302.55	9.31	263.24	18.48	253.24	25.55	280.07	6.57
3	7	447.43	450	1.81	485	26.56	480	23.02	425.91	15.21
4	8.03	637.13	604.95	22.75	730.06	65.71	716.20	55.91	625.30	8.36
5	9	820.9	750	50.13	996	123.81	992	120.98	786.48	24.33
6	10	981.15	900	57.38	1212	163.23	1278	209.90	960.39	14.67
7	11.01	1201.06	1052.55	105.01	1370.13	119.54	1401.78	141.92	1171.69	20.76
8	12	1392.21	1200	135.91	1464	50.75	1450	40.85	1346.63	32.23
9	13	1473.73	1350	87.49	1500	18.57	1500	18.57	1437.08	25.91
10	14	1500.18	1500	0.129	1500	0.12	1500	0.129	1476.85	16.49
11	15.23	1487.85	1500	8.59	1500	8.59	1500	8.59	1478.45	6.64
12	16.11	1475.53	1500	17.30	1500	17.30	1500	17.30	1468.35	5.07

Out of the five conventional models developed, which include models based on wind power equation, based on the power curve and based on RSM, the Model-1 which is based on fundamental equation of the wind power is found to be erroneous due to the theoretical assumptions made. The models based on power curve are found to be less accurate due to the difference in actual power and that provided by manufacturer's power curve and due to the restricted nature of the curve fitting techniques. Model-5 based on RSM has shown good agreement between simulated and measured values of power, as it considers other important variables affecting the power in addition to wind speed.

CHAPTER 4

ANN BASED MODELING TECHNIQUES

This chapter provides the details of different learning algorithms namely Backpropagation (BP) and Extreme Learning Machine (ELM) algorithm used in training Artificial Neural Network (ANN) models. Two widely used classes of neural network namely Multilayer Perceptron (MLP) neural network , Radial Basis Function (RBF) neural network and various center fixing strategies used in development of RBF neural network have been discussed in detail. An online learning approach for training neural network, namely Online Sequential Extreme Learning Machine (OSELM) algorithm used in this study is presented. The details of model development using various techniques is discussed followed by results of the ANN models developed.

This work tries to comprehensively investigate feed forward neural network models, namely MLP neural network and RBF neural network, by considering two learning algorithms namely BP and ELM. RBF neural network follows a different learning principle than MLP neural network, which involves two important aspects namely selection and location of number of centers and selection of width of the RBF units. Four different center selection strategies namely fixed centers selected at random, use of Fuzzy C-means (FCM), Conditional Fuzzy C-means (CFCM) algorithm and Particle Swarm Optimization (PSO) based FCM Algorithm (PSO-FCM) have been studied.

4.1 LEARNING ALGORITHMS

The learning in ANN leads to modification of synaptic weights and biases of the network in an orderly manner to attain the desired objective. Two very widely used learning algorithms used in feed forward neural networks namely Gradient descent or BP

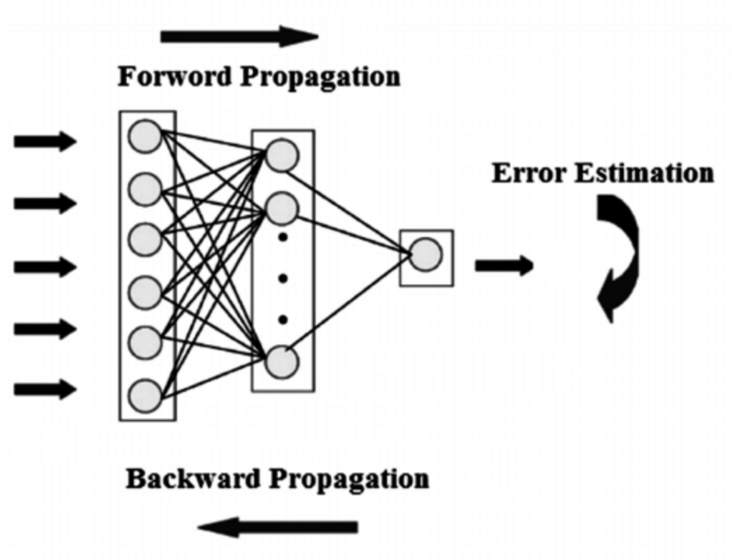


Figure 4.1: Backpropagation Algorithm

Source: (Morteza 2018)

and ELM have been used and compared in this study.

4.1.1 Backpropagation learning algorithm

Backpropagation learning algorithm is one of the popular error correction learning algorithms. This algorithm works as a closed feedback loop system as shown in Figure 4.1. Let, neuron k constitute the only computational node in the output layer of the feed forward network, n denotes time step of an iterative process involved in adjusting the synaptic weights of neuron k. In this algorithm, the network output $O_k(n)$ is compared with the desired response $y_k(n)$. This difference is the error signal represented by $e_k(n)$ which actuates a control mechanism. The purpose of this is to apply a corrective adjustment to the synaptic weights of neuron k, thus making the output signal $O_k(n)$ come closer to the desired response $y_k(n)$ in a step by step manner (Hagan *et al.* 1996).

This objective is achieved by minimizing a cost function E as given in Equation 4.1, defined in terms of the error signal $e_k(n)$.

$$E = \frac{1}{2} e_k(n)^2 \tag{4.1}$$

The Gradient descent algorithm works as follows:

The error signal denoted by e_k given in Equation 4.2, actuates the mechanism.

$$e_k(n) = (O_k(n) - y_k(n)) \tag{4.2}$$

The modification in the output layer weights are calculated as given in Equations 4.3 and 4.4.

$$\delta_k(n) = (O_k(n) - y_k(n))O_k(n)(1 - O_k(n)) \tag{4.3}$$

$$\Delta w_{kj}(n) = \delta_k(n)V_j(n)\eta\alpha \tag{4.4}$$

where, α is the momentum parameter

η is the learning rate

$V_j(n)$ is the hidden layer output

$y_k(n)$ is the target output

$\Delta w_{kj}(n)$ is the adjustment applied to the synaptic weights

Here, momentum rate α helps to decide the step size to avoid the problem of getting stuck in local minima, the learning rate η, controls the changes in the model in the learning process.

The updated value of the synaptic weights between hidden and output layer are determined by Equation 4.5

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n) \tag{4.5}$$

Similar steps are followed to update the synaptic weights between hidden and input layer.

Modification of the output weights will be terminated, when the stopping criteria is reached. In this study, the stopping criteria is set as either minimum specified error or maximum number of epochs. To ensure the stability or the convergence of the iterative process, it is important that η and α are carefully selected.

This algorithm suffers from many drawbacks, the primary limitation is its slow rate of convergence due to iterative learning steps. The algorithm easily converges to local minima, which is undesirable if the local minimum is located far above global minima. A suitable stopping criteria in cost function minimization procedure should be properly set. This may otherwise lead to over training of the network, leading to worst generalization performance (Huang *et al.* 2004; Hu and Hwang 2001). Further, tuning of the simulation parameters, namely learning and momentum parameters play a key role in the performance of this algorithm.

4.1.2 Extreme Learning Machine (ELM) algorithm

Extreme learning machine algorithm is a powerful non-iterative algorithm having universal approximation. It uses a generalized inverse (Moore- Penrose) operation on the hidden layer output matrix in order to determine the output weights. This algorithm calls for tuning of less number of network parameters, hence needs less human intervention. Further, it has demonstrated excellent generalization capabilities.

Considering a set of N data samples (x_i, t_i), where x and t are the input and target values respectively. where, $x_i = [x_{i1}, x_{i2}, \dots, x_{in}] \in R^n$ and $t_i = [t_{i1}, t_{i2}, \dots t_{im}] \in R^m$. Then, the network output of the single layer feedback neural network, with S hidden nodes and $f(x)$ activation function is as given in Equation 4.6.

$$O_j = \Sigma_{i=1}^{S} \beta_i f(w_i.x_j + b_i) \tag{4.6}$$

$j = 1, 2, \ldots, N$

where $w_i = [w_{i1}, w_{i2}, \ldots, w_{in}]^T$ is the weight vector connecting the input and hidden nodes, $\beta_i = [\beta_{i1}, \beta_{i2} \ldots \beta_{im}]^T$ is the weight vector connecting the hidden and output nodes and b_i is the bias. Equation 4.6 can be simplified into Equation 4.7.

$$H\beta = T \tag{4.7}$$

where,

H, β and T are as given in Equations 4.8, 4.9 and 4.10 respectively.

$$H = \begin{bmatrix} f(w_1 x_1 + b_1) & \ldots & f(w_S x_1 + b_S) \\ . & & . \\ . & \ldots & . \\ . & & . \\ f(w_1 x_N + b_1) & \ldots & f(w_S x_N + b_S) \end{bmatrix}_{NXS} \tag{4.8}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ . \\ . \\ . \\ \beta_S^T \end{bmatrix}_{SXm} \tag{4.9}$$

$$T = \begin{bmatrix} t_1^T \\ . \\ . \\ . \\ t_N^T \end{bmatrix}_{NXm} \tag{4.10}$$

The output weight matrix is computed using Equation 4.11.

$$\beta = H^{\dagger} T \tag{4.11}$$

where $H^{\dagger}$ is the Moore- Penrose generalized inverse matrix of H, which can be determined using Equation 4.12 (Huang *et al.* 2004).

$$H^{\dagger} = (H^T H)^{-1} H^T \tag{4.12}$$

4.2 MULTILAYER PERCEPTRON (MLP) NEURAL NETWORK

MLP are an important class of neural networks, that consists of input layer, one or more hidden layers and an output layer. The input signal propagates through the network in a forward direction, on a layer-by-layer basis. The hidden layer neuron consists of a nonlinear activation function, commonly a sigmoid function is used. The hidden neurons extract meaningful features from the input patterns, thereby enabling the network to learn the complex tasks. They construct global approximations to nonlinear input-output mapping (Haykin and Network 2004).

MLP algorithm is given as follows:

1. Initialize the weights and biases to small random values.

2. Choose the input pattern set $\{x_i^\mu, y_k^\mu\}$ from the training patterns. Present x_i^μ to the input layer and y_k^μ to the output layer, where , $\mu = 1, 2, \ldots n$ are the number of patterns $i = 1, 2, \ldots p$ are the number of input features and $k = 1, 2, \ldots r$ are the number of output features.

3. Compute the activation of the neurons in the hidden layer using Equation 4.13.

$$V_j^\mu = \frac{1}{1 + e^{-\Sigma w_{ji} x_i + b}} \tag{4.13}$$

   where, $j = 1, 2, 3 \ldots s$ are the number of hidden layer neurons.

4. Compute the output of each neuron in the output layer using Equation 4.14.

$$O_k^\mu = \frac{1}{1 + e^{-\Sigma w_{kj} V_j + b}} \tag{4.14}$$

where, b is the bias, w_{ji} is the weights between hidden and input layer and w_{kj} is the weights between output and hidden layer, which will be updated during learning as discussed in Section 4.1.1 (Pai 2004).

4.3 RADIAL BASIS FUNCTION (RBF) NEURAL NETWORK

RBF neural network works on a different approach, where learning is equivalent to finding a surface in a multidimensional space, that provides a best fit to the training data. The single hidden layer of an RBF neural network uses Gaussian activation function, where it computes the Eucledian norm between the input vector and the center of that unit. Each neuron of the hidden layer has two parameters, a center x_i and a width σ_i. The network input vector is compared with the center to produce a radially symmetrical response. The width controls the smoothness properties of the interpolating function.

Response of the hidden layer are scaled by the connection weights of the output layer and then combined to produce the network output (Haykin and Network 2004).

The basic algorithm for an RBF neural network is as given below:

1. Select the number of RBF units.

2. Initialize their centers x_j from input data using different strategies. Initialize the weights and bias to small random values.

3. Choose the input-output pattern set $\{x_i^\mu, y_k^\mu\}$ where, $\mu = 1, 2, 3 \ldots n$ are the number of patterns and $i = 1, 2, 3 \ldots p$ are the number of input features, $k = 1, 2, 3 \ldots r$ are the number of output features.

4. Compute the hidden layer output using Equation 4.15

$$V_j = e^{-\frac{1}{2\sigma^2}\|x_j - x_i\|^2} \tag{4.15}$$

   where, x_i is the input pattern x_j is the center and σ_i is width of the RBF unit, where $j = 1, 2, 3, \ldots c$ are the number of centers.

   The width σ_i controls the smoothness properties of the interpolating function in the radial basis function neural network.

5. Compute the output of each neuron in the output layer using Equation 4.16.

$$O_k^\mu = \frac{1}{1 + e^{-\Sigma w_{kj} V_j + b}} \tag{4.16}$$

   where b is the bias and w_{kj} is the weights between output and hidden layer, which will be updated during learning.

(Pai 2004),(Nagabhushana 1996)

4.4 CENTER FIXING STRATEGIES

In designing a RBF neural network, there are different learning strategies that can be followed, depending on how the centers of the RBFs of the network are specified. In this study four different learning strategies namely Fixed centers selected at random, Fuzzy C-means algorithm (FCM), Conditional Fuzzy C-means (CFCM) algorithm and PSO based FCM algorithm (PSO-FCM) have been used.

4.4.1 Fixed centers selected at random

This is a simplest approach, in which the locations of the centers are chosen randomly from the training data set. For higher accuracy it requires large training set of fixed size, which are distributed in representative manner for the problem in hand (Haykin and Network 2004). The only parameter that is learnt is the linear output weights, which can be done by a suitable learning algorithm.

4.4.2 Fuzzy C-means algorithm (FCM)

To overcome the limitations of the fixed centers, which requires large training set for better performance, a self organized selection of centers can be used. This estimates the appropriate location of the centers in the hidden layer. A clustering algorithm, which partitions the given set of the data points into subgroups is used. A fuzzy clustering algorithm in contrast to crisp clustering is suitable for real life problems, where the boundaries between natural classes may be overlapping. FCM is a very popular algorithm which has been widely used in pattern recognition, image processing etc. (Wang *et al.* 2006).

The objective function here is to minimize the sum of distances between the data points and the center of the cluster as given in Equation 4.17.

$$J = \Sigma_{k=1}^{N} \Sigma_{j=1}^{c} u_{kj}^{m} \parallel X_k - V_j \parallel^2 \tag{4.17}$$

where u_{kj} is the membership value, denotes the degree X_k belongs to cluster j, m is fuzziness parameter usually set to be 2, $X = X_1, X_2, \ldots X_N$ are the unlabeled data, V_j is the cluster center , c is the number of clusters.

This clustering technique assigns feature vectors X_k into c clusters. The certainty of the assignment of feature vector into various clusters is measured by the membership functions $u_{kj} \epsilon [0, 1]$, $1 \leq j \leq c$, which satisfy the condition in Equation 4.18

$$\Sigma_{j=1}^{c} u_{kj} = 1 \tag{4.18}$$

The centers and the membership values are calculated iteratively using Equations 4.19 and 4.20

$$V_j = \frac{\Sigma_{k=1}^{N} u_{kj}^{m} X_k}{\Sigma_{k=1}^{N} u_{kj}^{m}} \tag{4.19}$$

$$u_{kj} = \frac{1}{\Sigma_{i=1}^{c} \frac{\|X_k - V_j\|}{\|X_k - V_i\|}^{\frac{2}{m-1}}} \tag{4.20}$$

The details of this popularly known algorithm can be found in (Bezdek 2013).

4.4.3 Conditional Fuzzy C- means (CFCM)

The Fuzzy C-means clustering algorithm has a shortcoming associated with it that is, the learning method is completely unsupervised, the information about lable of pattern being expressed in the form of output variables is not used in the algorithm. The information about the output variable provided by the designer, based on domain knowledge, will guide in forming the relationship between the input and output variables.

The CFCM clustering method takes the advantage of the information of the output(s) available. Thus the structure in the input space is conditioned based upon some linguistic landmarks defined in the output space. In this sense, this approach is strongly designer-oriented and context sensitive. The algorithm is thus a modification of the general form of FCM. The modification is in terms of the conditioning aspect, that is introduced by taking into account the conditioning (output) variable assuming the values $f_1, f_2, ..f_n$ on the corresponding patterns. f_k represents the level of involvement of the unlabeled data x_k in the constructed clusters.

Let the linguistic term defined in the output space is expressed as a fuzzy set B, B: $R \in [0, 1]$. Then $f_k = B(y_k)$, k=1, 2,..n stands for degree of membership of y_k in B. The f_k is distributed additively across the entries of the k^{th} column of the pattern matrix such that, $\Sigma_{j=1}^{c} u_{jk} = f_k$.

The Conditional Fuzzy C-means clustering algorithm used to fix the centers of the RBF neural network is as given below.

1. Initialize the membership function $U^{(0)}$ = $\{u_{jk}\}, u_{jk}\epsilon[0, 1]$ data designed to j^{th} category, iter = 0.

   $v\{u_{jk}\epsilon[0, 1] \mid \Sigma_{j=1}^{c} u_{jk} = 1\ \forall_i$ and $0 < \Sigma_{k=1}^{n} u_{jk} < n\forall_j\}$.

2. iter = iter + 1, Calculate $V^{(iter)} = \{v_1, v_2, ..v_c\}$ are the cluster centers of fuzzy grades and $v_{j1} = \frac{\Sigma_{k=1}^{n} u_{jk}^2 x_{k1}}{\Sigma_{k=1}^{n} u_{jk}^2}$, $v_{j2} = \frac{\Sigma_{k=1}^{n} u_{jk}^2 x_{k2}}{\Sigma_{k=1}^{n} u_{jk}^2}$ and so on $v_{jz} = \frac{\Sigma_{k=1}^{n} u_{jk}^2 x_{jz}}{\Sigma_{k=1}^{n} u_{jk}^2}$ where, k =1, 2,...n are number of patterns, and j=1, 2, .. c are the number of clusters. $x = x_j$ is the feature vector where, $x_j = \{x_{j1}, x_{j2}...x_{jz}\}$

3. Find the new membership function $U^{(iter)} = \{u_{jk}\} = \frac{f_k}{\Sigma_{j=1}^{c} \frac{\|x_k - v_j\|}{\|x_k - v_i\|}^{\frac{2}{m-1}}}$,k=1,2,..n; j=1,2,..c. m is the fuzzification factor $m > 1.0$ usually taken as 2.

4. If $\|U^{(iter)} - U^{(iter+1)}\|$ < epsilon, stop. Otherwise $U^{(iter+1)} = U^{(iter)}$ and goto step 2 (Pedrycz 1998).

4.4.4 PSO based FCM algorithm (PSO-FCM)

FCM suffers from some problems such as getting trapped in the local minimum and slow convergence, since it is greatly influenced by the initial value of the membership functions. The swarm intelligence based algorithms are based on the collective behavior of insects, fish and birds and are robust to dynamic changes and are suitable for global optimization. Hence, these algorithms can be introduced in FCM to achieve global optimization and higher speed of convergence. Thus PSO-FCM is a hybrid clustering algorithm, which is a combination of a metaheuristic optimization algorithm namely Particle Swarm Optimization (PSO) and Fuzzy C-means (FCM) clustering algorithm.

PSO-FCM algorithm is given below:

1. Initialize M number of membership functions u_{kj} randomly, between upper and lower limits, which are 1 and 0 respectively. Initialization of u_{kj} satisfies the condition given in Equation 4.18. M is the population size. Calculate the centers using Equation 4.19.

2. Calculate the fitness value using the objective function of FCM described in Equation 4.17.

3. Update the membership function using Equation 4.20. Calculate the fitness value with respect to the updated membership function values.

4. Compare the fitness values calculated in Step 2 and 3. The best set of particles (which possesses the best fitness value) out of the two is considered pbest.

5. The best pbest among all the particles is considered gbest.

6. For each particle, update the velocity and position based on Equations 4.21 and 4.22.

$$u_i^{t+1} = w * u_i^t + C_1 rand1(pbest_i - S_i^t) + C_2 rand2(gbest - S_i^t) \qquad (4.21)$$

$$S_i^{t+1} = S_i^t + u_i^{t+1} \tag{4.22}$$

where $rand1$ and $rand2$ are uniformly distributed random variables, C_1 and C_2 are acceleration factors and w is inertia weight. Check the stopping criteria. Otherwise goto Step (2) and repeat the procedure until the stopping criteria is reached. Save the membership functions, calculate the final centers corresponding to it using Equation 4.19 (Chrouta *et al.* 2015).

4.5 PROPOSED PSO OPTIMIZED RBF NEURAL NETWORK MODEL

The metaheuristic optimization algorithms can also be used to optimize the model parameters such as number of centers and width of the RBF units in a RBF neural network model. Such an attempt has been done in the present study, by using the most widely used PSO optimization algorithm to optimize the model parameters. The objective function is maximizing the sum of prediction accuracies on training and test data.

The algorithm can be given as follows:

1. Number of particles (center and width) are randomly generated within the specified range.
2. The value generated by PSO for number of centers will be given as input to the clustering algorithm to select centers accordingly.
3. The centers thus generated along with the width values optimized by PSO in Step 1 is used in the RBF neural network model and fitness value of the objective function (sum of prediction accuracies on training and test data) is calculated.

4. Equations 4.21 and 4.22 will be used to update the velocity and position of the particles.

   These steps are repeated until the stopping criterion is reached.

4.6 ONLINE SEQUENTIAL EXTREME LEARNING MACHINE (OSELM) ALGORITHM

The sequential arrival of data in wind turbine appeals for online sequential learning instead of batch learning. The OSELM algorithm discards the training observation(s) immediately as the learning action for that observation(s) is complete.

OSELM works on two steps namely initialization and a sequential learning phase (Liang *et al.* 2006).

Step 1. Initialization phase

a) A small block of training data N0, is fed to the network for initial training with the condition that N0 is always greater than S. where S is the number of hidden nodes.

b) Allocate random values for weights and biases (for Sigmoid hidden nodes), centers and width (for RBF hidden nodes) values.

c) Obtain hidden layer output based on Equation 4.8.

d) Compute the initial output weight matrix using Equation 4.23.

$$\beta^{(0)} = P_0 H_0^T T_0 \tag{4.23}$$

where P_0 and T_0 are given by Equations 4.24 and 4.25 respectively. H_0 is the hidden layer output for initial block of data N0.

$$P_0 = (H_0^T H_0)^{-1} \tag{4.24}$$

Step 2. Sequential learning phase:

Present the $(k+1)^{th}$ chunk of new data

a) Compute the output of hidden layer by using Equation 4.26.

$$T_0 = \begin{bmatrix} t_1 \\ . \\ . \\ . \\ t_{N0} \end{bmatrix}^T \tag{4.25}$$

$$H_{k+1} = \begin{bmatrix} f(w_1 x_{(\sum_{j=0}^{k} N_j)+1} + b_1) & \dots & f(w_S x_{(\sum_{j=0}^{k} N_j)+1} + b_S) \\ . & & . \\ . & \dots & . \\ . & & . \\ f(w_1 x_{(\sum_{j=0}^{k+1} N_j)} + b_1) & \dots & f(w_S x_{(\sum_{j=0}^{k+1} N_j)} + b_S) \end{bmatrix}_{N_{k+1} X S} \tag{4.26}$$

where $N_{k+1} = \{(x_i, t_i)\}_{i=(\sum_{j=0}^{k} N_j)+1}^{\sum_{j=0}^{k+1} N_j}$ is the number of observations in $(k+1)^{th}$ chunk.

Set $T_{k+1} = [t_{(\sum_{j=0}^{k} N_j)+1}, ..., t_{\sum_{j=0}^{k+1} N_j}]^T$.

b) Determine the output weight matrix β^{k+1} using Equation 4.28 where, P_{K+1} can be determined by 4.27.

$$P_{K+1} = P_K - P_K H_{K+1}^T \big(I + H_{K+1} P_K H_{K+1}^T\big)^{-1} H_{K+1} P_K \tag{4.27}$$

$$\beta^{K+1} = \beta^{(K)} + P_{K+1} H_{K+1}^T (T_{K+1} - H_{K+1} \beta^{(K)}) \tag{4.28}$$

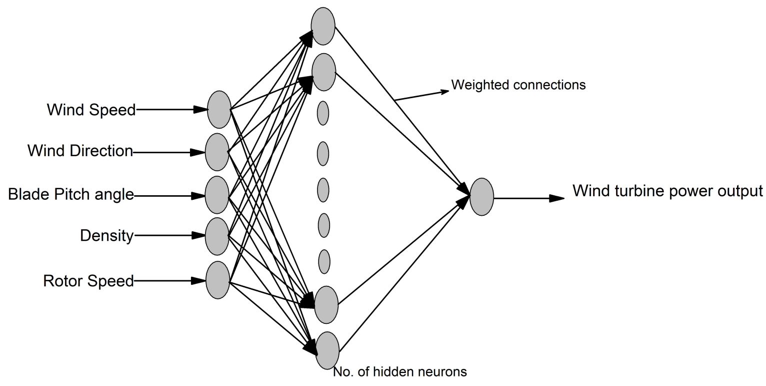


Figure 4.2: General architecture of the neural network model

4.7 MODEL DEVELOPMENT

The main objective of this study is to develop an ANN model to predict the power output of the wind turbine by using five input variables namely wind speed, density, blade pitch angle, wind direction and rotor speed. The general architecture of the neural network model is as shown in the Figure 4.2. Two popular types of feed forward neural networks namely MLP neural network and RBF neural network have been studied in this research. A recently developed ELM learning algorithm has been used in developing the neural network models in addition to the conventional BP learning algorithm, to explore its advantages. To study the performance and capability of online learning in wind power and modeling applications, OSELM algorithm is used.

A detailed study on RBF neural network model development is the novelty of this work. The RBF neural network model development involves three important aspects namely selection and location of centers, number of centers and selection of width of the RBF units. Different center fixing strategies namely, centers fixed randomly, FCM and CFCM algorithms have been used.

Further, in view of studying the advantages of using the metaheuristic algorithms in ANN model development, a widely used PSO algorithm has been used in developing a hybrid clustering algorithm namely PSO-FCM and to optimize the RBF neural network model parameters. A hybrid clustering algorithm namely PSO-FCM clustering algorithm, which is a combination of PSO and FCM has been used in the present study to overcome the drawbacks of FCM algorithm. Model parameters of a RBF neural network model namely number of centers and width of the RBF units, which otherwise is selected on trial and error basis have been optimized. Thus this effort is towards developing a fully tuned neural network model without any human intervention. All the simulations have been carried out using customized codes in MATLAB R2014a environment on a personal computer with an Intel i5-6200U, 2.3 GHz CPU and 4 GB RAM.

The details of the ANN models developed and compared in this study are given in Table 4.2.

4.7.1 Models using BP

Four models have been trained using backpropagation learning algorithm. The stopping criteria used in this work is minimum error or maximum number of epochs that have been fixed as 1×10^{-3} and 1000 respectively. Various simulation parameters, η, α for MLP neural network and η, α and σ for RBF neural network models have been fixed on trial and error basis, during training based on maximum prediction accuracy.

Training the models have been carried out with different number of hidden layer neurons and centers for MLP neural network and RBF neural network models respectively. Model-6 which is a MLP neural network model with α and η values 0.019 and 0.002 respectively gave best results for 25 number of hidden neurons.

Model-7, Model-8 and Model-9 are RBF neural network models, for which the RBF units of the hidden layer have been selected using three strategies namely fixed centers selected at random,use of FCM and CFCM algorithms respectively. A combination of fixed width value of 0.11, α of 0.23 and η of 0.32 resulted in best performance for

Table 4.1: Details of the models using BP learning

Model	Optimal configuration	σ	α	η
Model-6	5:25:1	-	0.019	0.002
Model-7	5:1125:1	0.11	0.23	0.32
Model-8	5:1075:1	0.52	0.37	0.25
Model-9	5:1025:1	0.30	0.45	0.43

Model-7. The corresponding values of the simulation parameters set in developing all four models, as well as the optimal configurations of the model are detailed in Table 4.1.

4.7.2 Models using ELM learning

An important feature of ELM, which makes it more suitable for large complex applications is, training requires only 1 epoch and less number of simulation parameters to be adjusted. The details of the ANN models that use ELM learning including their optimal configuration have been presented in Table 4.3.

Model-10 is a MLP neural network model, where only the number of hidden neurons needs to be adjusted. Which has been fixed on trial and error basis. Model-11, Model-12, Model-13 and Model-14 are RBF neural network models for which the RBF units of the hidden layer have been selected using strategies namely fixed centers selected at random, use of FCM, CFCM and PSO-FCM algorithms respectively. The learning of these RBF neural network models by ELM has only one simulation parameter that is width of the RBF units which has been adjusted on trial and error basis during training based on maximum prediction accuracy.

Table 4.2: Details of the ANN based models

Model	Type of neural network model	Learning algorithm used
Model-6	MLP neural network	BP
Model-7	RBF neural network with fixed centers selected at random	BP
Model-8	RBF neural network with Fuzzy c means clustering algorithm	BP
Model-9	RBF neural network with Conditional Fuzzy c means clustering algorithm	BP
Model-10	MLP neural network	ELM
Model-11	RBF neural network with fixed centers selected at random	ELM
Model-12	RBF neural network with Fuzzy c means clustering algorithm	ELM
Model-13	RBF neural network with Conditional Fuzzy c means clustering algorithm	ELM
Model-14	RBF neural network with PSO-FCM clustering algorithm	ELM
Model-15	PSO Optimized RBF neural network with PSO-FCM clustering algorithm	ELM

Table 4.3: Optimal configuration for different models using ELM learning

Model	Optimal configuration	σ
Model-10	5:25:1	-
Model-11	5:75:1	0.11
Model-12	5:60:1	0.93
Model-13	5:50:1	0.89
Model-14	5:40:1	0.95
Model-15	5:38:1	0.66

Model-15 is a fully optimized model trained automatically because, there are no parameters to be adjusted manually. The two parameters, number of centers and width values have been optimized using PSO algorithm. The position of the centers has been optimized using hybrid PSO-FCM algorithm. Hence it greatly reduces the effort and saves time for the designer in selecting the optimum value of these parameters of RBF neural network models.

It can be observed from Table 4.3 that, a width value of 0.11 gave best performance for Model-11. The best performance for Model-12, Model-13 and Model-14 have been observed for width values of 0.93,0.89 and 0.95 respectively. The optimal number of centers and width values have been selected as 38 and 0.66 respectively by PSO algorithm for Model-15.

4.7.3 Models using OSELM

ELM and BP are the batch learning algorithms, which demand complete set of data to train the model. OSELM is a widely used online learning algorithm, which uses the data sequentially as it arrives. The simulation parameters to be set in OSELM are number of hidden neurons and chunk size. The number of hidden neurons is determined based on trial and error basis to achieve minimum RMSE (%). Decrease in RMSE (%) is observed as the number of hidden neurons increased. The lowest RMSE (%) is obtained for 30 neurons, and it increased thereafter. This behavior was observed in case of both Sigmoid and RBF activation functions. To study the effect of chunk size, a fixed chunk size of 1-by-1, 50-by-50 and a random chunk size in the range of [10,100] has been used.

4.8 RESULTS AND DISCUSSION

The Root Mean Square Error (RMSE) as given in Equation 4.29 is the performance metric used for assessing the performance of models (training), Mean Square Error (MSE) as given in Equation 4.30 has been used for evaluating the accuracy of the models. The MSE of 1% has been fixed as the acceptable error limit between actual and simulated values for calculating the prediction accuracy of the models (testing). The prediction accuracy of the models are then calculated as the percentage of the test patterns that are within the acceptable error limit out of the total number.

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(X_i - \widetilde{X}_i)^2} \tag{4.29}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(X_i - \widetilde{X}_i)^2 \tag{4.30}$$

where, X_i and $\widetilde{X}_i$ are the i^{th} component of the actual and predicted wind power output respectively and n is the length of the the vector.

4.8.1 Models using BP learning

Table 4.4 through Table 4.7 shows the results of four models trained using BP learning algorithm. Table 4.4 shows performance of Model-6 on training and test data for different number of neurons in the hidden layer. For 25 hidden layer neurons, the highest prediction accuracy on training and test data of 97.10% and 96.17% have been observed with minimum RMSE of 5.93%.

From Table 4.5 it can be noted that for Model-7, the RMSE decreased with the increase in number of RBF centers and reached a minimum of 5.23% and increased thereafter. Thus highest prediction accuracy on training and test data is observed for 1125 number of RBF centers. The value of RMSE of 4.82% and best prediction accuracy on training and test data are achieved for 1050 number of RBF centers in case of Model-8 as shown in Table 4.6. As noted in Table 4.7, the values of RMSE, best prediction accuracy on training and test data are 4.47%, 99.16% and 98.19% respectively for 1025 RBF centers in case of Model-9 .

The performances of the three RBF neural network models in comparison with Model-6, which is based on MLP neural network are found to be better. Model-9, that used CFCM clustering algorithm showed superior results, when compared to other two RBF neural network models, thus proving the significance of using a good clustering algorithm for the selection of number and position of centers in the hidden layer. But MLP neural network always resulted in a compact ANN model with only 25 hidden neurons.

Table 4.4: Performance of Model-6

Number of Hidden neurons	5	10	15	20	25	30
Accuracy on training data (%)	70.34	75.93	86.04	91.47	97.10	95.59
Accuracy on test data (%)	66.89	71.84	83.33	90.76	96.17	92.34
Epochs	1000	1000	1000	1000	1000	1000
Training time taken(s)	12.68	13.02	15.20	21.07	21.54	25.31
RMSE (%)	8.68	7.72	6.90	6.54	5.93	6.27

Table 4.5: Performance of Model-7

Number of centers	1000	1025	1050	1075	1100	1125	1150
Accuracy on training data (%)	89.61	91.03	93.49	94.60	95.44	97.06	94.52
Accuracy on test data (%)	80.18	86.48	90.76	92.56	94.81	96.39	92.79
Epochs	1000	1000	1000	1000	1000	1000	1000
Training time taken(s)	22.54	24.20	27.83	30.36	33.23	35.75	36.12
RMSE (%)	10.51	9.28	8.72	7.00	5.52	5.23	6.59

Table 4.6: Performance of Model-8

Number of centers	1000	1025	1050	1075	1100	1125	1150
Accuracy on training data (%)	96.86	97.22	98.13	98.05	96.74	95.75	93.81
Accuracy on test data (%)	92.79	94.81	97.97	95.49	94.14	93.39	92.34
Epochs	1000	1000	1000	1000	1000	1000	1000
Training time taken(s)	22.95	23.31	27.90	29.42	32.72	34.87	35.93
RMSE (%)	7.01	5.68	4.82	5.07	5.97	6.78	7.29

Table 4.7: Performance of Model-9

Number of centers	1000	1025	1050	1075	1100	1125	1150
Accuracy on training data (%)	97.89	99.16	98.57	97.97	97.02	96.59	94.40
Accuracy on test data (%)	95.14	98.19	95.27	94.14	93.91	93.01	92.11
Epochs	1000	1000	1000	1000	1000	1000	1000
Training time taken(s)	23.28	24.01	27.56	30.17	33.18	34.93	36.23
RMSE (%)	5.29	4.47	5.59	6.30	6.78	6.90	7.41

4.8.2 Models using ELM learning

The performance of the Models - 10, 11, 12 and 13, which use ELM learning algorithm have been presented in Table 4.8 through 4.11. It is observed that, for Model-10 the minimum RMSE of 2.61% is reached for 25 numbers of hidden neurons with prediction accuracy on training and test data of 99.92% and 99.54% respectively.

The best results for Model-11 are observed for 75 RBF centers in the hidden layer, taking 0.6314 s of time with prediction accuracy of 100 % and 99.78% on training and test data respectively. Model-12 took 0.5932 s of time with 100% prediction accuracy on both training and test data and a RMSE of 1.86% for 60 RBF centers in the hidden layer. A 100% prediction accuracy on training and test data can be observed for Model-13 with 50 RBF centers in the hidden layer. Thus, out of the three RBF neural network models discussed, Model-13 using CFCM clustering algorithm has resulted in accurate and a compact model with only 50 RBF centers in the hidden layer.

Model-14 has been developed by using a hybrid clustering algorithm namely PSO-FCM clustering algorithm for selecting the RBF units in the hidden layer. Swarm intelligence has been combined with FCM algorithm in this clustering algorithm, which

Table 4.8: Performance of Model-10

Number of centers	5	10	15	20	25	30
Accuracy on training data (%)	99.32	99.60	99.76	99.88	99.92	99.92
Accuracy on test data (%)	97.74	98.19	98.64	99.09	99.54	99.54
Training time taken(s)	0.3101	0.3645	0.3889	0.4475	0.4586	0.4943
RMSE (%)	4.03	3.94	3.78	2.90	2.61	2.69

Table 4.9: Performance of Model-11

Number of centers	55	60	65	70	75	80
Accuracy on training data (%)	99.36	99.44	99.84	99.92	100	100
Accuracy on test data (%)	99.09	99.32	99.55	99.55	99.78	99.78
Training time taken(s)	0.5910	0.5958	0.6023	0.6275	0.6314	0.6487
RMSE (%)	3.58	3.04	2.78	2.37	1.96	2.06

helps in overcoming the drawbacks of FCM algorithm with regard to slow convergence and getting trapped in local minima.

The performance of the Model-14 has been presented in Table 4.12. It can be observed that, RMSE value of 1.856% is reached for 40 RBF centers with 100% prediction accuracy both on training and test data.

In developing the Model-14, the number of centers and the width value of the RBF units has been selected based on trial and error basis to obtain best performance. In view of overcoming this tedious and time consuming operation, which also needs some prior knowledge, swarm intelligence has been further used to optimize the number of centers and width value. Thus moving a step ahead in neural network model development by optimizing the major aspects of RBF neural network model.

Thus, Model-15 is a fully optimized RBF neural network model in terms of position of centers, number of centers and width of RBF units, based on maximum prediction accuracy. The variation of the fitness value with number of iterations is shown in Figure 4.3. The best performance of the model is found for 38 number of centers and 0.66 width. Model-15 has resulted in 100% accuracies on training and test data with RMSE value of 1.733%. The actual and power predicted by Model-15 is shown in Figure 4.4.

Table 4.10: Performance of Model-12

Number of centers	35	40	45	50	55	60
Accuracy on training data (%)	99.58	99.88	99.92	99.96	100	100
Accuracy on test data (%)	99.55	99.78	99.78	99.78	99.78	100
Training time taken(s)	0.5032	0.5110	0.5356	0.5536	0.5678	0.5932
RMSE (%)	2.51	2.09	2.08	1.96	1.95	1.86

Table 4.11: Performance of Model-13

Number of centers	35	40	45	50	55	60
Accuracy on training data (%)	99.76	99.84	99.96	100	100	99.92
Accuracy on test data (%)	99.32	99.78	99.78	100	99.78	99.78
Training time taken(s)	0.5023	0.5179	0.5436	0.5613	0.5694	0.6105
RMSE (%)	2.58	2.17	1.96	1.86	1.94	1.98

Thus, Model-15 is a fully optimized compact and accurate neural network model, which takes least time for development from among all the models developed and results in 100% prediction accuracy.

4.8.3 Models using OSELM learning

To study the use of online learning in wind modeling and prediction applications, an effort has been made to exploit various modeling aspects of OSELM with respect to activation function and mode of learning. The results are compared with other models based on ELM and BP batch learning to understand the applicability of this technique.

The performance of three different cases of OSELM using 1-by-1, use of fixed chunk size of 50 and randomly varying chunk size in the range of (10,100) is compared by considering two widely used activation functions namely Sigmoid and RBF. The results are summarized in Table 4.13. From the table it is clear that for both Sigmoid and RBF activation functions, the time taken for training by using data, 1-by-1 is longest and is lesser for chunk- by-chunk case. It can be noted that time taken for RBF function is much higher in comparison to Sigmoid activation function. This is because Sigmoid function satisfies a property given in Equation 4.31, between its derivative f'

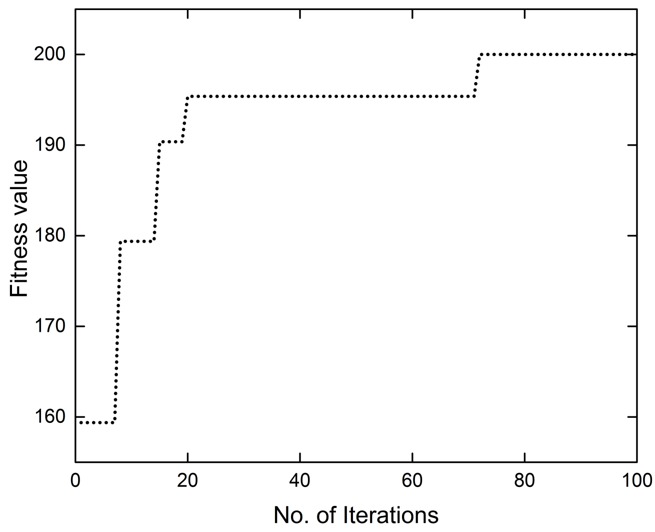


Figure 4.3: Variation of fitness with number of iterations

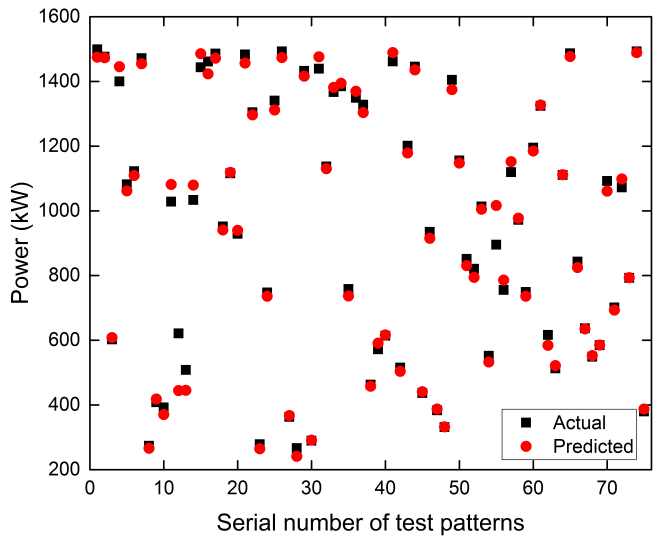


Figure 4.4: Comparison of actual power and power predicted from Model-15

Table 4.12: Performance of Model-14

Number of centers	35	40	45	50	55	60
Accuracy on training data (%)	99.96	100	100	100	100	100
Accuracy on test data (%)	99.78	100	99.78	99.78	99.78	99.78
Training time taken(s)	0.5060	0.5095	0.5328	0.5571	0.5590	0.6069
RMSE (%)	1.98	1.856	1.89	1.90	1.91	2.19

Table 4.13: Performance comparison of different cases of OSELM

Activation functions	Algorithm	Learning mode	Time (Seconds) functions	RMSE (%)	Accuracy on training data (%)	Accuracy on test data (%)
Sigmoid	OSELM	1-by-1	0.159	1.98	100	99.93
		50-by-50	0.080	1.96	100	99.95
		[10,100]	0.083	1.97	100	99.93
RBF	OSELM	1-by-1	0.462	2.01	100	99.89
		50-by-50	0.097	2.04	100	99.86
		[10,100]	0.092	2.01	100	99.93

and itself. Thus, it is simpler and easy to perform when compared to RBF which uses Gaussian function.

$$f^{'} = f(1-f) \tag{4.31}$$

The RMSE (%) values are slightly higher for RBF activation function compared to Sigmoid. However, there is no noticeable difference in the prediction accuracies on training and test data for both the activation functions. This is because the accuracy (%) calculation is based on a MSE limit of 0.01 kW^2. It is found that chunk-by-chunk case gives better performance for the considered wind data.

Two more activation functions namely Hardlim and Sin have been studied in OS-ELM and the performance is compared with that of Sigmoid and RBF in Table4.14. The results are tabulated for fixed chunk size of 50-by-50. Hardlim resulted in poor performance of 64.83% and 63.86% accuracy for training and test data respectively in comparison to the other three. It resulted in a RMSE value of 21.69 % which is a high

Table 4.14: Performance of OSELM with different activation functions

	RBF	Sigmoid	Sin	Hardlim
Accuracy on training data (%)	100	100	100	64.83
Accuracy on test data (%)	99.86	99.95	99.91	63.86
RMSE (%)	2.04	1.96	2.01	21.69
Training time (s)	0.097	0.080	0.076	0.083

value. Sigmoid activation function gives the best possible performance of 100% and 99.95% accuracy on training and test data respectively, with least RMSE value.

4.8.4 Discussion

A total of ten ANN models based on batch learning have been developed by considering different i) classes of ANN, ii) learning algorithms and iii) center fixing strategies. A fully optimized ANN model has been developed by using PSO optimization algorithm to optimize the various aspects of RBF neural network model. Models based on BP and ELM learning are discussed under separate sections. It is found that ELM learning is much faster compared to BP as it takes only one epoch. Good generalization performance and compact network architecture are the other two benefits of ELM based models as observed from this study. This is because ELM overcomes overfitting, underfitting and local minima problems in comparison to BP. It is observed that, MLP neural network models always result in a compact structure, as they construct global approximations to nonlinear input-output mapping in comparison to RBF neural network that construct local approximations.

Four different center fixing strategies namely fixed centers selected at random, use of FCM, CFCM and PSO-FCM algorithms have been used in RBF neural network model development. It is found that there is a definite impact of proper selection of centers of RBF on the performance of RBF neural network. Out of the four strategies studied, PSO-FCM algorithm has been found to be superior, as it combines the advantages of swarm intelligence and FCM. Further, the use of PSO in optimizing the model

Table 4.15: Comparison of performances of the best online and batch learning models

Model	RMSE (%)	Accuracy on training data (%)	Accuracy on test data (%)	Number of hidden neurons
Model-15	1.733	100	100	38
OSELM Model	1.96	100	99.95	30

parameters has resulted in compact and accurate model with less human effort.

A detailed study on using OSELM for wind power modeling and prediction throws light on various modeling aspects of this online learning algorithm. The online learning has its own advantages, in comparison to batch learning, such as eliminating the need for storing large amount of data to train the model. Hence is most suitable for simulation and modeling of wind data, where the data arrives in a sequential manner, through a SCADA.

Inspite of the above observations, it is worth comparing the performance of OS-ELM with the best batch learning algorithm proposed in the current study. The OSELM model with Sigmoid activation function and fixed chunk size of 50-by-50 is observed to be the best out of all other combinations. Model-15 a fully optimized neural network model showed best performance among all the other ANN models based on batch learning.

Hence the performance of these two models are compared in Table 4.15. From the table it can be seen that Model-15, a fully optimized RBF neural network model based on batch learning resulted in slightly better performance compared to the OSELM model. The OSELM model resulted in a compact structure with only 30 hidden neurons in comparison to Model-15. Hence based on the performance, it can be concluded that Model-15 is most feasible. However, the other advantages of OSELM over models based on batch learning certainly proves it be more suitable for this application because of the online nature of the wind data.

It can be noted that, to evaluate the accuracy of these models, MSE of 1% between simulated and actual values has been fixed as the acceptable error limit. The prediction accuracy of the models are then calculated as the percentage of the test patterns out of the total number that are within the acceptable error limit. Hence , 100 % prediction accuracy does not imply that all the predicted values are exactly same as actual, but are within the acceptable error limit (Pai *et al.* 2011).

4.9 COMPARISON OF CONVENTIONAL AND ANN MODELS FOR WIND POWER MODELING

In the present study, the model based on RSM performed better than the four other conventional models developed with least RMSE value. Hence can be termed as the most suitable conventional model in the present study.

This research is focused on developing a fast, accurate and compact neural network model for wind power modeling. Hence ten different batch learning ANN models have been developed and performances are compared. The study revealed that the models based on ELM are superior with respect to the learning speed, accuracy and the network structure. Out of the ELM based models, Model-15, a fully optimized RBF neural network model is found to be the best.

Further, in order to study the relevance of online learning in this application, an Online Sequential ELM (OSELM) has been studied and compared with Model-15, a best model based on batch learning. It is found that OSELM model resulted in marginal improvement in accuracy and a compact network structure compared to Model-15. Hence, OSELM model is found to be the most suitable model in the study.

A final comparison of the results are therefore done by taking into consideration Model-5, based on RSM a conventional modeling technique and OSELM model, the ANN model selected. It can be observed from Table 4.16 and Figure 4.5 that OSELM model resulted in lowest values of RMSE. The reason behind this being, ANN has universal approximation capability to approximate all kinds of non-linear functions and

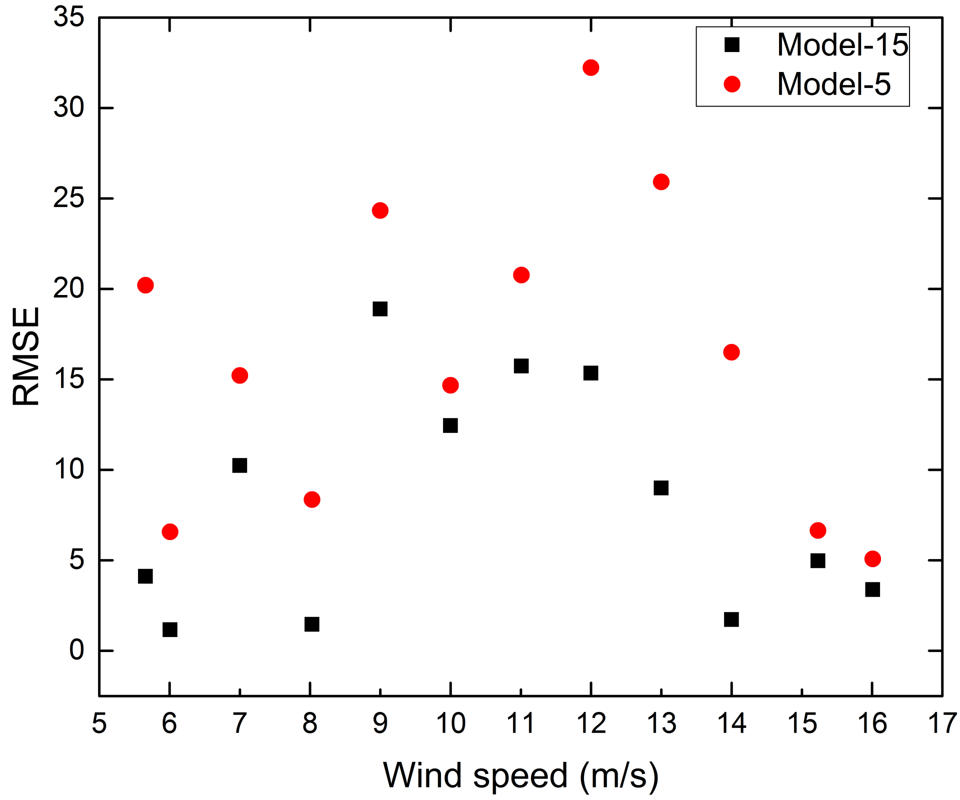


Figure 4.5: Comparison of RMSE for different modeling methods

it does not require any prior specification of suitable fitting functions. Whereas, RSM is restricted only for quadratic approximations (Moghaddam and Khajeh 2011).

4.9.1 Validation

The OSELM model which is proved to be the best for wind power modeling and simulation has been validated by taking 5% of the entire data which has not been used in training as well as test data set. The data collected during May 2017, from an identical turbine in the same wind farm has been used. The model is proved to be efficient with 97.68% prediction accuracy and RMSE of 5.12% on validation data. The corresponding results are shown in Figure 4.6 for 25 number of randomly selected data. Good

Table 4.16: Comparison of conventional model based on RSM with ANN model

Sl.No	Wind speed m/s	Actual power (kW)	Model- 5		OSELM Model	
			Power (kW)	RMSE	Power (kW)	RMSE
1	5.66	250.85	222.28	20.19	245.03	4.11
2	6.01	289.38	280.07	6.57	291.01	1.16
3	7	447.43	425.91	15.21	432.95	10.24
4	8.03	637.13	625.30	8.36	635.07	1.45
5	9	820.9	786.48	24.33	794.19	18.89
6	10	981.15	960.39	14.67	961.54	12.45
7	11.01	1201.06	1171.69	20.76	1178.81	15.73
8	12	1392.21	1346.63	32.23	1370.52	15.34
9	13	1473.73	1437.08	25.91	1461.00	9.00
10	14	1500.18	1476.85	16.49	1497.74	1.72
11	15.23	1487.85	1478.45	6.64	1494.87	4.97
12	16.11	1475.53	1468.35	5.07	1461.75	3.38

agreement between predicted and actual values can be observed.

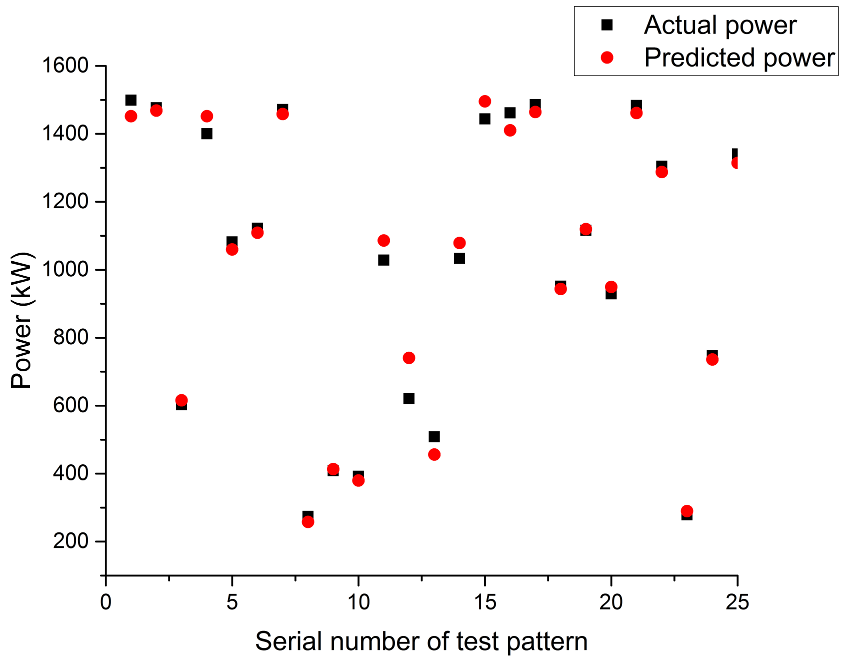


Figure 4.6: Comparison of actual and predicted power by OSELM Model (Validation data)

CHAPTER 5

OPTIMIZATION OF WIND POWER OUTPUT

This chapter provides the detailed explanation of three metaheuristic optimization algorithms namely Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC) and Cuckoo Search (CS). Two methods of objective function development namely Artificial Neural Network (ANN) and Response Surface Methodology (RSM) are discussed and the results are compared to derive a most suitable method to accurately frame the objective function of the problem under consideration. The practical and theoretical constraints are defined, followed by the results of each of the metaheuristic optimization algorithms.

In the present study the objective is to maximize the power output of a wind turbine. The set of uncontrollable parameters affecting the wind turbine power have been identified as wind speed, air density, rotor speed and wind direction and the only controllable parameter in maximizing the power is blade pitch angle.

5.1 METAHEURISTIC ALGORITHMS CONSIDERED

Three stochastic, population based metaheuristic algorithms namely Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC) and Cuckoo Search (CS) have been used in this study. These algorithms work on the concept of pattern matrix consisting of random candidate solutions. For instance, in each of these algorithms say PSO, ABC and CS, the pattern refers to a particle, a nectar source and a cuckoo nest respectively. The value of the objective function is evaluated for every candidate solution to derive the best solution out of the initial pattern matrix. Then, each of these algorithms use its own search mechanism to improve the solution obtained initially on an iterative basis.

A particle, a nectar source and a cuckoo nest corresponds to the controllable variable that is being optimized, which is blade pitch angle in the current study. The value of the objective function is wind power value.

5.1.1 Particle Swarm Optimization (PSO) algorithm

Particle Swarm Optimization (PSO) is a swarm based optimization algorithm developed by Kennedy and Eberhart based on social behavior of fish and bird in nature (Kennedy 2010). PSO optimizes the objective function by adjusting the trajectories of individual agents. These agents are called particles. In the search space, these particles approach the optima according to simple mathematical formulae for the particles position and velocity.

The first step in this algorithm is generating a group of particles (population) randomly. These particles are then set into motion. Each particle sets the movement, by adjusting its position based on its own best known position as well as the entire swarms best known position. By doing so, both local and global search mechanisms can be achieved effectively. At the end of each iteration (generation), all particles observe the fitness (objective function) value and move toward better positions. The particle in PSO refers to the candidate solution.

For the algorithm, the velocity (u) and position (s) of each particle i in iteration t+1 can be computed as in Equations 5.1 and 5.2.

$$u_i^{t+1} = w * u_i^t + C_1 rand1(pbest_i - S_i^t) + C_2 rand2(gbest - S_i^t) \tag{5.1}$$

$$S_i^{t+1} = S_i^t + u_i^{t+1} \tag{5.2}$$

where,

u_i^t = velocity of the particle at iteration t

u_i^{t+1} = velocity of the particle at iteration t+1.

s_i^t= position of the particle at iteration t.

s_i^{t+1}= position of the particle at iteration t+1.

$rand1$ and $rand2$= uniformly distributed random variable.

C_1 and C_2= acceleration factors.

pbest = local best position.

gbest= global best position.

The pseudo code of the algorithm is as follows:

For each particle

Initialize particle

End

Do

For each particle calculate fitness value

If the value is better than the best fitness value (pbest) in the history, set current value as the new pbest

End

Choose the particle with the best fitness value of all the particles as the gbest

For each particle

Calculate particle velocity according to Equation 5.1.

Update the position according to Equation 5.2

End

While maximum iterations or minimum error criteria is not attained (Kennedy 2010).

5.1.2 Artificial Bee Colony (ABC) optimization algorithm

This algorithm proposed by Karaboga in 2005, works based on the nectar searching behavior of bees(Karaboga 2005). The ABC algorithm targets the location of nectar source having the maximum amount of nectar. The nectar source refers to the candidate solution (blade pitch angle) and amount of nectar refers to the value of objective function (wind power) in this algorithm.

The process of searching for nectar in a bee hive takes place with the help of two kinds of bees namely employed and onlooker bees. Every employed bee will be dealing with only one nectar source. The employed bee randomly picks a nectar source, evaluates the nectar amount and also exploits for better food source in the neighborhood. Each of these employed bees share the information about the quality of food source they have found to be the best, with onlooker bees in the hive.

Based on the quality of the food source, on a probability basis, the onlooker bee selects the food source and does a search for better source in the neighborhood. If a food source can not be improved after the completion of search process by both employed and onlooker bees, the source is abandoned. The employed bee now picks up a new randomly discovered food source.

The algorithm is as follows:

Step 1. Initialize the nectar sources (solutions) randomly according to Equation 5.3

$$X_{ij} = X_j^{min} + (X_j^{max} - X_j^{min}) * rand \tag{5.3}$$

where i= $1, 2, \ldots, SN$, j=$1, 2, \ldots, m$. SN, m are the number of food sources and optimization parameters respectively.

Step 2. Searching for the new food sources in the neighborhood to see if there are any better source. This search for new food source is defined by Equation 5.4. The food sources v_{ij} and X_{ij} are compared and the better source is saved and the worst source is discarded.

$$v_{ij} = X_{ij} + (X_{ij} - X_{jk}) * \phi_{ij} \tag{5.4}$$

where, $\phi_{ij}$is the uniformly distributed random number in the range [-1,1], j is the random integer in the range [1,m] , k$\in \{1, 2, \ldots, SN\}$.

Step 3. Calculating the probability values related to the nectar amount in the source to choose the best food source. The probability values can be obtained using the Equation 5.5.

$$P_i = \frac{Fitness_i}{\Sigma_{j=1}^{SN} Fitness_j} \tag{5.5}$$

where, $Fitness = \begin{cases} \frac{1}{1+f_i} & f_i \geq 0 \\ 1 + |f_i| & f_i < 0 \end{cases}$

f_i is the cost value for the solution v_{ij}.

Step 4. If a food source cannot be improved in Steps 3 and 4 for a predetermined number of trials (Abandonment Limit), it is abandoned in each cycle and is replaced by a new randomly discovered food source.

These steps are repeated until the stopping criteria is reached (Karaboga 2005). The algorithmic control parameter namely Abandonment Limit Parameter is set 0.6*m*SN based on trial and error.

5.1.3 Cuckoo Search (CS) optimization algorithm

Yang and Deb introduced a metaheuristic algorithm named Cuckoo Search in 2009 (Yang and Deb 2009). This algorithm is inspired by the parasitic breeding behavior of Cuckoo bird. This algorithm uses $Lévy$ flight search mechanism to explore the search space in a more efficient manner. $Lévy$ flight is a random walk mechanism, where the steps are made in isotropic random directions with the step length following $Lévy$ distribution (Yang 2010).

This algorithm works on the generalized rules listed below:

1. At a time, every Cuckoo bird lays only one egg and dumps it in a nest that is chosen on a random basis.

2. The nests that are carried over to next generation are the ones with high quality eggs.

3. There will be fixed number of host nests. The host bird discovers the presence of alien egg with the probability of P_a which varies from 0 to 1. Depending on the probability, the host bird either discards the egg or abandons the nest and builds altogether a new one.

The algorithm is as follows:

Step 1. Initialization: The host nest location is initialized by using Equation 5.6

$$nest_{i,j} = Lb_{min,j} + (Ub_{max,j} - Lb_{min,j}) * rand_1 \tag{5.6}$$

where, $nest_{i,j}$ denotes the i^{th} host nest in the population,

$nest_{i,j} = \{nest_{i,1}, nest_{i,2} \ldots, nest_{i,n}\}$ there are m number of randomly generated n-dimensional real valued vectors.

where, $i = 1, 2, \ldots, m, j = 1, 2, \ldots, n$. $Lb_{min,j}$ and $Ub_{max,j}$ are the lower and upper boundary values of the dimension j and $rand_1$ is a random number within the range of (0, 1).

Step 2. Evaluate the fitness function to find $Xbest_i$ and $Gbest$

Step 3. Generation of new nests by $Lévy$ flights based on the previous best nest.

For each of the nest, the new solution can be calculated by using Equation 5.7

$$X_i^{new} = Xbest_i + \alpha * rand_2 * \Delta X_i^{new} \tag{5.7}$$

where, $\alpha > 0$ and $rand_2$ is a random number, ΔX_i^{new} can be calculated using Equation 5.8

$$\Delta X_i^{new} = v * \frac{\sigma_x(\beta)}{\sigma_y(\beta)} * (Xbest_i - Gbest) \tag{5.8}$$

where, $v = \frac{rand\ x}{|rand\ y|^{\frac{1}{\beta}}}$

where, $rand\ x$ and $rand\ y$ are the random numbers with the standard deviation $\sigma_x(\beta)$ and $\sigma_y(\beta)$ given by Equations 5.9 and 5.10 respectively

$$\sigma_x\left(\beta\right) = \left[\frac{\Gamma\left(1+\beta\right) * \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) * \beta * 2^{\frac{\beta-1}{2}}}\right]^{\frac{1}{\beta}} \tag{5.9}$$

$$\sigma_y\left(\beta\right) = 1 \tag{5.10}$$

Step 4. Alien egg discovery and randomization:

The new solution is created to the problem based on the discovery action of the alien egg in the nest of a host bird, with the probability P_a by using Equation 5.11.

$$X_i^{dis} = Xbest_i + K * \mathit{\Delta} X_i^{dis} \tag{5.11}$$

where K is the updated coefficient based on the probability that the host bird discovers in its nest the alien egg

$$K = \begin{cases} 1 \;\; if\; rand_3 < P_a \\ 0 \qquad otherwise \end{cases}$$

The value ΔX_i^{dis} is calculated using Equation 5.12

$$\Delta X_i^{dis} = rand_3 * [randp_1\,(Xbest_i) - randp_2\,(Xbest_i)] \tag{5.12}$$

where, $rand_3$ is the random number in the range [0,1] , $randp_1$ $(Xbest_i)$ and $randp_2(Xbest_i)$ are the random perturbations for position of nests in $Xbest_i$.

Step 5. Compare the fitness value corresponding to old and new nest and keep the nest owning best fitness value. Repeat the procedure till the stopping criteria is reached.

In this study, the stopping criteria is set as 100 iterations. The value of n, β and P_a is set to be 10, 1.5 and 0.25 respectively.

5.2 OBJECTIVE FUNCTION DEVELOPMENT

The problem definition is an important step in optimization. This involves proper framing of objective function and constraints. Any mistakes in this step certainly leads to wrong solution. An objective function should properly describe the relationship between controllable and uncontrollable input parameters with that of the output mathematically.

In this study, to investigate the most appropriate method to develop the objective function that is power output of a wind turbine as a function of wind speed, air density, rotor speed, wind direction and blade pitch angle, two approaches have been tried.

Approach-1: ANN based objective function

Approach-2: Response Surface Method (RSM) based objective function

5.2.1 ANN based objective function

ANN technique has already been proved to be efficient in mapping the input output relationships in modeling studies in this research. Hence it has been used to develop the objective function for the optimization studies. A Radial Basis Function (RBF) neural network model with centers selected using conditional Fuzzy C-means clustering algorithm and ELM learning has been used in this study. The details of this model has already been discussed in Section 4.7.2.

An ANN model develops a mathematical relationship between the input and output variables during training with the help of weights and biases. Such a relationship developed by the above said ANN model has been used as the objective function in this study.

5.2.2 Response Surface Method (RSM) based objective function

Response Surface Method (RSM) is a collection of mathematical and statistical techniques for model building. This technique is successfully applied to both linear and nonlinear problems.

The mathematical relationship between independent input variables and the dependent output variable have been obtained in terms of a second order model of the form $Y = ax^2 + bx + c$, where, Y is the predicted response; x is the input variable that influences the response variable. The regression model development has been performed using MINITAB-17 (Minitab 2014). 99% confidence level has been set and backward elimination method has been used in this work. The R^2 value obtained for the model is 0.9954. The resulting relationship is given by Equation 5.13.

$$
\begin{aligned}
Y = 3356 + 310.8X_1 + 126.2X_2 + 1.34X_3 + 738.4X_4 866X_5 - 0.8122X_3^2 \\
+35.02X_5^2 - 7.7526X_1X_2 - 0.882X_1X_3 - 8.99X_1X_5 - 2.434X_2X_5
\end{aligned}
\tag{5.13}
$$

where, X_1, X_2, X_3, X_4, X_5 are Wind speed, blade pitch angle, wind direction, density and rotor speeds respectively and Y is the wind power output.

5.3 CONSTRAINTS

While designing the optimization problem, it is important to specify the constraints (if any) to the problem. The power that a wind turbine can produce has practical and theoretical constraints. The practical constraint being the maximum power generation capacity of the turbine. The turbine under consideration is designed to produce maximum power of 1600 kW.

According to Betz law, the theoretical maximum power that a turbine can generate cannot exceed 59% of the kinetic energy in the wind (Manwell *et al.* 2010). The wind power is given in Equation 5.14.

$$P = \frac{1}{2}\rho\pi R^2 v^3 \tag{5.14}$$

where, P is the power captured by the rotor of a wind turbine, kW

ρ is the air density, kg/m^3

R is the radius of the rotor, m

v is the wind speed, m/s.

The swept area of the turbine under consideration is 5281 m^2 and the average air density in the area is 1.15 kg/m^3. Thus, the theoretical power of the turbine should be smaller than 3.036 v^3. Hence, by considering both of these constraints, the power output of the turbine under consideration should be a minimum of 1600 kW and 3.036 v^3 for a given wind speed.

5.4 RESULTS AND DISCUSSION

The objective function has been developed by using 2522 (85%) data instances out of 2966 total data. The remaining 444 test patterns are optimized to get the maximum power output. Each data point in the test pattern is optimized for optimal control settings (blade pitch angle) in order to maximize the power output from the wind turbine without changing the values of the non-controllable variables, wind speed, density wind direction and rotor speed. Based on analysis of historical data, the range of blade pitch angle is limited to [-2.616, 19.733].

The common control parameters for the optimization algorithms investigated in this study namely population size and number of iterations are set to same value. The stopping criteria used for the optimization model is maximum number of iterations which has been set as 100. The population size is fixed to be 10.

Root Mean Square Error (RMSE) as defined in Equation 5.15 has been used as the metric to evaluate Approach-1 and Approach-2.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(P_{Optimized}\left(i\right) - P_{Actual}\left(i\right)\right)^2} \tag{5.15}$$

where, $P_{Optimized}$ and P_{Actual} are optimized and actual wind power output in kW respectively and N is the total number of test patterns.

Two metrics, Mean PG and Mean Relative PG as given by Equation 5.16 and 5.17 which indicates how much kW of the power will be gained on an average for every hour due to optimization, have been used to evaluate the performance of optimization algorithms.

$$Mean\ PG = \frac{1}{N_{Test}}\Sigma_{t\epsilon Test}P_{optimized}(t) - P_{actual}(t) \tag{5.16}$$

Table 5.1: Error summary

Objective function	RMSE
Approach-1	66.89
Approach-2	190.97

$$Mean\,Relative\,PG = \frac{1}{N_{Test}}\Sigma_{t\epsilon Test}\frac{P_{optimized}(t)-P_{actual}(t)}{P_{actual}(t)} * 100 \tag{5.17}$$

(Kusiak *et al.* 2010*b*)

where,

$P_{optimized}$ =Optimal power in kW

P_{actual}= Actual power in kW

N_{Test}= Total number of test patterns

Test= Set containing all test patterns

5.4.1 Particle Swarm Optimization (PSO)

In this study, PSO has been used to evaluate the methods for developing the objective function. The best objective function thus found is considered in further studies using ABC and CS algorithms. The algorithmic control parameters of PSO namely Acceleration constants C_1 and C_2 are set as 2 on trial and error basis.

The power optimized by Approach-1 and Approach-2 for 50 randomly selected test patterns are shown in Figure 5.1 and 5.2 , whereas Figure 5.3 and 5.4 shows the optimized blade pitch angle values obtained by using PSO with these two approaches of objective function development.

From Table 5.1 it can be noted that, RMSE value is higher for Approach-2, which used objective function based on RSM. It shows that, this approach fails to map the input-output relations properly, thus resulting in higher error. Approach-1, which is

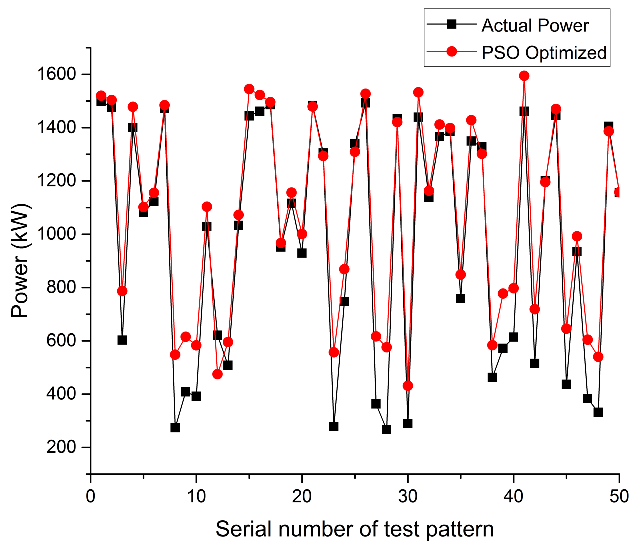


Figure 5.1: Comparison of optimal power obtained using Approach-1 with actual power

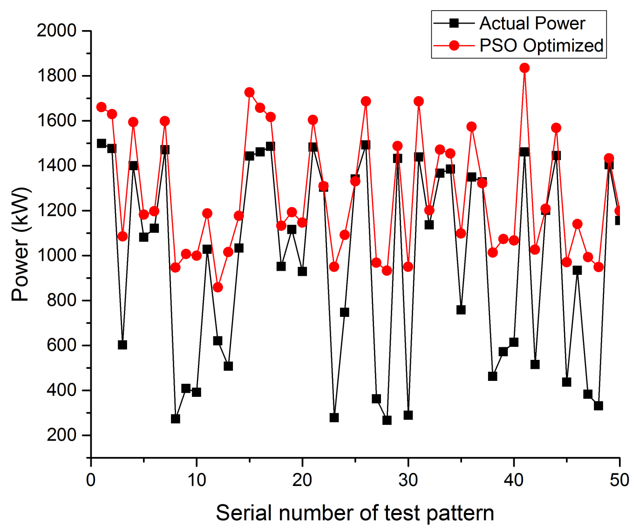


Figure 5.2: Comparison of optimal power obtained using Approach-2 with actual power

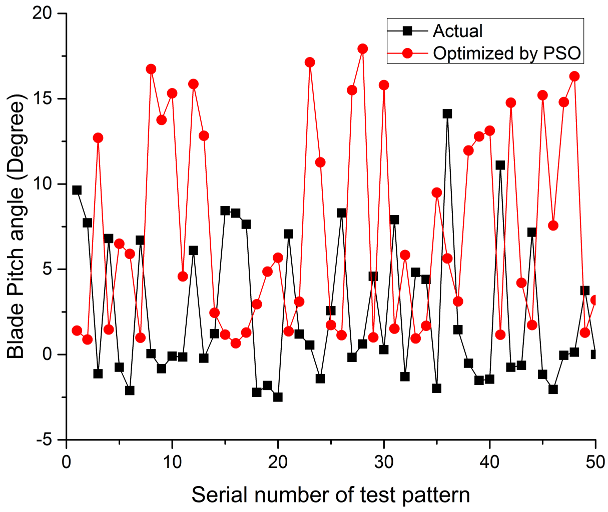


Figure 5.3: Comparison of actual and optimal blade pitch angle obtained using Approach-1

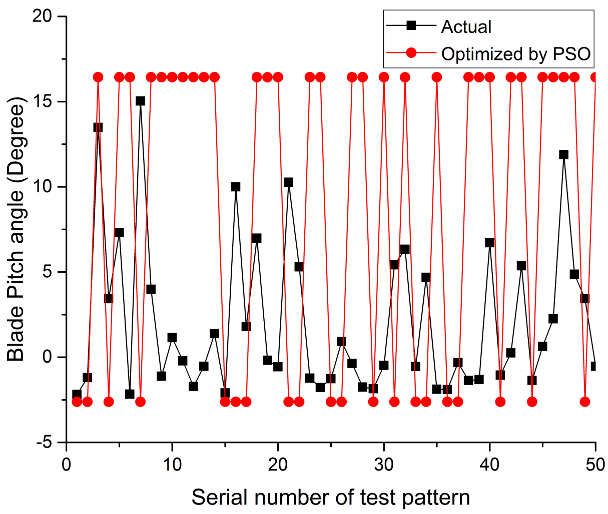


Figure 5.4: Comparison of actual and optimal blade pitch angle obtained using Approach-2

based on ANN, has resulted in power values quite closer to the actual values as depicted in Figure 5.1. This establishes the ability of ANN to capture the input-output relations more efficiently.

Though the objective of the present work is to maximize the power of the wind turbine, when it comes to the evaluation of objective functions, the accuracy of mapping the relationship between parameters is a matter of concern. Thus, from the results it can be found that RSM technique is less efficient compared to ANN technique in building appropriate relationship between the input and output parameters .

The blade pitch angle values obtained from Approach-1 and Approach-2 are shown in Figure 5.3 and 5.4 respectively. From Figure 5.4, it can be observed that the optimal blade pitch angle obtained from Approach-2 shoots to either lowest or highest without any in between values. This shows that the objective function is one of the major factor affecting the optimization performance of the algorithm. Hence Approach-2 is found to be less efficient in optimizing the blade pitch angle compared to Approach-1.

For further study, the objective function developed using ANN has been considered. The variation of objective function value with iterations for one of the test pattern is shown in Figure 5.5. It can be seen that PSO reached global maximum in 33^{rd} iteration.The Mean PG and Mean Relative PG obtained from PSO algorithm are 87.637 kW and 17.316 % respectively.

5.4.2 Artificial Bee Colony (ABC)

The variation of objective function value, that is power in this study, with iterations for ABC algorithm is shown in Figure 5.6. It can be seen that ABC reached global maximum in 33^{rd} iteration. Figure 5.8 and 5.9 show the comparison of actual , optimized power and blade pitch angles obtained from ABC algorithm for 50 randomly selected test patterns. The Mean PG and Mean Relative PG obtained from ABC algorithm are 87.013 kW and 17.183 % respectively.

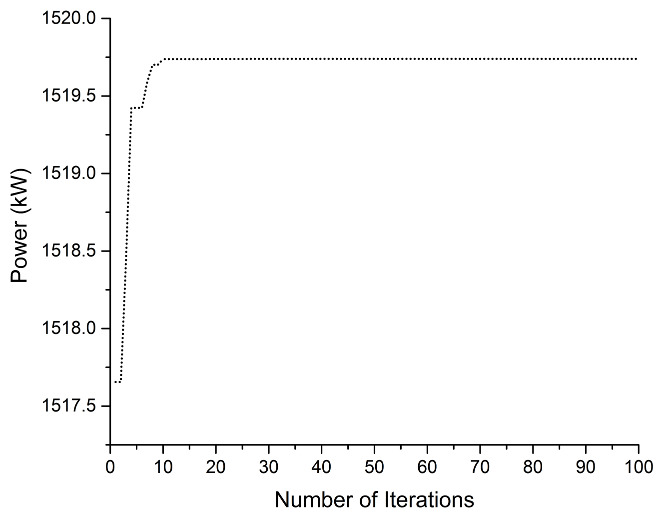


Figure 5.5: Variation of objective function value with iterations for PSO algorithm

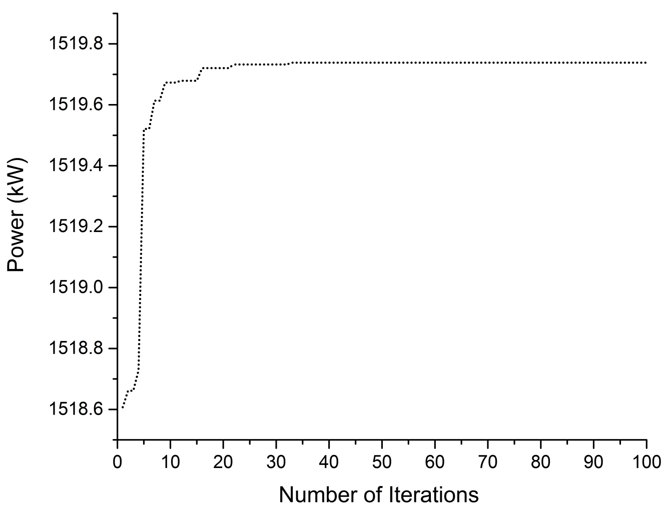


Figure 5.6: Variation of objective function value with iterations for ABC algorithm

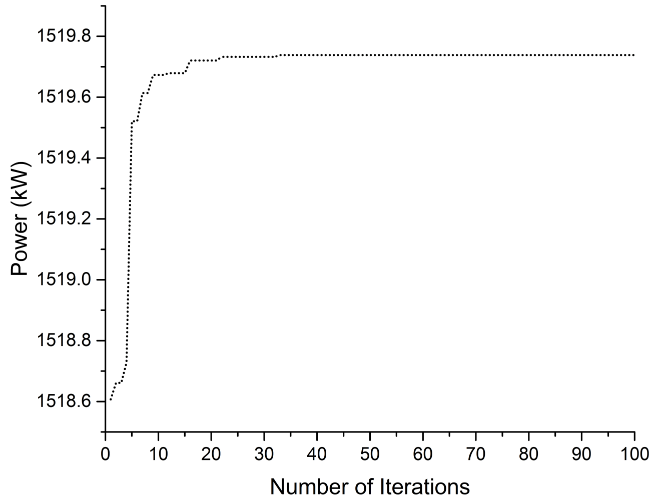


Figure 5.7: Variation of objective function value with iterations for CS algorithm

5.4.3 Cuckoo Search (CS)

The variation of objective function value with iterations for CS algorithm is shown in Figure 5.7. The CS algorithm converges faster, because it uses $Lévy$ flight search while generating new solution. It can be seen that CS reached global maximum for 26^{th} iteration. Figures 5.8 and 5.9 shows the comparison of actual and optimized power and blade pitch angles obtained from CS algorithm for 50 randomly selected test patterns. From Figure 5.8, it can be observed that, the actual and optimized power values are close for higher power range above 1000 kW, but the optimized power values are higher than actual for lower range of power. It is observed that the Mean PG and Mean Relative PG obtained from CS algorithm are 87.668 kW and 17.329 % respectively.

5.4.4 Comparison of optimization algorithms

The comparison of performance of three optimization algorithms used in the present study namely PSO, ABC, CS has been done in this section. Power gain in terms of Mean PG and Mean Relative PG of all the three algorithms are presented in Table 5.2.

Table 5.2: Power Gain Summary

Optimization algorithm	Mean PG (kW)	Mean Relative PG (%)
PSO	87.637	17.316
ABC	87.013	17.183
CS	87.668	17.329

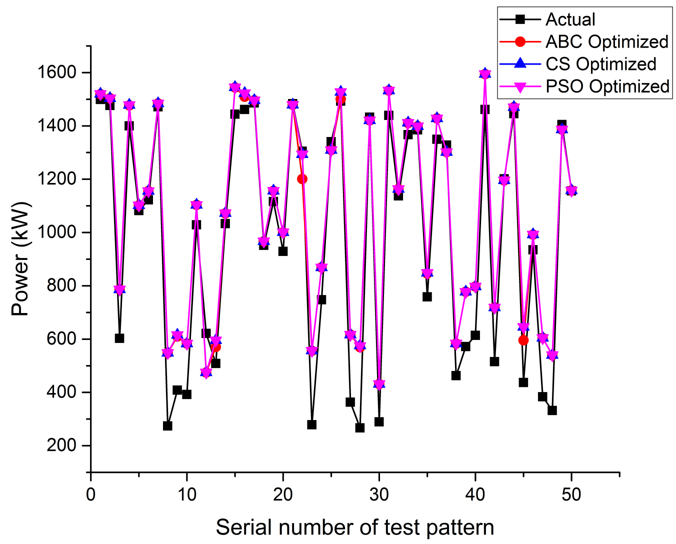


Figure 5.8: Comparison of optimized power obtained using PSO, ABC and CS algorithm with Actual

From the table it can be noted that the values of Mean PG and Mean Relative PG are higher for CS and low for ABC.

From Figure 5.5, 5.6 and 5.7 which shows the variation of fitness value with iterations for PSO, ABC and CS respectively it can be seen that, CS algorithm converges faster in only 26 iterations compared to 33 iterations for the other two algorithms. From Figure 5.8 and 5.9 it can be seen that the optimum values of power and blade pitch angle obtained from CS and PSO algorithms are close to each other. But the optimized power values obtained from ABC are lower at few points, such as 22^{nd} and 45^{th} points out of the 50 randomly selected values plotted.

The performance of CS is observed to be better than PSO and ABC in the current study. This is due to the use of $Lévy$ flight search mechanism which is more efficient in

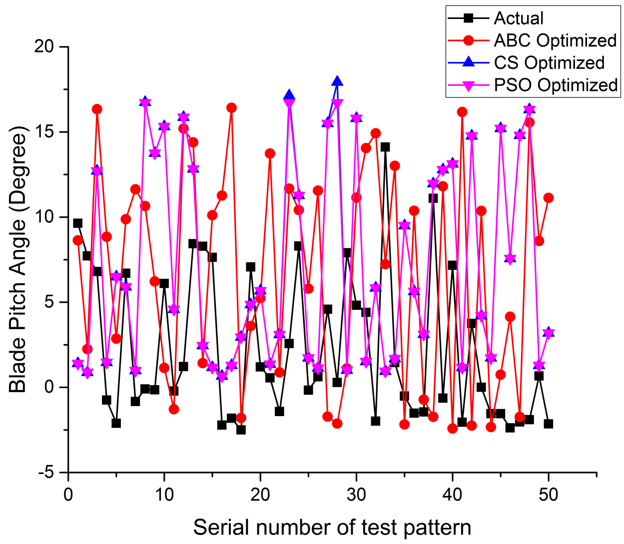


Figure 5.9: Comparison of optimized blade pitch angle obtained using PSO, ABC and CS algorithm with Actual

exploring the search space while generating the new solutions. The chances of getting trapped in local minima are less in this method. Further, new solutions are generated around the best solution obtained so far by the *Lévy* flight in CS. This speeds up the local search process in the algorithm (Yang and Deb 2009). However, the decision mechanism of ABC algorithm that assists in identifying the potential search space requiring a more detailed survey in discovering new nectar source is powerful, hence it has resulted in a comparable performance with PSO and CS respectively.

This study has considered two approaches for developing the objective functions based on ANN and RSM techniques. The ANN based objective function resulted in good mapping of input output relationship, thus proving to be best amongst the two. Three metaheuristic algorithms namely PSO, ABC and CS were investigated to optimize the blade pitch angle, the only controllable parameter out of the five input parameters considered. PSO and ABC converged in 33 iterations, while the CS converged in 26 iterations hence found to be faster.

CHAPTER 6

ANN BASED WIND SPEED FORECASTING

This chapter includes the information of two data denoising techniques namely Emperical Mode Decomposition (EMD) and Optimized Variational Mode Decomposition (OVMD). The methodology of proposed hybrid ANN model is given in detail followed by the results of the proposed model and benchmark models.

6.1 DATA PRE-PROCESSING

The wind speed data obtained from Supervisory Control and Data Acquisition (SCADA) system of a turbine may contain noise due to various reasons like anemometer error depending on the angle of attack (Nakai and Shimoyama 2012), non-linearity of rotation of anemometer, over estimation at turbulent gusty wind conditions etc. (Kaganov and Yaglom 1976). All these reasons may lead to missing values, out of range values, inconsistency and redundancy in the data. Such noise in wind speed data affects the performance of the forecasting models greatly. Thus, denoising is an essential part of wind speed data pre-processing.

Various researchers have used traditional filtering to remove the ineffective frequency components, so that the wind speed signal becomes smoother and thus more predictable (Riahy and Abedi 2008). Traditional filtering approaches such as Butterworth low-pass filter has been applied for filtering the wind speed data. The wavelet denoising methods have been widely used for variety of applications and are proved to be better than the traditional filtering approaches (Lin *et al.* 2013). But the performance of the Wavelet Transform (WT) techniques depends greatly on the selection of the mother wavelet and decomposition levels. Emperical Mode Decomposition (EMD) are proved to be better than WT in some of the literature (Labate *et al.* 2013). Though EMD have been proved to be better than WT and used in wide range of applications including wind speed forecasting (Liu *et al.* 2015*a*),(Liu *et al.* 2012*a*),(Ren *et al.* 2015),(Zhang *et al.* 2016*a*), it lacks theoretical background and suffers from mode mixing problem, is sensitive to noise and sampling (Dragomiretskiy and Zosso 2014). Variational Mode Decomposition (VMD) and its variants are emerging as popular data pre-processing methods in wind speed forecasting, since it overcomes the mode mixing problem of EMD.

6.1.1 Emperical Mode Decomposition (EMD)

N. E. Huang initially proposed EMD, a data driven algorithm in 1998 (Huang *et al.* 1998). This algorithm is useful in handling the non-stationary and non-linear signals. In EMD ,the complicated signal is decomposed into finite number of locally narrow band components. These are termed as Intrinsic Mode Functions (IMF).

The following are the two conditions IMFs have to satisfy:

1) The number of zero crossings and the number of extrema must either differ by one or be equal.

2) Mean value of the envelopes that are defined by local minima and local maxima, at any point must be zero.

The signal is decomposed in EMD according to the following steps:

Step 1. For the given signal $y(t)$, identify the local extrema. Using cubic spline, join the local maxima and local minima to get the upper and lower envelopes respectively.

Step 2. Calculate the mean value $m(t)$ of the two envelopes. Obtain the detailed component by subtracting $m(t)$ from $y(t)$ as given in Equation 6.1.

$$h(t) = y(t) - m(t) \tag{6.1}$$

Step 3. Check whether $h(t)$ is an IMF.

If yes then set it as the value of first IMF, $c_1(t) = h(t)$, calculate the residue $r_1 = y(t) - c_1(t)$ and replace $y(t)$ with r_1. Keep repeating the steps n times to obtain the IMF values until residue becomes smaller than the predefined threshold value or becomes a monotonic function. If no, take value of $h(t)$ as new $y(t)$ and repeat Step 1 and Step 2 until criteria 6.2 is satisfied.

$$\Sigma_{t=1}^{T} = \frac{[h_{j-1}(t) - h_j(t)]^2}{[h_{j-1}(t)]^2} \leq \delta \tag{6.2}$$

where, j is the number of iterations and T denotes the length of the signal. The value of δ is set in the range of 0.2-0.3.

The original signal can be reconstructed by summing up different IMF values $c_i(t)$ and the final residue r_n as given in Equation 6.3

$$y(t) = \Sigma_{i=1}^{n} c_i(t) + r_n \tag{6.3}$$

where $i = 1, 2, \ldots, n$.

6.1.2 Optimized Variational Mode Decomposition (OVMD)

Variational Mode Decomposition (VMD), a signal processing method successfully used in many applications, is proposed by Dragomiretskiy and Zosso in 2014 (Dragomiretskiy and Zosso 2014). In VMD, the one dimension input signal is decomposed into number of sub-signals termed as modes. These modes $u_k = (k = 1, 2, \ldots, K)$ show specific sparsity properties in the process of reproducing the input. It is assumed that every variational mode u_k is compact around a center pulsation (frequency) ω_k which is determined during the process of decomposition.

The steps used to estimate the band width of every mode are as follows:

1. Use Hilbert transform to compute the analytic signal of every mode u_k, in order to get the unilateral frequency spectrum.

2. For each mode, transfer the subseries frequency spectrum to baseband (narrow frequency) using an exponential tuned to the respective estimated center frequency.

3. The H^1 Gaussian smoothness of the demodulated signal is used to calculate the bandwidth of every mode.

Hence the constrained, variational problem can be given as in Equation 6.4

$$\underset{\{u_k\},\{\omega_k\}}{min}\{\Sigma_K||\delta_t[(\delta(t)+\frac{j}{\pi t})\otimes u_k(t)]e^{-j\omega_k t}||_2^2\} \tag{6.4}$$

subjected to $\Sigma_{k=1}^{K}u_k = f$

where , f is the original signal and u_k is the k^{th} component of the signal, K is the number of modes, δ is the Dirac distribution and $\otimes$ is the convolution operator. By considering penalty term and Lagrangian multipliers λ, the above constrained problem can be converted into the unconstrained one.

$$L(\{u_k\},\{\omega_k\},\lambda) = \alpha\Sigma_{k=1}^{K}||\delta_t[(\delta(t)+\frac{j}{\pi t})\otimes u_k(t)]e^{-j\omega_k t}||_2^2 + ||f(t)-\Sigma_{k=1}^{K}u_k(t)||_2^2 + \langle\lambda(t), f(t)-\Sigma_{k=1}^{K}u_k(t)\rangle \tag{6.5}$$

where α is the balancing parameter.

Alternate Method of Multipliers (ADMM) is used to solve the above problem presented in Equation 6.5 by finding the saddle point of the Lagrangian argument L. The solution for u_k and ω_k are presented as follows:

$$\hat{u}_k^{n+1} = \frac{\hat{f}(\omega) - \Sigma_{i \neq k} \hat{u}_i(\omega) + \frac{\hat{\lambda}(\omega)}{2}}{1 + 2\alpha(\omega - \omega_k)^2} \tag{6.6}$$

$$\omega_k^{n+1} = \frac{\int_0^\infty \omega |\hat{u}_k(\omega)|^2 d\omega}{\int_0^\infty |\hat{u}_k(\omega)|^2 d\omega} \tag{6.7}$$

where $\hat{u}_k^{n+1}$,$\hat{u}_i(\omega)$, $\hat{f}(\omega)$,$\hat{\lambda}(\omega)$ represent the Fourier transform of u_k^{n+1}, $u_i(\omega)$, $f(\omega)$, and $\lambda(\omega)$ respectively. n is the number of iterations.

The main steps in VMD are as given below:

Step 1. Initialize $\{\hat{u}_k^1\}, \{\omega_k^1\}$ and $\hat{\lambda}^1$

Step 2. Obtain $\hat{u}_k^{n+1}$ and ω_k^{n+1} for each subseries u_k using Equation 6.6 and 6.7 respectively

Step 3. λ is updated by using Equation 6.8.

$$\hat{\lambda}^{n+1}(\omega) = \hat{\lambda}^n(\omega) + \tau(\hat{f}(\omega)) + \hat{u}_k^{n+1}(\omega) \tag{6.8}$$

where, τ is the update parameter .

Step 4. Check the stopping criterion

$\sum_{k=1}^{K} \frac{||\hat{u}_k^{n+1} - \hat{u}_k^n||_2^2}{||\hat{u}_k^n||_2^2} < \epsilon$, stop if it is satisfied; else go to Step 2. and the process is repeated until the stopping criteria is reached.

In VMD, the number of modes the signal has to be decomposed need to be predefined. It has been observed from the previous studies that the number of modes greatly affects the efficiency of VMD (Abdoos 2016), (Dragomiretskiy and Zosso 2014), (Zhang *et al.* 2016*b*).

Dragomiretskiy discussed in (Dragomiretskiy and Zosso 2014) that too small and too large number of modes lead to under-segmentation and over-segmentation respectively. In under-segmentation, the mode either gets shared with the neighboring ones or gets disappeared. In case of over-segmentation, some modes may capture noise or two or more modes share the signal resulting in mode duplication.

Hence to overcome these shortcomings of VMD, Chu Zhang et al (Civicioglu and Besdok 2013) proposed OVMD to search the optimal values of, number of modes and update parameter τ of VMD. To find the optimum value of number of modes, it was proposed to calculate and analyze the center frequencies of the decomposed modes. If the center frequencies are same for successive number of modes k and k+1, then k is the optimum value. The optimum value of τ is decided based on the Residual Evaluation Index (REI) as given in Equation 6.9.

$$REI = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|\sum_{j=1}^{K} u_k\,(i) - f(i)|} \tag{6.9}$$

where N is the length of the time series, $u_k(i)$ is the value of k^{th} decomposed subseries and $f(i)$ is the observed wind speed at i^{th} data point.

The value of τ resulting in lowest REI is considered to be the optimal one.

6.2 ANN MODELS FOR WIND SPEED FORECASTING

It is observed from the literature that MLP and ELM are the widely used ANN techniques for WSF (Zhang *et al.* 2017). ELM is a batch learning algorithm that finds vast application due to several advantages. To list a few, it is a fast learning algorithm with better generalization performance, less simulation parameters and avoids the difficulties of local minima.

The use of online learning algorithms can be beneficial because, the wind speed data is online in nature. An Online Sequential ELM (OSELM) algorithm introduced by Liang et al. in 2006 is capable of handling data arriving 1-by-1 or chunk-by-chunk

(Liang *et al.* 2006).

Hybridizing ANN models by combining data pre-processing technique and optimization algorithm enhances its ability to model nonlinear and non stationary characteristics of the wind speed data. Hence a hybrid ANN model that considers all these factors has been proposed in this work for short term wind speed forecasting and the performance is compared with some standard benchmark models that are extensively used in the literature.

The novelty of the present hybrid ANN model is as follows:

- Optimizing the chunk size, weights and biases of OSELM, in addition to number of hidden nodes by using CS algorithm.
- Exploiting CS algorithm for selection of input features
- Comparing the performance of CS algorithm and Partial Auto Correlation Function (PACF), a linear method most widely used for feature selection.

6.2.1 Proposed OVMD-CS-OSELM model

The proposed OVMD-CS-OSELM model works on the methodology presented in Figure 6.1. In this model, OVMD is used as a data pre-processing technique to decompose the wind speed series data into number of modes. The pre-processing of data using OVMD helps in reducing the complex abrupt changes in the wind speed. Fixing of two parameters of OVMD namely update parameter and number of modes are done based on REI values and center frequencies of the decomposed modes.

Once the data is decomposed, optimum number of input parameters are selected by Cuckoo Search (CS) optimization algorithm. Based on this number, the input matrix is prepared for every mode. After this step, the data is fed to CS-OSELM algorithm explained below for forecasting.

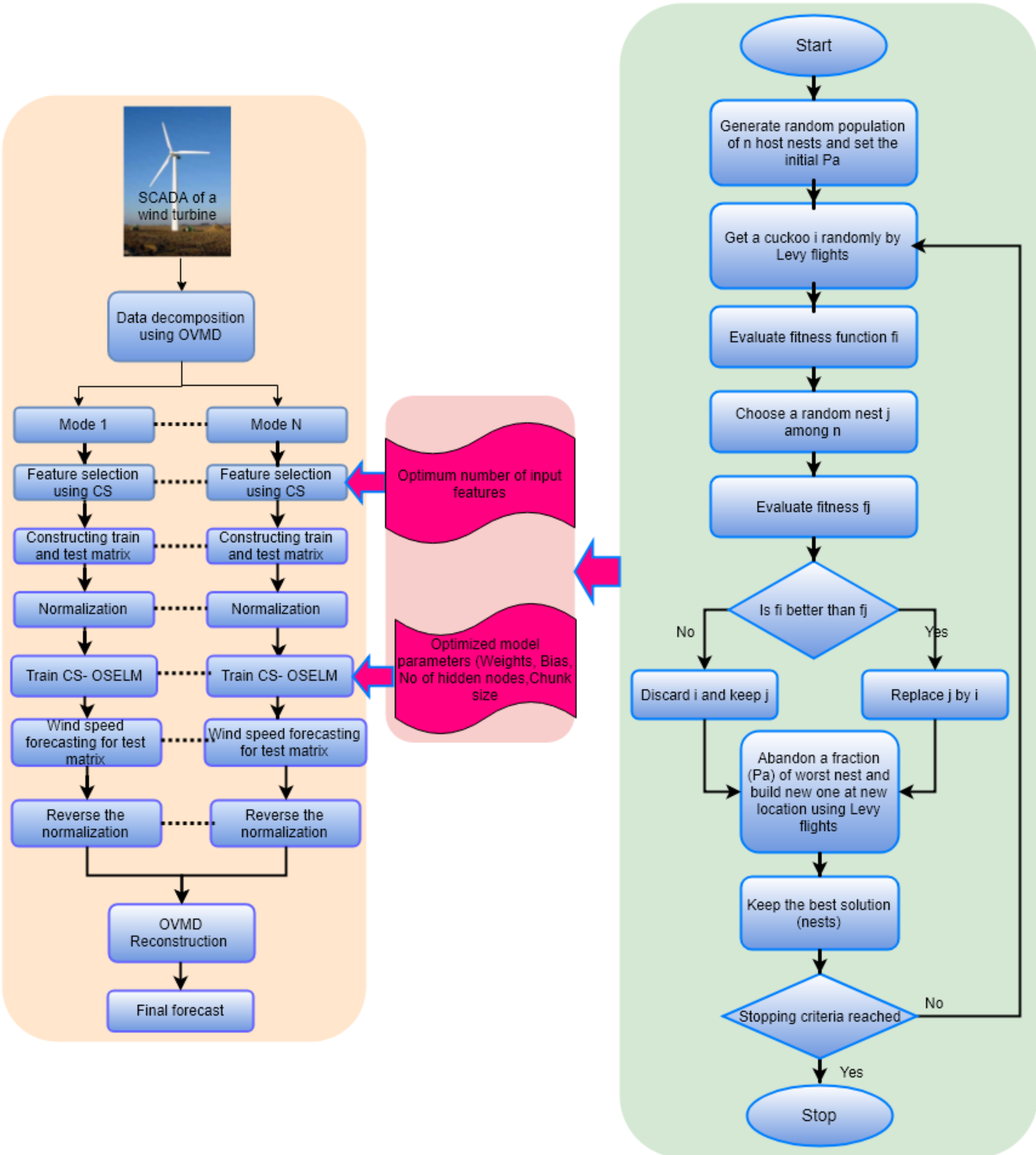


Figure 6.1: Framework of proposed OVMD-CS-OSELM WSF model

In CS-OSELM, CS is used to optimize various parameters which can be listed as:

i) Input feature selection (inp)

ii) Model parameters. The main model parameters affecting the performance of OSELM are number of hidden nodes (hid), chunk size (chk), weights (wt) and bias (bs).

Hence, in total there are five parameters (inp, hid, chk, wt, bs) which are being optimized in the proposed model. The objective function is to minimize the Root Mean Square Error (RMSE) between forecasted and observed values of wind speed, $\tilde{y}_i$ and y_i respectively as given in Equation 6.10.

$$fitness = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\tilde{y}_i - y_i\right)^2} \tag{6.10}$$

where N is the total number of training samples.

The steps for implementing CS-OSELM algorithm are given below:

Step 1: Normalize the data between [0,1], fix the parameters of CS such as population size n, number of iterations K , the distribution factor β, the probability P_a and the initial block of data N0 of OSELM. Set the upper and lower bound values of the parameters to be optimized and ensure that upper bound of $hid \leq$ N0.

Step 2: Randomly generate n initial populations ns_{ij} according to Equation 5.6 of CS algorithm, for all the five parameters of the model being optimized. where i=1,2,... n, j=1,2,... K.

Step 3: Prepare the dataset according to inp initialized by CS as shown in Figure 6.2. where, beginning with the first data D_1, inp number of data in series will be considered as the inputs, while the $(inp + 1)^{th}$ data is considered as the output. At the time of generating the second input-output set, the first data D_1 will be left out and the process will begin from the second data D_2. This process is repeated until the last data is reached. The total dataset is then split into training (85%) and test (15%) set.

Step 4: Use hid, chk, wt, bs values initialized by CS to obtain the training output from OSELM model, and evaluate the fitness values f_i, i=1,2,...,n, according to Equation 6.10, to determine the local best $Xbest$ and global best $Gbest$ among the local best. Set $Iter$ =1.

Step 5: Create new nests by $Lévy$ flights and calculate the new solution for every newly generated nest using Equation 5.7.

Step 6: Compare the old and new solutions and replace a fraction P_a of worse nests with the concept of alien egg discovery using Equation 5.11.

Step 7: Obtain the output of OSELM model for the new solution and compute the fitness function. Retain the best solutions.

Step 8: If $Iter\ <$ K, $Iter =\ Iter + 1$ and go to Step 5 and repeat the steps.

Else go to Step 9.

Step 9: Test the model using the optimal values of inp, hid, chk, wt, bs. De-normalize the output.

Finally the forecasting results from every mode is aggregated to obtain the final forecasted wind speed.

6.2.2 Benchmark models used for comparison

To evaluate the proposed model, the models given in Table 6.1, available and used in the literature for similar applications have been considered as benchmark models. Selection of suitable number of input features for all these models have been done using CS algorithm. Persistence model is a basic model used for wind forecasting to evaluate any newly developed model in the literature. An ELM based ANN model is selected as the benchmark model due to the reason that these models are widely used in these

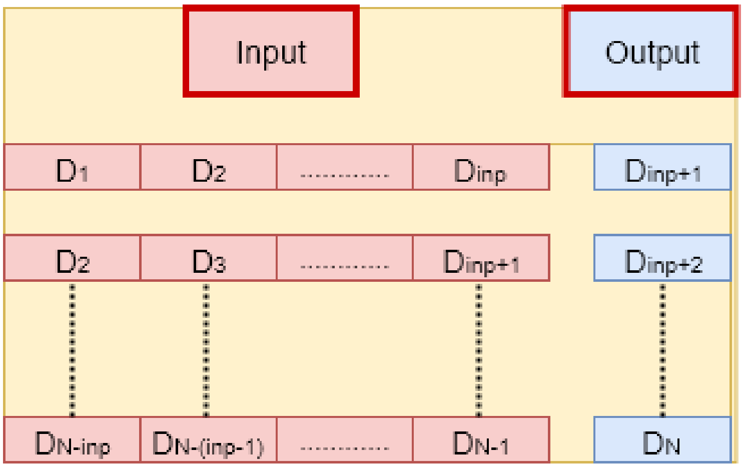


Figure 6.2: Input output dataset preparation

Table 6.1: Details of the benchmark models

Model	Description
WSF-Model-1	Persistence Model
WSF-Model-2	ELM
WSF-Model-3	OSELM
WSF-Model-4	EMD-CSO-OSELM

applications for its learning speed and accuracy. Further, EMD is a widely used data pre processing algorithm for its ability to decompose non- stationary and non-linear signals.

6.3 MODEL DEVELOPMENT

6.3.1 Details of the data used

Ten min resolution historical wind speed data collected in the year 2015 from SCADA system of a 1.5 MW wind turbine has been used for wind forecasting study in this work.

Wind speed fluctuation is mainly dependent on the season. To cover different climatic conditions and fluctuation characteristics, one month representative data from two seasons namely monsoon (July) and winter (January) have been used. This helps in establishing the forecasting performance and practicability of the proposed model.

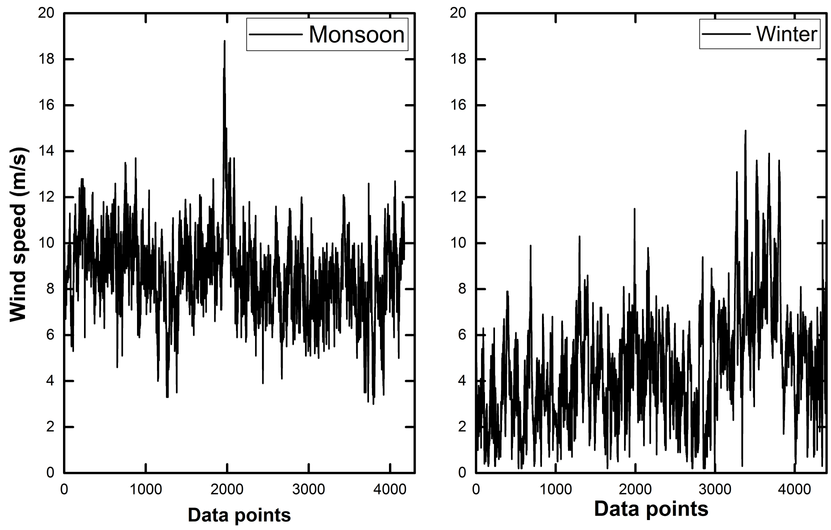


Figure 6.3: Original wind speed data (Monsoon and Winter)

Table 6.2: Statistics of wind speed data (m/s)

	Max	Min	Mean	Median
Winter	14.9	0.2	4.59	4.1
Monsoon	18.8	3	8.66	8.6

Variation of wind in these two seasons are plotted in Figure 6.3. A total of 4301 and 4400 data points are considered for monsoon and winter seasons respectively. The statistical parameters of wind data is presented in Table 6.2.

6.3.2 Decomposition of the wind speed series using OVMD

The noise in wind speed data is inevitable due to many reasons. This affects the performance of the WSF in a great manner. In view of improving the forecasting accuracy, denoising of the data by time series decomposition has been done in this study by using OVMD method. OVMD decomposes the original wind speed data into a number of subseries.

In OVMD, defining of K, the number of modes to be decomposed and τ, the update parameter, have been done using center frequency analysis and observing the REI value respectively as discussed above. The center frequencies corresponding to different values of K for decomposition of wind speed data for the two seasons are presented in Table 6.3. It can be noted that for both cases, similar frequencies are observed for mode number 5 and 6. Hence the optimal value of K is identified as 5 for both the seasons.

Fixing of update parameter τ is done based on trial and error method for minimum REI value. The REI values for different τ values are plotted in Figure 6.4. From the graph, it can be observed that, optimal values of τ for monsoon and winter are 0.9 and 0.92 respectively. The original wind speed data is hence decomposed using OVMD with the optimal values of K and τ . The original and decomposed subseries of wind speed data for monsoon and winter season are presented in Figure 6.5. From the figure it can be observed that Mode 1 is a low-frequency mode which shows the low-frequency oscillations and Mode 5 captures most of the noise hence is the high frequency mode.

6.3.3 Parameter selection

Inputs selection

The selection of suitable number of input features for all the models have been done using CS algorithm in this work. The number of features selected by CS algorithm for different cases such as original data, data decomposed by OVMD (Modes) and EMD (IMFs) are presented in Table 6.4. The number of IMFs extracted by EMD algorithm from monsoon and winter season are 16 and 18 respectively. In Table 6.4, the details for only 7 IMFs have been presented as a sample.

Parameter setting for CS algorithm

The values of the simulation parameters used in CS algorithm are listed below.

1. Stopping criteria =100 iterations.

Table 6.3: Values of center frequency using VMD method for different K values

Season	K	Center frequency					
Monsoon	2	2.234x 10^{-5}	0.018				
	3	1.61x10^{-5}	0.014	0.061			
	4	1.37x10^{-5}	0.013	0.047	0.109		
	5	1.25x10^{-5}	0.012	0.039	0.080	0.168	
	6	1.25x10^{-5}	0.012	0.039	0.080	0.167	0.423
Winter	2	3.883x10^{-4}	0.245				
	3	3.84x10^{-4}	0.158	0.331			
	4	3.79x10^{-4}	0.113	0.247	0.397		
	5	3.71x10^{-4}	0.088	0.197	0.297	0.397	
	6	3.70x10^{-4}	0.083	0.190	0.281	0.393	0.417

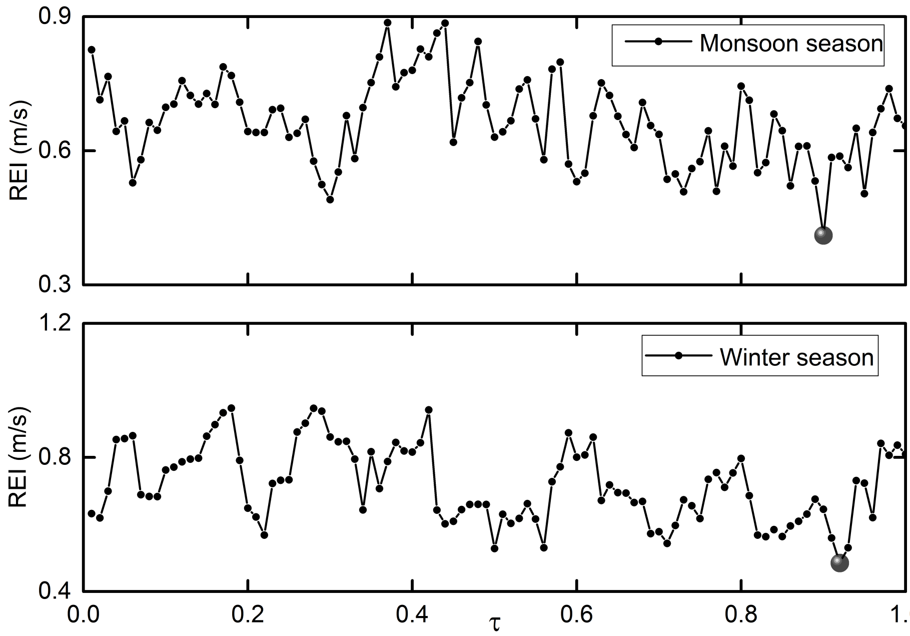
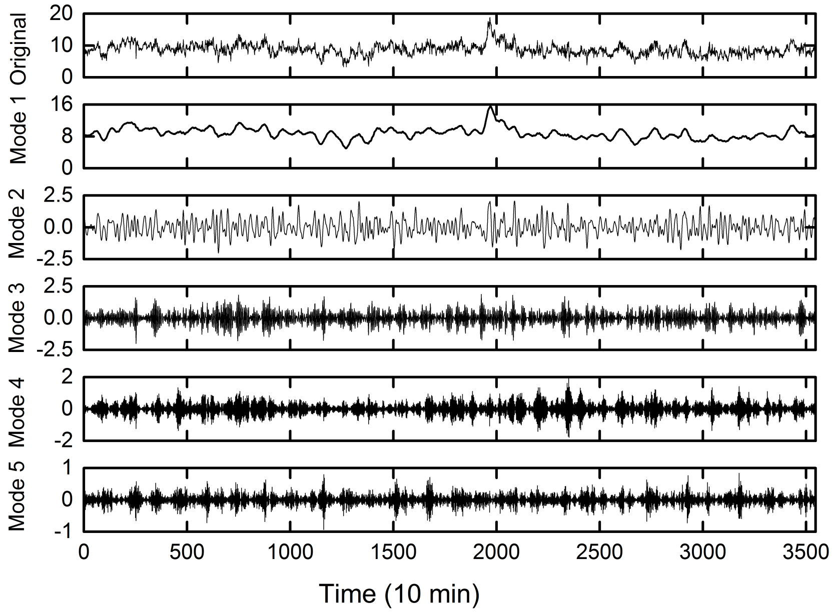
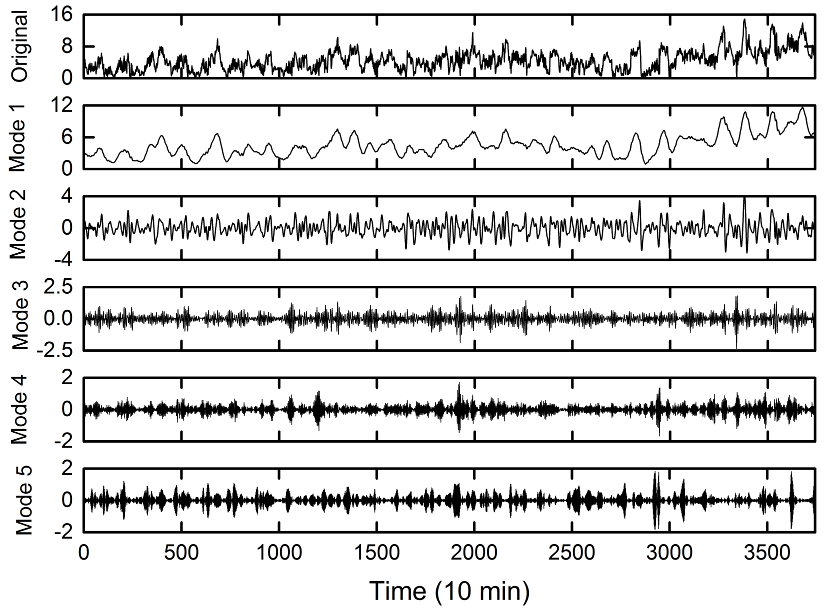


Figure 6.4: The variation of REI value with τ for wind speed data (monsoon and winter)



(a) Time series for monsoon season



(b) Time series for winter season

Figure 6.5: The original series and the decomposed subseries of the 10 min wind speed data by OVMD

Table 6.4: Details of the number of input features, Number of hidden neurons and chunk size, selected by CSO

	Monsoon season			Winter season		
Data	Number of input features	Number of hidden neurons	Chunk size	Number of input features	Number of hidden neurons	Chunk size
Original	9	15	76	13	92	40
OVMD-Mode 1	15	92	40	15	96	85
OVMD-Mode 2	14	96	18	15	96	85
OVMD-Mode 3	15	99	18	13	96	85
OVMD-Mode 4	15	96	85	13	92	40
OVMD-Mode 5	15	79	65	11	96	85
EMD-IMF 1	6	43	75	4	80	68
EMD-IMF 2	5	95	85	5	80	68
EMD-IMF 3	10	80	68	6	92	40
EMD-IMF 4	10	96	85	10	97	5
EMD-IMF 5	6	96	85	5	15	76
EMD-IMF 6	7	96	85	6	45	75
EMD-IMF 7	7	96	18	6	96	85

2. Distribution factor β=1.5.
3. Probability P_a =0.25.
4. Population size n =10.

Parameter setting for OSELM algorithm

In OSELM, the value of initial dataset N0 is fixed as 100. The other parameters needed to be set are number of hidden nodes and chunk size, which are optimized by CS algorithm. The lower and upper limit for these two variables have been set as 1 and 100 respectively. The optimum values have been presented in Table 6.4.

6.4 RESULTS AND DISCUSSION

To evaluate the forecasting ability of different models, three performance metrics namely RMSE, Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) as defined in Equation 6.11, 6.12 and 6.13 respectively are used in this study.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_p\left(i\right) - y_o\left(i\right)\right)^2} \tag{6.11}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y_p\left(i\right) - y_o\left(i\right)\right| \tag{6.12}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|y_p\left(i\right) - y_o\left(i\right)\right|}{y_o\left(i\right)} \times 100 \tag{6.13}$$

where $y_p\left(i\right)$, $y_o\left(i\right)$ are the forecasted and actual wind speed values at data point i respectively, N is the total number of data points.

To analyze the performance of the proposed model with other benchmark models selected for study in multi-step wind forecasting, the results for different horizons in the test period for monsoon season are presented in Table 6.5 and the performance rankings are shown in Figure 6.6.

The WSF-Model-1 though performed better than WSF-Model-2 in 1-step forecasting, performed poorly for higher steps in forecasting. WSF-Model-2 and WSF-Model-3 being single ANN models, showed sufficiently better performance for multistep WSF than WSF-Model-1, a persistence model. This is due to the ability of the ANN models to map complex input output relationships. Smaller values of MAE, MAPE and RMSE in WSF-Model-3 for all forecasting intervals in comparison to Model-2 is observed. This is due to online learning used in WSF-Model-3.Further, WSF-Model-4, an EMD based model showed moderate performance. It is better than WSF-Model-3 which did

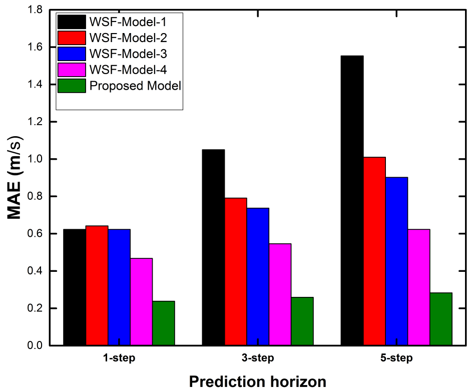
Table 6.5: Multi-step forecasting performance of the proposed model and benchmark models for monsoon season

	Proposed model			WSF-Model-4		
Index	1-step	3-step	5-step	1-step	3-step	5-step
MAE (m/s)	0.238	0.259	0.283	0.468	0.546	0.623
MAPE (%)	3.408	3.982	4.376	6.579	8.070	9.039
RMSE (m/s)	0.168	0.183	0.200	0.335	0.383	0.428
	WSF-Model-3			WSF-Model-2		
Index	1-step	3-step	5-step	1-step	3-step	5-step
MAE (m/s)	0.623	0.737	0.902	0.642	0.791	1.01
MAPE (%)	8.669	10.790	12.290	8.970	11.510	13.955
RMSE (m/s)	0.440	0.521	0.638	0.456	0.560	0.713
	WSF-Model-1					
Index	1-step	3-step	5-step			
MAE (m/s)	0.623	1.050	1.553			
MAPE (%)	8.592	14.587	20.962			
RMSE (m/s)	0.442	0.742	1.105			

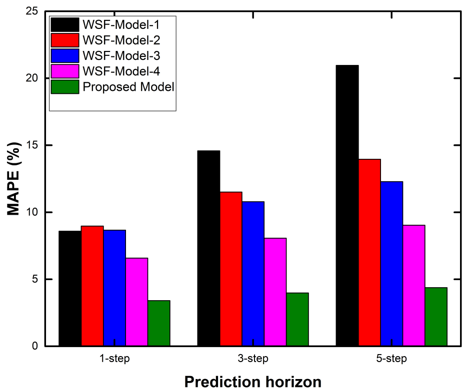
not use any denoising process. Use of data denoising improves the performance of the WSF model.

The performance of the proposed model (OVMD-CS-OSELM) in this study has been found to be the best among all models studied, for all forecasting horizons. The values of MAE, MAPE and RMSE are found to be the least. The graphical illustration of the test results of the proposed model and the actual values are presented in Figure 6.7. The forecasted values are close to actual thus proving the superiority of the model in 1-step as well as multi-step forecasting. However, the error is found to be increasing with increasing forecasting horizon.

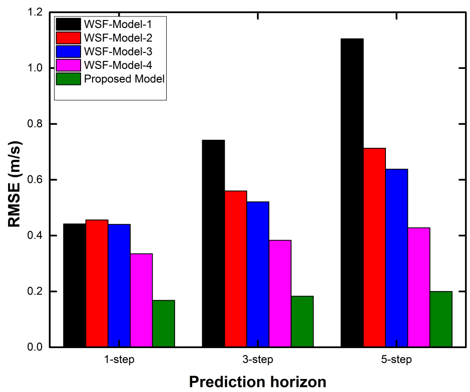
Table 6.6, Figures 6.8 and 6.9 presents similar results for winter season. The results show a similar trend and supports the observation that the proposed model gives the best possible performance for multi-step wind speed forecasting among the various models considered in the study.



(a) MAE

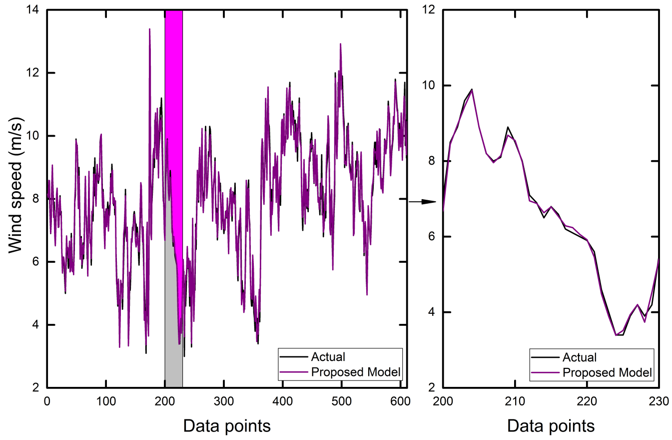


(b) MAPE

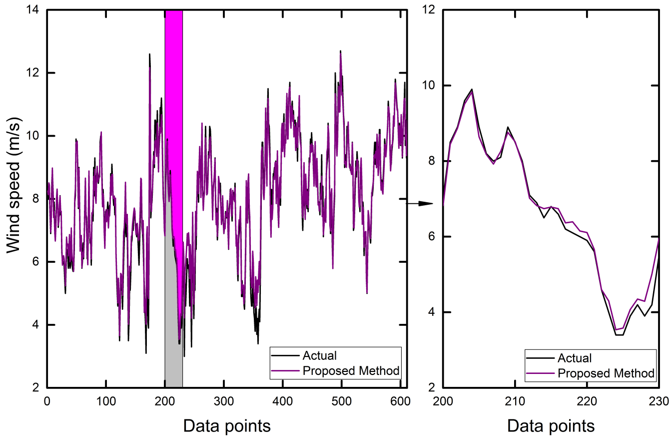


(c) RMSE

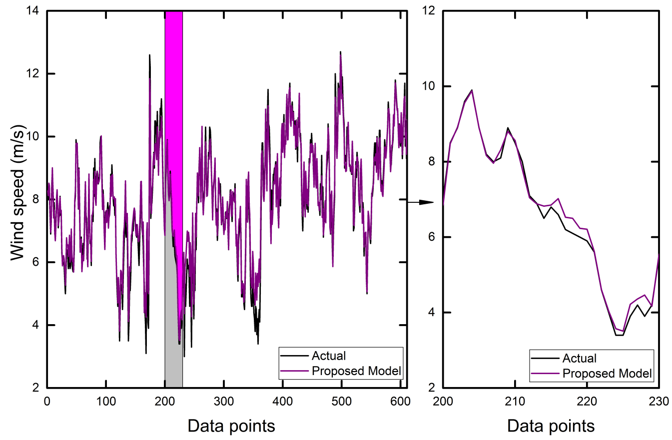
Figure 6.6: Multistep forecasting performance of models in terms of a) MAE, b)MAPE, c)RMSE in the test period for monsoon season



(a) 1-step ahead forecasting

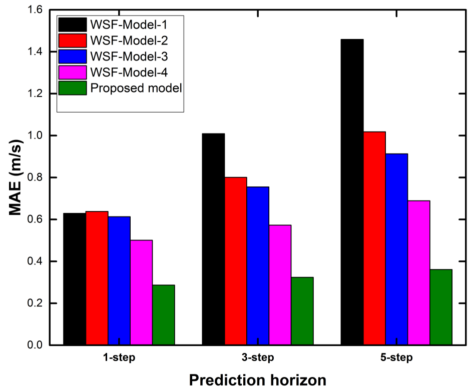


(b) 3-step ahead forecasting

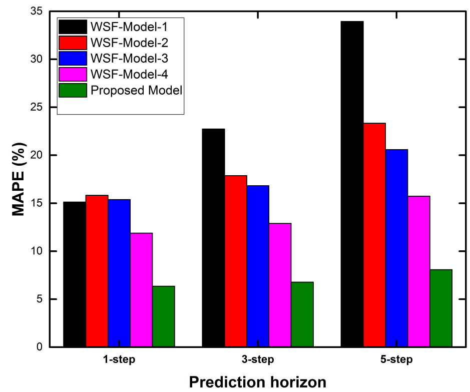


(c) 5-step ahead forecasting

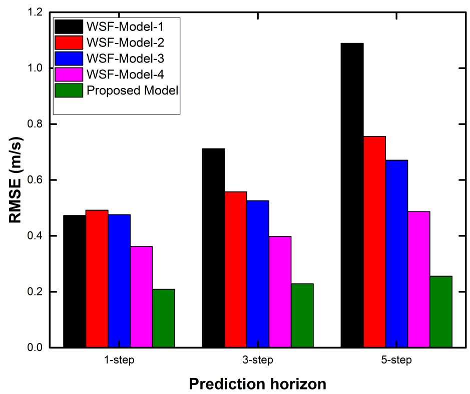
Figure 6.7: Multi-step forecasting results for test data of monsoon season



(a) MAE

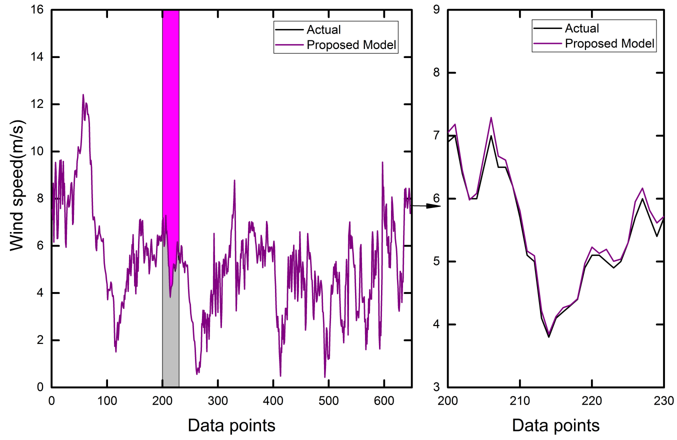


(b) MAPE

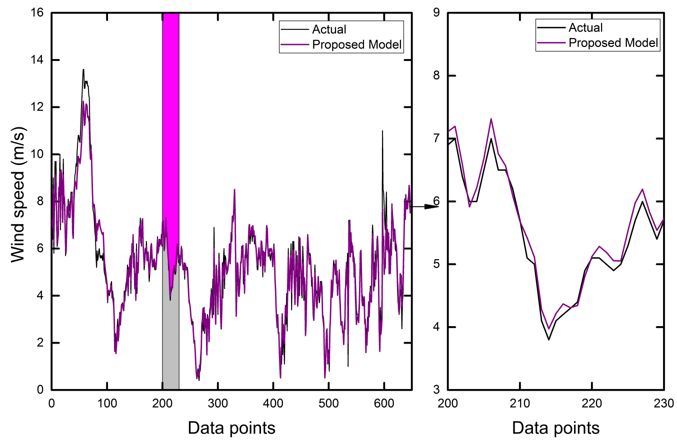


(c) RMSE

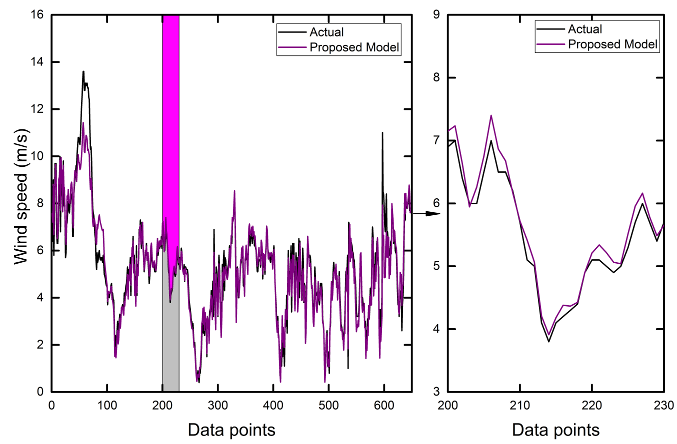
Figure 6.8: Multistep forecasting performance of models in terms of a) MAE, b)MAPE, c)RMSE in the test period for winter season



(a) 1-step ahead forecasting



(b) 3-step ahead forecasting



(c) 5-step ahead forecasting

Figure 6.9: Multi-step forecasting results for test data of winter season

Table 6.6: Multi-step forecasting performance of proposed model and benchmark models for winter season

	Proposed model			WSF-Model-4		
Index	1-step	3-step	5-step	1-step	3-step	5-step
MAE (m/s)	0.287	0.324	0.361	0.501	0.573	0.689
MAPE (%)	6.352	6.776	8.079	11.889	12.898	15.735
RMSE (m/s)	0.209	0.229	0.256	0.362	0.398	0.487
	WSF-Model-3			WSF-Model-2		
Index	1-step	3-step	5-step	1-step	3-step	5-step
MAE (m/s)	0.613	0.755	0.913	0.638	0.801	1.018
MAPE (%)	15.380	16.828	20.584	15.827	17.881	23.336
RMSE (m/s)	0.476	0.526	0.671	0.492	0.558	0.756
	WSF-Model-1					
Index	1-step	3-step	5-step			
MAE (m/s)	0.629	1.009	1.459			
MAPE (%)	15.122	22.739	33.941			
RMSE (m/s)	0.473	0.712	1.089			

6.5 COMPARISON OF WIND SPEED FORECASTING MODELS

6.5.1 Comparison of proposed model with other benchmark models

The comparison of performance of the proposed model for wind speed forecasting with that of the benchmark models is presented in this section. The comparison is done with respect to the method of learning (batch, online) and use of data pre-processing techniques (EMD, OVMD). This helps in understanding the merits and de-merits of the techniques in terms of percentage improvement in the performance. The performance of WSF-Model-3 and WSF-Model-2 are analyzed to demonstrate the advantages of OSELM over ELM. The percentage improvements of WSF-Model-3 over WSF-Model-2 are given in Table 6.7.

The benefits of OSELM in multi-step forecasting is clearly established from this

Table 6.7: Percentage improvement in performance of OSELM over ELM

Models	Prediction horizon	Monsoon season	Winter season
		PE_{MAPE}	PE_{MAPE}
WSF-Model-3 vs WSF-Model-2	1-step	3.355	2.824
	3-step	6.255	5.888
	5-step	11.931	11.792

table. An improvement in MAPE of 3.355% is observed for 1-step and for 5-step it is found as 11.931%. Hence it is clear that there is a drastic improvement in the performance with the prediction horizon. The online learning in OSELM has a definite impact on the forecasting performance of the model. The model adjusts to the changing conditions and fits to the recent dynamics of the wind speed data. The OSELM model has been compared and proved to be better than ELM. Further, the comparison of various OSELM based models are done in order to understand the benefits of denoising. Three OSELM based models, WSF-Model-3, WSF-Model-4 and the proposed model are analyzed in Table 6.8. The comparison is done progressively in two steps.

In the first step, the comparison is done between WSF-Model-3 and WSF-Model-4, where the former is a single OSELM model and the later is a hybrid of Emperical Mode Decomposition (EMD) and OSELM. An improvement of 24.108% and 26.452% is observed for WSF-Model-4 over WSF-Model-3 for 1-step and 5-step ahead forecasting respectively. This shows the significance of data denoising in improving the forecasting performance of the WSF model. Next, comparison of WSF-Model-4 and proposed model is done to illustrate the superiority of OVMD over EMD, a widely used denoising technique. An improvement of 48.198% and 51.587% for 1-step and 5-step forecasting respectively can be seen from the table. The improvement of proposed model over Model-4 with respect to denoising, is similar to that obtained by Chu Zhang et al, where there was an improvement of 52.14% (Zhang *et al.* 2017). From this it is evident that OVMD is superior to EMD. The reason for this is the sensitivity of EMD to noise and sampling. EMD mainly suffers from mode mixing problem. On the other hand OVMD

Table 6.8: Percentage improvement between WSF-Model-3, WSF-Model-4 and WSF-Model-4, proposed model

Model compared	Prediction horizon	Monsoon season	Winter season
		PE_{MAPE}	PE_{MAPE}
WSF-Model-4 vs WSF-Model-3	1-step	24.108	22.698
	3-step	25.208	23.353
	5-step	26.452	23.557
Proposed model vs WSF-Model-4	1-step	48.198	46.572
	3-step	50.656	47.464
	5-step	51.587	48.655

being a variant of VMD is robust than EMD and searches modes in a better way to reproduce the input data precisely. It uses Wiener filtering in Fourier domain for updating the modes. The OVMD follows a methodology to optimally fix the number of modes and update parameter and hence overcomes the limitations of VMD.

A critical comparison of the ANN models used for WSF has been carried out in this study. WSF-Model-2 and WSF-Model-3 are compared to understand the benefits of OSELM, an online learning algorithm over ELM, a batch learning algorithm in WSF. Comparison of WSF-Model-3 and WSF-Model-4 helps in understanding the advantages of data pre-processing in WSF model development. Finally, the proposed model is compared with WSF-Model-4 to study the advantages of OVMD over EMD algorithm. The three stage comparison helps in understanding the superiority and benefits of the proposed model over the existing ones.

6.5.2 Comparison of CS feature selection with PACF feature selection

The selection of input features for all the models including the proposed has been done by using the CS optimization algorithm. The details of number of features selected and the performance of the proposed model for monsoon season can be found in Table 6.4 and 6.5 respectively.

To further investigate the advantage of using CS, a metaheuristic algorithm for input feature selection, an attempt has been made to compare it with PACF based feature selection. The data for monsoon season has been used for this analysis. The partial correlogram is shown in Figure 6.10. From the partial correlogram, the features for which

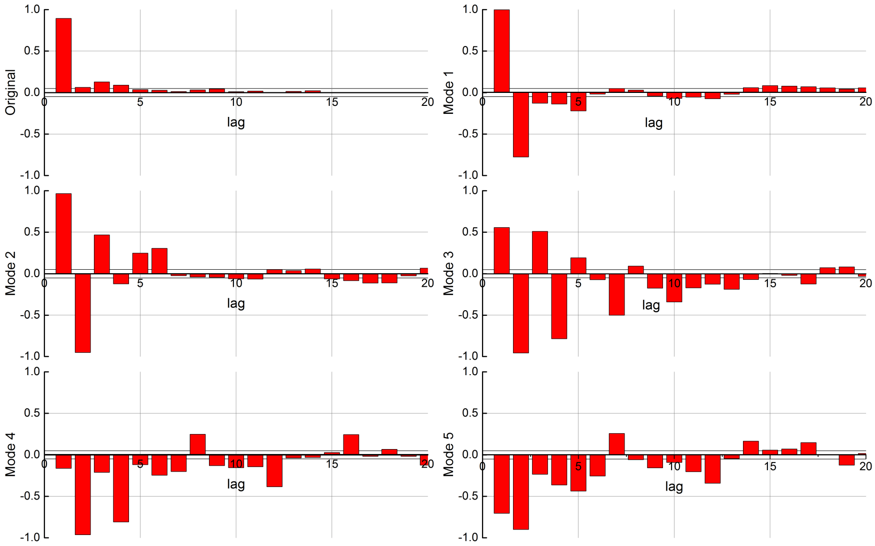


Figure 6.10: Partial correlogram of original and modes of OVMD for wind speed series(monsoon)

high correlation have been obtained with significance level above 95% are tabulated in Table 6.9. The training and test datasets are accordingly formed and are used in the proposed model. The multi-step forecasting results are presented in Table 6.10.

In order to compare the performance of the two input feature selection methods, the degree of improvement in terms MAPE is considered by using the metric given in Equation 6.14.

Table 6.9: Details of the number of input features selected using PACF

Monsoon season	Number of input features
Original	4
OVMD-Mode 1	5
OVMD-Mode 2	6
OVMD-Mode 3	14
OVMD-Mode 4	12
OVMD-Mode 5	17

Table 6.10: Multi-step forecasting results for the proposed model using PACF selected features

Index	1-step	3-step	5-step
MAE (m/s)	0.285	0.302	0.335
MAPE (%)	3.875	4.491	4.954
RMSE (m/s)	0.201	0.213	0.237

$$PE_{MAPE} = \left(\frac{MAPE2 - MAPE1}{MAPE2} \right) * 100 \tag{6.14}$$

where $MAPE1$ and $MAPE2$ are the performance indices of the two models using CS and PACF feature selection method (1 vs 2) considered for comparison.

The percentage improvement in MAPE by considering the results of the proposed model with input feature selection based on CS, presented in Table 6.5 over PACF, presented in Table 6.10, is given in Table 6.11. From the table, it can be seen that the WSF model with CS based feature selection shows 12.051%, 11.333% and 11.667% improvement in forecasting accuracy for 1-step, 3-step and 5-step forecasting respectively. This is because PACF, being a filter based algorithm, works on the assumption that each feature independently contributes in predicting the output. But CS being a wrapper based algorithm, considers multiple features for evaluation. Since in wind speed data, the assumption of independence is not valid, PACF results in suboptimal solution. Thus it is proved that CS, a widely used meta heuristic algorithm can greatly enhance the forecasting accuracy, when used for selection of input features in WSF model development.

Table 6.11: Percentage improvement in proposed model performance based on feature selection using CS

Comparison of input feature selection	Monsoon season	
	Prediction horizon	PE_{MAPE}
CSO vs PACF	1-step	12.051
	3-step	11.333
	5-step	11.667

The study proposed a hybrid WSF model by combining OSELM, OVMD and CS algorithm. OVMD is used to decompose the data into subseries, CS to select the input features for every subseries and to optimize the model parameters of the OSELM.

Ten min average wind speed data collected in monsoon and winter seasons from a SCADA system of a 1.5 MW wind turbine situated in central dry zone of Karnataka, India has been used. For the brief understanding of performance of some widely used ANN models found in the literature, the results of four benchmark models have been presented along with that of the proposed model. To prove the ability of CS algorithm to properly select the input features, the comparison of CS feature selection with PACF has been done. The results reveal the effectiveness of the proposed model in capturing the nonlinear characteristics of the wind speed and thus resulting in accurate wind speed forecasts.

6.5.3 Case study

To further prove the effectiveness of the proposed model, 10 min wind speed data collected in the database of meteorological station of GECAD renewable energy lab, a largest research and development unit from Polytechnic sub-system of Portugal has been used (IEEEdatasets 2019). Out of total data points of 4464 collected for January 2011, first 3795 were used for training and the rest 669 for test.

Portugal being a part of Iberian Peninsula, has a complex topography (Azorin-

Table 6.12: Statistics of GECAD data (m/s)

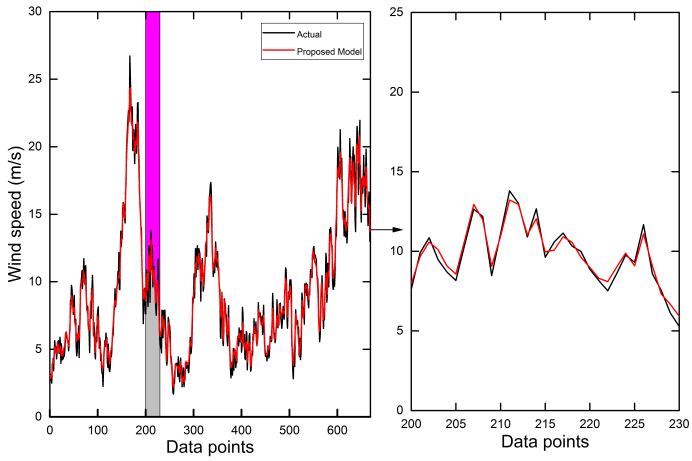
Max	Min	Mean	Median
37.25	0	10.027	8.365

Table 6.13: Multi-step forecasting performance of the proposed model for GECAD data

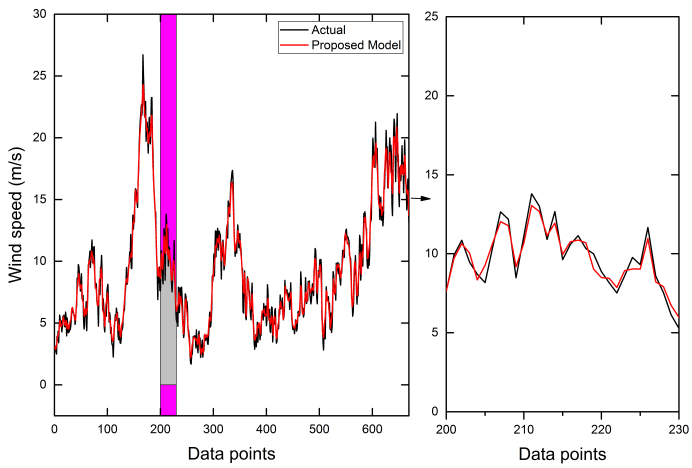
Index	1-step	3-step	5-step
MAE (m/s)	0.566	0.567	0.569
MAPE (%)	7.508	7.515	7.541
RMSE (m/s)	0.283	0.284	0.289

Molina *et al.* 2016).The statistical details of the wind speed data are given in Table 6.12. From Table 6.12, it can be observed that there is high variation in wind speed with minimum of 0 m/s and a maximum of 37.25 m/s in a period of one month.

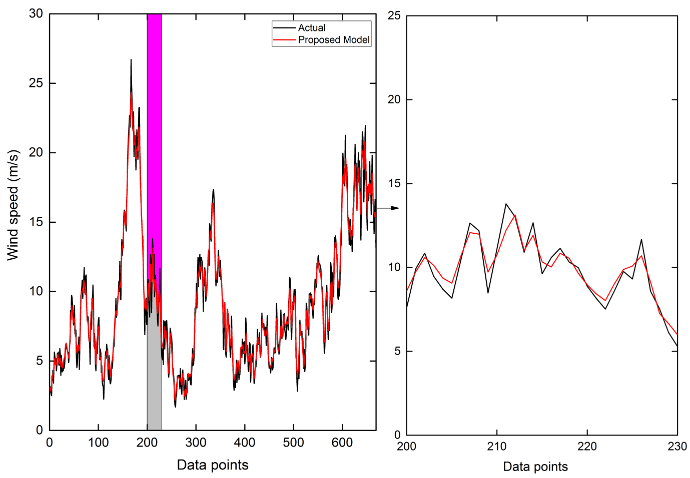
The forecasting results obtained from the proposed model has been presented in Table 6.13 . It can be noted that the model resulted in a reasonable accuracy for multi step forecasting, proving its effectiveness in forecasting wind speed of a region with complex topography. Figure 6.10 gives the corresponding results of multi step forecasting for 1- step, 3-step and 5-step. The error slightly increases with increase in the number of steps of forecasting.



(a) 1-step ahead forecasting



(b) 3-step ahead forecasting



(c) 5-step ahead forecasting

Figure 6.11: Multi-step forecasting results for GECAD data

CHAPTER 7

CONCLUSIONS AND SCOPE FOR FUTURE WORK

7.1 CONCLUSIONS

A comprehensive investigation of various conventional and ANN models for wind power modeling and optimization has been carried out in this study. The best ANN and optimization algorithms from this study is used to propose hybrid wind forecasting model. The data from Supervisory Control and Data Acquisition (SCADA) system of a wind farm present in Karnataka state, India has been used.

The conventional models developed in the study include wind power equation, models based on presumed shape of power curve, actual power curves supplied by the manufacturer and Response Surface Methodology (RSM).

The study focuses mainly on developing ANN models by considering two widely studied feed forward neural networks namely Multilayer Perceptron (MLP) and Radial Basis Function (RBF) and two different batch learning algorithms namely Backpropagation (BP) and Extreme Learning Machine (ELM). Detailed study on different center selection strategies in RBF neural network model development is carried out. Further, Online Sequential ELM (OSELM), a neural network model based on online learning is applied. Various aspects of OSELM such as activation function and modes of learning are studied.

Three metaheurisic optimization algorithms namely Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC) and Cuckoo Search (CS) are used to optimize the power output of the wind turbine by optimizing the blade pitch angle and the performances are compared. Two methods namely RSM and ANN are compared to develop the objective function.

The best ANN and optimization algorithms found from the above study are used to develop an efficient hybrid multistep wind speed forecasting (WSF) model. In addition to the use of OSELM and CS, a data pre-processing technique namely optimized Variational Mode Decomposition (OVMD) is combined to further improve the forecasting accuracy of the model.

The results obtained from the study lead to the following conclusions:

1. Model-5, based on RSM is found to be the best, among the various conventional models studied, as it considers many parameters affecting the power in addition to wind speed. Other conventional models consider only wind speed as the input, are complex and inaccurate. The difference in actual power and that provided in manufacture's power curve in addition to restricted nature of the curve fitting technique make models based on power curve inefficient.

2. Models based on MLP neural network show compact network structure compared to RBF neural network models, but have less generalization capability. The performance of RBF model improves remarkably with the use of clustering algorithms. PSO-FCM clustering algorithm performed well in comparison to three other strategies used in the study.

3. Models based on ELM learning have less simulation parameters. They converge extremely faster with better generalization performance and generate compact network structure compared to that based on BP. Model-15, a fully optimized RBF model, using PSO for optimizing number of center and width of RBF units, is found to be the best ANN model based on ELM batch learning in terms of performance and ease of fixing the model parameters.

4. The OSELM model is equally efficient compared to Model-15 with only a marginal accuracy difference. But it is more suitable for the present application, due to the

online nature of wind, which otherwise demands large data storage. Hence OS-ELM is proposed as the most suitable model for wind power prediction.

5. ANN based objective function is superior compared to RSM in wind power optimization problem. CS is found to be the most suitable optimization algorithm compared to ABC and PSO due to fast convergence and higher values of Mean PG and Mean Relative PG. Increase of 17.329 (%) in mean relative hourly power output of the wind turbine on considering the optimized blade pitch angle obtained by CS can be a significant contribution towards making wind energy economical.

6. The proposed hybrid multi-step wind speed forecasting model which uses OVMD for denoising the data, CS algorithm for selection of input features and model parameters of OSELM model resulted in superior performance compared to all the other benchmark models selected from literature by showing clear benefits of OSELM over ELM, OVMD over EMD and CS over PACF for modeling, data pre-processing and input feature selection respectively.

7.2 SCOPE FOR FUTURE WORK

The deep learning techniques have been found to be advantageous, when compared to the conventional machine learning algorithms, especially when a large and complex dataset in terms of number of features and their inter relationships is available for training. These are proved to be effective in solving complex problems. For wind power and speed related modeling applications, huge data is available in either the SCADA of the wind turbine or the meteorological stations and this data is online in nature. Thus, deep learning techniques associated with online learning, which uses data 1- by-1 or chunk-by-chunk can be used in developing a wind power prediction and wind speed forecasting model.

Different metaheuristic algorithms use different method of exploration and exploitation in searching the optimal solution, each having its own pros and cons. A proper combination of two metaheuristic algorithms can result in better solutions compared to the use of single algorithm. Hence, a hybrid optimization algorithm can be used for optimizing the blade pitch angle of the wind turbine and hence maximize the wind turbine power output.

REFERENCES

Abdoos, A. A. (2016). "A new intelligent method based on combination of VMD and ELM for short term wind power forecasting". *Neurocomputing*, 203, 111–120.

Abouzahr, I. and R. Ramakumar (1990). "Loss of power supply probability of stand-alone wind electric conversion systems: A closed form solution approach". *IEEE Transactions on Energy Conversion*, 5(3), 445–452.

Ai, B., H. Yang, H. Shen, and X. Liao (2003). "Computer-aided design of PV/wind hybrid system". *Renewable energy*, 28(10), 1491–1512.

Azorin-Molina, C., J.-A. Guijarro, T. R. McVicar, S. M. Vicente-Serrano, D. Chen, S. Jerez, and F. Espírito-Santo (2016). "Trends of daily peak wind gusts in Spain and Portugal, 1961–2014". *Journal of Geophysical Research: Atmospheres*, 121(3), 1059–1078.

Bagul, A. C., A. M. Kulkarni, and A. S. Dayma (2018). "Review of Wind Energy Market".

Bezdek, J. C., *"Pattern recognition with fuzzy objective function algorithms"*. Springer Science & Business Media, 2013.

Cadenas, E. and W. Rivera (2010). "Wind speed forecasting in three different regions of Mexico, using a hybrid ARIMA–ANN model". *Renewable Energy*, 35(12), 2732–2738.

Cancelliere, R., A. Gosso, and A. Grosso, "Neural networks for wind power generation forecasting: A case study". *In Networking, Sensing and Control (ICNSC), 2013 10th IEEE International Conference on*. IEEE, 2013.

Carrillo, C., A. O. Montaño, J. Cidrás, and E. Díaz-Dorado (2013). "Review of power curve modelling for wind turbines". *Renewable and Sustainable Energy Reviews*, 21, 572–581.

Chang, G., H. Lu, Y. Chang, and Y. Lee (2017). "An improved neural network-based approach for short-term wind speed and power forecast". *Renewable energy*, 105, 301–311.

Chrouta, J., A. Zaafouri, and M. Jemli, "Identification of an irrigation station using hybrid fuzzy clustering algorithms based on particle swarm optimization". *In Systems, Signals & Devices (SSD), 2015 12th International Multi-Conference on*. IEEE, 2015.

Civicioglu, P. and E. Besdok (2013). “A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms”. *Artificial intelligence review*, 39(4), 315–346.

Commission, I. E. *et al.* (1998). “IEC61400-12: Wind turbine generator systems-Part 12: Wind turbine power performance esting”.

Council, G. W. E. (2016). “Indian wind energy: a brief outlook 2016”. *Global Wind Energy Council, Brussels, Belgium.*

Diaf, S., D. Diaf, M. Belhamel, M. Haddadi, and A. Louche (2007). “A methodology for optimal sizing of autonomous hybrid PV/wind system”. *Energy Policy*, 35(11), 5708–5718.

Dragomiretskiy, K. and D. Zosso (2014). “Variational mode decomposition”. *IEEE transactions on signal processing*, 62(3), 531–544.

Erdem, E. and J. Shi (2011). “ ARMA based approaches for forecasting the tuple of wind speed and direction”. *Applied Energy*, 88(4), 1405–1414.

Fan, S., J. R. Liao, R. Yokoyama, L. Chen, and W.-J. Lee (2009). “Forecasting the wind generation using a two-stage network based on meteorological information”. *IEEE Transactions on Energy Conversion*, 24(2), 474–482.

Feng, C., M. Cui, B.-M. Hodge, and J. Zhang (2017). “A data-driven multi-model methodology with deep feature selection for short-term wind forecasting”. *Applied Energy*, 190, 1245–1257.

Gautam, P. K. and G. K. Venayagamoorthy, “Dynamic performance model of wind turbine generators”. *In Computational Intelligence Applications In Smart Grid (CIASG), 2013 IEEE Symposium on*. IEEE, 2013.

Goudarzi, A., I. E. Davidson, A. Ahmadi, and G. K. Venayagamoorthy, “Intelligent analysis of wind turbine power curve models”. *In Computational Intelligence Applications in Smart Grid (CIASG), 2014 IEEE Symposium on*. IEEE, 2014.

Grady, S., M. Hussaini, and M. M. Abdullah (2005). “Placement of wind turbines using genetic algorithms”. *Renewable energy*, 30(2), 259–270.

Guo, L., C. Wang, P. Gao, Y. Wang, Y. Zhong, and M. Huang, “An online short-term wind power prediction considering wind speed correction and error interval evaluation”. *In Information Science, Electronics and Electrical Engineering (ISEEE), 2014 International Conference on*volume1. IEEE, 2014.

Guo, Z., J. Zhao, W. Zhang, and J. Wang (2011). “A corrected hybrid approach for wind speed prediction in Hexi Corridor of China”. *Energy*, 36(3), 1668–1679.

Hagan, M. T., H. B. Demuth, M. H. Beale, and O. De Jesús, *“Neural network design”*volume20. PWS publishing company Boston, 1996.

Hameed, Z., Y. Hong, Y. Cho, S. Ahn, and C. Song (2009). “Condition monitoring and fault detection of wind turbines and related algorithms: A review”. *Renewable and Sustainable energy reviews*, 13(1), 1–39.

Haykin, S. and N. Network (2004). “A comprehensive foundation”. *Neural Networks*, 2(2004), 41.

Herbert, G. J., S. Iniyan, and R. Goic (2010). “Performance, reliability and failure analysis of wind farm in a developing country”. *Renewable Energy*, 35(12), 2739–2751.

Hocaoğlu, F. O., Ö. N. Gerek, and M. Kurban (2009). “A novel hybrid (wind–photovoltaic) system sizing procedure”. *Solar Energy*, 83(11), 2019–2028.

Hu, Y. H. and J.-N. Hwang, *“Handbook of neural network signal processing”*. CRC press, 2001.

Huang, G., G.-B. Huang, S. Song, and K. You (2015). “Trends in extreme learning machines: a review”. *Neural Networks*, 61, 32–48.

Huang, G.-B., Q.-Y. Zhu, and C.-K. Siew, “Extreme learning machine: a new learning scheme of feedforward neural networks”. *In Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*volume2. IEEE, 2004.

Huang, G.-B., Q.-Y. Zhu, and C.-K. Siew (2006). “Extreme learning machine: theory and applications”. *Neurocomputing*, 70(1), 489–501.

Huang, N. E., Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, “The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis”. *In Proceedings of the Royal Society of London A: mathematical, physical and engineering sciences*volume454. The Royal Society, 1998.

IEEEdatasets (2019). Accessed: 2019-02-04.

Kaganov, E. and A. Yaglom (1976). “Errors in wind-speed measurements by rotation anemometers”. *Boundary-Layer Meteorology*, 10(1), 15–34.

Kalogirou, S. A. (2001). “Artificial neural networks in renewable energy systems applications: a review”. *Renewable and sustainable energy reviews*, 5(4), 373–401.

Karaboga, D. (2005). “An idea based on honey bee swarm for numerical optimization”. Technical report, Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.

Kennedy, J., “Particle swarm optimization”. *In Encyclopedia of Machine Learning*. Springer, 2010, 760–766.

Kenway, G. and J. Martins, “Aerostructural shape optimization of wind turbine blades considering site-specific winds”. *In 12th AIAA/ISSMO multidisciplinary analysis and optimization conference*. 2008.

Khare, V., S. Nema, and P. Baredar (2013). “Status of solar wind renewable energy in India”. *Renewable and Sustainable Energy Reviews*, 27, 1–10.

Kongnam, C. and S. Nuchprayoon (2010). “A particle swarm optimization for wind energy control problem”. *Renewable Energy*, 35(11), 2431–2438.

Koziel, S. and X.-S. Yang, *“Computational optimization, methods and algorithms”*volume356. Springer, 2011.

Kusiak, A., Z. Song, and H. Zheng (2009*a*). “Anticipatory control of wind turbines with data-driven predictive models”. *IEEE Transactions on Energy Conversion*, 24(3), 766–774.

Kusiak, A. and Z. Zhang (2011). “Adaptive control of a wind turbine with data mining and swarm intelligence”. *IEEE Transactions on Sustainable Energy*, 2(1), 28–36.

Kusiak, A. and Z. Zhang (2012). “Control of wind turbine power and vibration with a data-driven approach”. *Renewable Energy*, 43, 73–82.

Kusiak, A., Z. Zhang, and M. Li (2010*a*). “Optimization of wind turbine performance with data-driven models”. *IEEE Transactions on Sustainable Energy*, 1(2), 66–76.

Kusiak, A., H. Zheng, and Z. Song (2009*b*). “Models for monitoring wind farm power”. *Renewable Energy*, 34(3), 583–590.

Kusiak, A., H. Zheng, and Z. Song (2009*c*). “On-line monitoring of power curves”. *Renewable Energy*, 34(6), 1487–1493.

Kusiak, A., H. Zheng, and Z. Song (2010*b*). “Power optimization of wind turbines with data mining and evolutionary computation”. *Renewable energy*, 35(3), 695–702.

Labate, D., F. Foresta, G. Occhiuto, F. C. Morabito, A. Lay-Ekuakille, and P. Vergallo (2013). “Empirical mode decomposition vs. wavelet decomposition for the extraction of respiratory signal from single-channel ECG: A comparison”. *IEEE Sensors Journal*, 13(7), 2666–2674.

Lapira, E., D. Brisset, H. D. Ardakani, D. Siegel, and J. Lee (2012). “Wind turbine performance assessment using multi-regime modeling approach”. *Renewable Energy*, 45, 86–95.

Li, G. and J. Shi (2010*a*). “Application of Bayesian model averaging in modeling long-term wind speed distributions”. *Renewable Energy*, 35(6), 1192–1202.

Li, G. and J. Shi (2010*b*). "On comparing three artificial neural networks for wind speed forecasting". *Applied Energy*, 87(7), 2313–2320.

Li, G., J. Shi, and J. Zhou (2011). "Bayesian adaptive combination of short-term wind speed forecasts from neural network models". *Renewable Energy*, 36(1), 352–359.

Li, S., D. C. Wunsch, E. OHair, and M. G. Giesselmann (2001*a*). "Comparative analysis of regression and artificial neural network models for wind turbine power curve estimation". *Journal of Solar Energy Engineering*, 123(4), 327–332.

Li, S., D. C. Wunsch, E. O'Hair, M. G. Giesselmann, *et al.* (2001*b*). "Using neural networks to estimate wind turbine power generation". *Energy conversion, IEEE Transactions on*, 16(3), 276–282.

Li, Z., L. Ye, Y. Zhao, X. Song, J. Teng, and J. Jin (2016). "Short-term wind power prediction based on extreme learning machine with error correction". *Protection and Control of Modern Power Systems*, 1(1), 1.

Liang, N.-Y., G.-B. Huang, P. Saratchandran, and N. Sundararajan (2006). "A fast and accurate online sequential learning algorithm for feedforward networks". *IEEE Transactions on neural networks*, 17(6), 1411–1423.

Lin, Q., J. Wang, and W. Qiao, "Denoising of wind speed data by wavelet thresholding". *In Chinese Automation Congress (CAC), 2013*. IEEE, 2013.

Liu, D., D. Niu, H. Wang, and L. Fan (2014). "Short-term wind speed forecasting using wavelet transform and support vector machines optimized by genetic algorithm". *Renewable Energy*, 62, 592–597.

Liu, H., C. Chen, H.-q. Tian, and Y.-f. Li (2012*a*). "A hybrid model for wind speed prediction using empirical mode decomposition and artificial neural networks". *Renewable Energy*, 48, 545–556.

Liu, H., Z. Duan, F.-z. Han, and Y.-f. Li (2018). "Big multi-step wind speed forecasting model based on secondary decomposition, ensemble method and error correction algorithm". *Energy Conversion and Management*, 156, 525–541.

Liu, H., H. Tian, X. Liang, and Y. Li (2015*a*). "New wind speed forecasting approaches using fast ensemble empirical model decomposition, genetic algorithm, Mind Evolutionary Algorithm and Artificial Neural Networks". *Renewable Energy*, 83, 1066–1075.

Liu, H., H.-q. Tian, and Y.-f. Li (2012*b*). "Comparison of two new ARIMA-ANN and ARIMA-Kalman hybrid methods for wind speed prediction". *Applied Energy*, 98, 415–424.

Liu, H., H.-q. Tian, and Y.-f. Li (2015*b*). "Four wind speed multi-step forecasting models using extreme learning machines and signal decomposing algorithms". *Energy conversion and management*, 100, 16–22.

Liu, H., H.-q. Tian, X.-f. Liang, and Y.-f. Li (2015*c*). "Wind speed forecasting approach using secondary decomposition algorithm and Elman neural networks". *Applied Energy*, 157, 183–194.

Liu, H., H.-q. Tian, D.-f. Pan, and Y.-f. Li (2013). "Forecasting models for wind speed using wavelet, wavelet packet, time series and Artificial Neural Networks". *Applied Energy*, 107, 191–208.

Liu, Z., W. Gao, Y.-H. Wan, and E. Muljadi, "Wind power plant prediction by using neural networks". *In Energy Conversion Congress and Exposition (ECCE)*. IEEE, 2012*c*.

Lydia, M., S. S. Kumar, A. I. Selvakumar, and G. E. P. Kumar (2014). "A comprehensive review on wind turbine power curve modeling techniques". *Renewable and Sustainable Energy Reviews*, 30, 452–460.

Lydia, M., A. I. Selvakumar, S. S. Kumar, and G. E. P. Kumar (2013). "Advanced algorithms for wind turbine power curve modeling". *IEEE Transactions on sustainable energy*, 4(3), 827–835.

Mabel, M. C. and E. Fernandez (2008). "Analysis of wind power generation and prediction using ANN: a case study". *Renewable Energy*, 33(5), 986–992.

Mabel, M. C. and E. Fernandez (2009). "Estimation of energy yield from wind farms using artificial neural networks". *Energy Conversion, IEEE Transactions on*, 24(2), 459–464.

Manwell, J. F., J. G. McGowan, and A. L. Rogers, *"Wind energy explained: theory, design and application"*. John Wiley & Sons, 2010.

Marvuglia, A. and A. Messineo (2012). "Monitoring of wind farms power curves using machine learning techniques". *Applied Energy*, 98, 574–583.

Mi, X.-w., H. Liu, and Y.-f. Li (2017). "Wind speed forecasting method using wavelet, extreme learning machine and outlier correction algorithm". *Energy Conversion and Management*, 151, 709–722.

Minitab, I. (2014). " MINITAB release 17: statistical software for windows". *Minitab Inc, USA*.

Mittal, A. (2010). *"Optimization of the layout of large wind farms using a genetic algorithm"*. Ph.D. thesis, Case Western Reserve University.

Moghaddam, M. G. and M. Khajeh (2011). "Comparison of response surface methodology and artificial neural network in predicting the microwave-assisted extraction procedure to determine zinc in fish muscles". *Food and Nutrition Sciences*, 2(08), 803.

Mohandes, M. A., T. O. Halawani, S. Rehman, and A. A. Hussain (2004). “Support vector machines for wind speed prediction”. *Renewable Energy*, 29(6), 939–947.

Molina, M. G. and J. G. Alvarez, “Technical and regulatory exigencies for grid connection of wind generation”. *In Wind Farm-Technical Regulations, Potential Estimation and Siting Assessment*. InTech, 2011.

Morteza (2018). `https://www.google.com/search?client=firefox-b-ab&biw`. Accessed: 2018-12-06.

Nagabhushana, T. (1996). *“Fault diagnosis of AC and AC-DC systems using constructive learning RBF neural networks [Ph. D. thesis]”*. Ph.D. thesis, Dept. of HighVoltage Engineering, IISc, Bangalore,India.

Naik, J., S. Dash, P. Dash, and R. Bisoi (2018). “Short term wind power forecasting using hybrid variational mode decomposition and multi-kernel regularized pseudo inverse neural network”. *Renewable Energy*, 118, 180–212.

Nakai, T. and K. Shimoyama (2012). “Ultrasonic anemometer angle of attack errors under turbulent conditions”. *Agricultural and forest meteorology*, 162, 14–26.

Nigim, K. and P. Parker (2007). “Heuristic and probabilistic wind power availability estimation procedures: Improved tools for technology and site selection”. *Renewable Energy*, 32(4), 638–648.

Niu, T., J. Wang, K. Zhang, and P. Du (2018). “Multi-step-ahead wind speed forecasting based on optimal feature selection and a modified bat algorithm with the cognition strategy”. *Renewable Energy*, 118, 213–229.

Ouyang, T., A. Kusiak, and Y. He (2017). ”Modeling wind-turbine power curve: A data partitioning and mining approach. *Renewable Energy*, 102, 1–8.

Pai, P. S. (2004). *“Acoustic emission based tool wear monitoring using some improved neural network methodologies”*. Ph.D. thesis, SJ College of Engineering, University of Mysore, Mysore, India.

Pai, P. S., B. S. Rao, *et al.* (2011). “Artificial neural network based prediction of performance and emission characteristics of a variable compression ratio CI engine using WCO as a biodiesel at different injection timings”. *Applied Energy*, 88(7), 2344–2354.

Paliwal, M. and U. A. Kumar (2009). “Neural networks and statistical techniques: A review of applications”. *Expert systems with applications*, 36(1), 2–17.

Pedrycz, W. (1998). “Conditional fuzzy clustering in the design of radial basis function neural networks”. *Neural Networks, IEEE Transactions on*, 9(4), 601–612.

Pelletier, F., C. Masson, and A. Tahan (2016). “Wind turbine power curve modelling using artificial neural network”. *Renewable Energy*, 89, 207–214.

Peng, X., W. Zheng, D. Zhang, Y. Liu, D. Lu, and L. Lin (2017). "A novel probabilistic wind speed forecasting based on combination of the adaptive ensemble of on-line sequential ORELM (Outlier Robust Extreme Learning Machine) and TVMCF (time-varying mixture copula function)". *Energy Conversion and Management*, 138, 587–602.

Petković, D. and S. Shamshirband (2015). "Soft methodology selection of wind turbine parameters to large affect wind energy conversion". *International Journal of Electrical Power & Energy Systems*, 69, 98–103.

Progress, P. (2017). "Ministry of New and Renewable Energy". *Govt. of India*, 31.

Ramachandra, T. and B. Shruthi (2003). "Wind energy potential in Karnataka, India". *Wind Engineering*, 27(6), 549–553.

Ramasamy, P., S. Chandel, and A. K. Yadav (2015). "Wind speed prediction in the mountainous region of India using an artificial neural network model". *Renewable Energy*, 80, 338–347.

Ren, Y., P. Suganthan, and N. Srikanth (2015). "A comparative study of empirical mode decomposition-based short-term wind speed forecasting methods". *IEEE Transactions on Sustainable Energy*, 6(1), 236–244.

Riahy, G. and M. Abedi (2008). "Short term wind speed forecasting for wind turbine applications using linear prediction method". *Renewable energy*, 33(1), 35–41.

Salcedo-Sanz, S., A. Pastor-Sánchez, J. Del Ser, L. Prieto, and Z.-W. Geem (2015). "A coral reefs optimization algorithm with harmony search operators for accurate wind speed prediction". *Renewable Energy*, 75, 93–101.

Salcedo-Sanz, S., A. Pastor-Sánchez, L. Prieto, A. Blanco-Aguilera, and R. García-Herrera (2014). "Feature selection in wind speed prediction systems based on a hybrid coral reefs optimization–Extreme learning machine approach". *Energy Conversion and Management*, 87, 10–18.

Sánchez, I. (2008). "Adaptive combination of forecasts with application to wind energy". *International Journal of Forecasting*, 24(4), 679–693.

Schlechtingen, M., I. F. Santos, and S. Achiche (2013). "Using data-mining approaches for wind turbine power curve monitoring: a comparative study". *IEEE Transactions on Sustainable Energy*, 4(3), 671–679.

Schlink, U. and G. Tetzlaff (1998). "Wind speed forecasting from 1 to 30 minutes". *Theoretical and applied climatology*, 60(1-4), 191–198.

Sfetsos, A. (2002). "A novel approach for the forecasting of mean hourly wind speed time series". *Renewable energy*, 27(2), 163–174.

Shokrzadeh, S., M. J. Jozani, and E. Bibeau (2014). "Wind turbine power curve modeling using advanced parametric and nonparametric methods". *IEEE Transactions on Sustainable Energy*, 5(4), 1262–1269.

Soriano, J. B., S. P. Wani, A. N. Rao, G. L. Sawargaonkar, and J. A. Gowda (2018). "Comparative evaluation of direct dry-seeded and transplanted rice in the dry zone of Karnataka, India". *Philippine Journal of Science*, 147(1), 165–174.

StatSoft, I. (2001). "STATISTICA (data analysis software system), version 6". *Tulsa, USA*, 150.

Thapar, V., G. Agnihotri, and V. K. Sethi (2011). "Critical analysis of methods for mathematical modelling of wind turbines". *Renewable Energy*, 36(11), 3166–3177.

Tu, Y.-L., T.-J. Chang, C.-L. Chen, and Y.-J. Chang (2012). "Estimation of monthly wind power outputs of WECS with limited record period using artificial neural networks". *Energy conversion and management*, 59, 114–121.

Tu, Y.-L., T.-J. Chang, C.-I. Hsieh, and J.-Y. Shih (2010). "Artificial neural networks in the estimation of monthly capacity factors of WECS in Taiwan". *Energy Conversion and Management*, 51(12), 2938–2946.

Üstüntaş, T. and A. D. Şahin (2008). "Wind turbine power curve estimation based on cluster center fuzzy logic modeling". *Journal of Wind Engineering and Industrial Aerodynamics*, 96(5), 611–620.

Wan, C., Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong (2014). "Probabilistic forecasting of wind power generation using extreme learning machine". *Power Systems, IEEE Transactions on*, 29(3), 1033–1044.

Wang, D., H. Luo, O. Grunder, and Y. Lin (2017). "Multi-step ahead wind speed forecasting using an improved wavelet neural network combining variational mode decomposition and phase space reconstruction". *Renewable Energy*, 113, 1345–1358.

Wang, J., J. Hu, K. Ma, and Y. Zhang (2015). "A self-adaptive hybrid approach for wind speed forecasting". *Renewable Energy*, 78, 374–385.

Wang, S., N. Zhang, L. Wu, and Y. Wang (2016). "Wind speed forecasting based on the hybrid ensemble empirical mode decomposition and GA-BP neural network method". *Renewable Energy*, 94, 629–636.

Wang, W., Y. Zhang, Y. Li, and X. Zhang, "The global fuzzy c-means clustering algorithm". *In Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*volume1. IEEE, 2006.

Yang, X.-S., *"Nature-inspired metaheuristic algorithms"*. Luniver press, 2010.

Yang, X.-S. and S. Deb, "Cuckoo search via Lévy flights". *In Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*. IEEE, 2009.

Zhang, C., H. Wei, J. Zhao, T. Liu, T. Zhu, and K. Zhang (2016*a*). "Short-term wind speed forecasting using empirical mode decomposition and feature selection". *Renewable Energy*, 96, 727–737.

Zhang, C., J. Zhou, C. Li, W. Fu, and T. Peng (2017). "A compound structure of ELM based on feature selection and parameter optimization using hybrid backtracking search algorithm for wind speed forecasting". *Energy Conversion and Management*, 143, 360–376.

Zhang, Y., K. Liu, L. Qin, and X. An (2016*b*). "Deterministic and probabilistic interval prediction for short-term wind power generation based on variational mode decomposition and machine learning methods". *Energy Conversion and Management*, 112, 208–219.

Zhang, Y., J. Wang, and X. Wang (2014). "Review on probabilistic forecasting of wind power generation". *Renewable and Sustainable Energy Reviews*, 32, 255–270.

Zhao, J., J. Wang, and F. Liu (2015). "Multistep forecasting for short-term wind speed using an optimized extreme learning machine network with decomposition-based signal filtering". *Journal of Energy Engineering*, 142(3), 04015036.

Zheng-zhong, Z., J. Yi-min, Z. Wen-hui, and X. Tian, "Prediction of short-term power output of wind farms based on extreme learning machine". *In Unifying Electrical Engineering and Electronics Engineering*. Springer, 2014, 1029–1035.

PUBLICATIONS BASED ON THIS THESIS

Journal Papers

1. **Rashmi P Shetty**, Sathyabhama A and Srinivasa Pai P.,Comparison of modeling methods for wind power prediction: A critical study.*Frontiers in Energy*, Springer, https://doi.org/10.1007/s11708-018-0553-3.

2. **Rashmi P Shetty**, Sathyabhama A and Srinivasa Pai P.,Efficient Modelling and simulation of wind power using online sequential learning algorithm for feed forward networks. *Journal of Mechanical Engineering Research and Development*, Zebline International Publications, Vol 42(1), 109-115, 2019.

3. **Rashmi P Shetty**, Sathyabhama A and Srinivasa Pai P., Wind power modeling and simulation : A comparison of feed forward neural networks. *International Journal of Advances in Soft Computing and Its Applications (IJASCA)* (Under review).

4. **Rashmi P Shetty**, Sathyabhama A and Srinivasa Pai P., An efficient OSELM model based on parameter optimization and feature selection using cuckoo search algorithm for multistep wind speed forecasting, *Soft computing*, Springer (Under review).

Conference Papers

1. **Rashmi P Shetty**., Sathyabhama A and Srinivasa Pai P., Prediction of wind turbine using ANN Prediction of wind turbine power using ANN. *WEENTECH Proceedings in Energy - Volume 4*, National Institute Of Technology , Patna , Bihar, March $4-6$ 2016, 50-55.

2. **Rashmi P Shetty**, A.Sathyabhama and Srinivasa Pai P., Wind turbine power optimization studies using Particle swarm optimization. *In Proceedings of Eighth National Conference on Advances in Energy Conversion Technologies - 2016*, Manipal Institute Of Technology, Manipal, Karnataka, January $28-30$ 2016, 1-5.

3. **Rashmi P Shetty**, Sathyabhama A, Srinivasa Pai P and A. Adarsh Rai., Optimized Radial Basis Function Neural Network model for wind power prediction. *In proceedings of Second International Conference on Cognitive Computing and Information Processing (CCIP)IEEE*, SJCE, Mysuru,Karnataka, August $12-13$ 2016, 1-6.

4. **Rashmi P Shetty**, Sathyabhama A and Srinivasa Pai P., Wind power optimization: A comparison of meta-heuristic algorithms, *In proceedings of International Conference on advances in Manufacturing Materials and Energy Engineering (ICon MMEE2018)*, MIT, Moodbidri, Karnataka, March $2-3$ 2018, Published in *Conf. Series: Materials science and Engineering*, DOI: 10.1088/1757-899X/376/1/012021.

5. **Rashmi P Shetty**, Sathyabhama A, Srinivasa Pai P and Ranjith Shetty K., Wind speed Forecasting in different seasons using ELM batch learning algorithm in Indian context, *In proceedings of International conference on Emerging Trends in Engineering*, NMAM, Institute of Technology, Nitte, Karnataka, May $14-15$ 2018, Published in *International Journal of Engineering and Technology*, Vol 7, 705-709.

CURRICULUM VITAE

RASHMI P SHETTY
'Kshithija' Kuthyar Post and village
Udupi District, Karnataka, India, 574504
Email: iprashmi@nitte.edu.in
Contact Details: +919448529222

PROFESSIONAL SUMMARY

She is currently working as Assistant Professor in the Department of Mechanical Engineering, N.M.A.M Institute of Technology, Nitte, India. She has 9 years of teaching experience and has guided 4 M.Tech and 9 B.E projects.

AWARDS RECEIVED

- Gold Medal in M.Tech Energy Systems Engineering, N.M.A.M Institute of Technology, Nitte, VTU, Belagavi, 2012.
- Best paper award, "Wind power optimization: A comparison of metaheuristic algorithms", International conference on Advances in manufacturing materials and energy engineering, MITE, Moodbidri,2018.

SKILLS

- MATLAB programming, C, C++
- AUTO CADD, SOLIDWORKS
- MINITAB 17, STATISTICA 12.0
- LATEX, ORIGIN Pro

WORKSHOP/FACULTY DEVELOPMENT PROGRAMS ATTENDED/ORGANIZED

- Attended a Workshop on Introduction to Therapeutic counselling for engineering Teachers, 13-16 July 2015, N.M.A.M Institute of Technology, Nitte.
- Attended a Faculty Development Program on Recent developments in measurement science, 22-23 April 2016, N.M.A.M Institute of Technology, Nitte

- Attended a Faculty Development Program on Writing an effective project proposal, 24 October 2016, N.M.A.M Institute of Technology, Nitte.
- Organizing committee member of Faculty development program on Theoretical and computational mechanics, 19-21 January, 2017, N.M.A.M Institute of Technology, Nitte.
- Organizing committee member of National conference on Machining of difficult to machine materials: Recent developments, issues and solutions, 3-5 August, 2017, N.M.A.M Institute of Technology, Nitte.
- Organizing committee member of Workshop on Teaching engineering and pedagogy for effective implementation of outcome based education, 16-21 July, 2018, N.M.A.M Institute of Technology, Nitte.

ACADEMIC RECORDS

- Doctoral Program: Wind Energy, Artificial Intelligence, National Institute of Technology Karnataka, Surathkal, 2019.
- M.Tech: Energy Systems Engineering, N.M.A.M Institute of Technology, Nitte/VTU, Belagavi, 2012.
- B.E: Industrial Production, N.M.A.M Institute of Technology, Nitte/VTU, Belagavi, 2004