Ensemble Time Series Forecasting with Applications in Renewable Energy

Ye REN

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of

Doctor of Philosophy

August 2015

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other University. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where specifically indicated in the text.

Ye REN
August 2015

Acknowledgements

I would like to take this opportunity to express my deep gratitude to my project supervisor Associate Professor P. N. Suganthan, who has given me invaluable advice and suggestions throughout my work towards the project.

I am also very grateful to Dr N. Srikanth for giving many useful suggestions on my project.

I would like to thank Economic Development Board of Singapore (EDB) for their academic and financial support on Clean Energy PhD Scholarship Program.

I would like to thank Energy Research Institute @ NTU (ERI@N) for Joint Industrial PhD Program (JIPP) management and opportunities for exchange and workshops.

I would like to thank the Singapore National Research Foundation (NRF) under its Campus for Research Excellence And Technological Enterprise (CREATE) programme, and Cambridge Advanced Research Centre in Energy Efficiency in Singapore (CARES), C4T project for resource and technical support.

Last but not least, I wish to give thanks to a number of my friends and colleagues for their support.

Summary

Renewable/sustainable energy sources draw increasing attention of researchers due to the shortage of fossil fuel. The fossil fuel also has adverse impact on our environment. However, because of the intermittent nature of the renewable energy sources, it is difficult to be integrated into the power grid. Accurate forecasting is required for improving the reliability of power dispatch, unit commitment, reducing the cost of energy storage devices, optimizing the operation and maintenance schedule. Besides time series forecasting, wind power ramp classification and detection is also covered in the thesis due to its importance for wind farm operation and power integration.

The first part of the thesis focuses on the development of ensemble forecasting methods related to renewable energy: wind speed and wind power and solar irradiance.

Firstly, state-of-the-art ensemble time series forecasting methods are reviewed and categorized into competitive ensemble forecasting and cooperative ensemble forecasting. The advantages and limitations of each category are discussed. Secondly, empirical mode decomposition (EMD) based time series forecasting methods are proposed and discussed. A support vector regression (SVR) takes the decomposed and re-constructed feature sets from EMD for short-term wind speed forecasting and the proposed model has outstanding performance. Next, the ensemble versions of EMD are investigated. An AdaBoost based EMD-artificial neural network (ABEMD-ANN) is proposed and the performance on wind speed forecasting outperformed the EMD-ANN. Another ensemble approach to EMD is noise assistance and among the various realizations. Complete ensemble EMD with adaptive noise (CEEMDAN) based SVR has the best performance among the ensemble EMD based forecasting methods on wind speed forecasting. The performance of ensemble EMD based SVR methods on solar irradiance forecasting are also superior to other state-of-the-art forecasting methods.

The second part of the thesis focuses on the analysis of random vector functional link (RVFL) neural network and its applications in wind speed and solar irradiance forecasting and wind power ramp forecasting. First the importance of direct input output connections in the RVFL network is studied based on the comparison of eight different RVFL configurations and two hierarchical ensemble forecasting methods on wind speed forecasting. Then the RVFL network is applied to classify the significant wind power ramp in the next 6 or 12 hours. The performance of RVFL network on power ramp classification outperformed other popularly used machine learning methods.

Finally, some future research directions are pointed out.

List of Publications

[1] **Y. Ren**, P. N. Suganthan, "Empirical comparison of bagging-based ensemble classifiers," in: *Proc. International Conference on Information Fusion (Fusion'12)*, Singapore, Jul. 2012, pp. 917–924.

[2] **Y. Ren**, P. N. Suganthan, "A kernel-ensemble bagging support vector machine," in: *Proc. International Conference on Intelligent Systems Design and Applications (ISDA'12)*, Brunei, Oct. 2012.

[3] **Y. Ren**, P. N. Suganthan, N. Srikanth, "Wind speed forecasting: a comparison between statistical approach and learning-based approach," in: *Smart Microgrids, New Advances, Challenges and Opportunities in the Actual Power Systems*, US:NOVA, 2013.

[4] **Y. Ren**, L. Zhang, P. N. Suganthan, "K-nearest neighbor based bagging SVM pruning," in: *Proc. IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL'13)*, Singapore, Apr. 2013.

[5] L. Zhang, **Y. Ren**, P. N. Suganthan, "Instance based random forest with rotated feature space," in: *Proc. IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL'13)*, Singapore, Apr. 2013.

[6] **Y. Ren**, P. N. Suganthan, N. Srikanth, "A hybrid ARIMA-DENFIS method for wind speed forecasting," in: *Proc. IEEE International Conference on Fuzzy Systems (Fuzz'13)*, India, Jul. 2013.

[7] **Y. Ren**, P. N. Suganthan, and N. Srikanth, "A comparative study of empirical modedecomposition-based short-term wind speed forecasting methods," *IEEE Trans. Sustain.Energy*, vol. 6, no. 1, pp. 236–244, Jan. 2015.

[8] **Y. Ren** and P. N. Suganthan, "EMD-kNN models for wind speed forecasting," *Journal of Power and Energy Engineering*, vol. 2, no. 4, pp. 176–185, 2014.

[9] **Y. Ren**, P. N. Suganthan, and N. Srikanth, "A novel empirical mode decomposition with support vector regression for windspeed forecasting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. PP, 2015, doi: 10.1109/TNNLS.2014.2351391.

[10] **Y. Ren**, X. Qiu, and P. N. Suganthan, "EMD based adaboost-BPNN method for wind-speed forecasting," in *Proc. IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL'14)*, Orlando, US, Dec. 2014.

[11] L. Zhang, **Y. Ren**, P. N. Suganthan, "Towards Generating Random Forests via Extremely Randomized Trees," in *Proc. IEEE World Congress on Computational Intelligence (WCCI'14)*, Beijing, Jul. 2014.

[12] X. Qiu, L. Zhang, **Y. Ren**, P. N. Suganthan and G. Amaratunga, "Ensemble Deep Learning for Regression and TimeSeries Forecasting," in *Proc. IEEE Symposium on Computational Intelligence and En-semble Learning (CIEL'14)*, Orlando, US, Dec. 2014.

[13] D. Laha, **Y. Ren** and P. N. Suganthan, "Modeling of Steelmaking Process with Effective Machine Learning Techniques, " *Expert Systems With Applications*, vol. 42, pp. 4687-4696, 2015.

[14] **Y. Ren**, L. Zhang, and P. N. Suganthan, "Ensemble Classification and Regression – Recent Developments, Applications and Future Directions," *IEEE Comput. Intell. Mag.*, 2016, doi: 10.1109/MCI.2015.2471235.

[15] **Y. Ren**, P. N. Suganthan, N. Srikanth and G. Amaratunga, "Single Hidden Layer Neural Networks with Random Weights for Short-term Electricity Load Demand Forecasting," *Information Sciences, Elsevier*, 2015, doi: 10.1016/j.ins.2015.11.039.

[16] **Y. Ren**, P. N. Suganthan, N. Srikanth, "Ensemble methods for wind and solar power forecasting – a state-of-the-art review," *Renewable and Sustainable Energy Reviews, Elsevier*, vol.50, pp. 82-91, 2015.

[17] **Y. Ren**, X. H. Qiu, P. N. Suganthan, N. Srikanth and G. Amaratunga, "Detecting Wind Power Ramp with Random Vector Functional Link (RVFL) Network," in: *Proc. IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL'15)*, Cape Town, South Africa, Dec. 2015.

Table of Contents

List of Figures

List of Tables

Nomenclature

(n)MAE	(normalized) mean absolute error
(n)RMSE	(normalized) root mean square error
(s)MAPE	(symmetric) mean absolute percentage error
ACF	autocorrelation function
AdaBoost	adaptive boosting
AIC	Akaiki information criterion
ANFIS	adaptive neuro-fuzzy inference system
ANN	artificial neural network
ARIMA	autoregressive integrated moving average
ARMA	autoregressive moving average
ARTMAP	predictive adaptive resonance theory
Bagging	bootstrap aggregation
BIC	Baysian information criterion
BP	backpropagation
CSP	concentrated solar power
CVNN	complex-valued neural network
DENFIS	dynamic evolving neural-fuzzy inference system
DWT	discrete wavelet transform
ECMWF	European center for medium-range weather forecasting
EMD	empirical mode decomposition
EnKF	ensemble Kalman filter
FIR	finite impulse response
GARCH	generalized autoregressive conditional heteroscedasticity
GF	grey forecast
GHI	global horizontal irradiation
kNN	k-nearest neighbors

LEP	Lyapunov exponent prediction
LS-SVR	least square support vector regression
MASE	mean absolute scaled error
MLP	multi-layer perceptron
MRE	mean relative error
MSE	mean square error
NDBC	national data buoy center
NECP	national center for environmental prediction
NREL	national renewable energy laboratory
NSRDB	national solar radiation data base
NWP	numerical weather prediction
PACF	partial autocorrelation function
PDF	probability density function
PRNN	pipelined recurrent neural network
PSO	particle swarm optimization
PV	photovoltaic
RF	random forest
RNN	recurrent neural network
RVFL	random vector functional link
RVNN	real-valued neural network
SMA	seasonal moving average
SMO	sequential minimization optimization
SVC	support vector classification
SVD	singular value decomposition
SVM	support vector machine
SVR	support vector regression
TS	time series
TSI	total satellite image

Chapter 1

Introduction

1.1 Motivation

Renewable energy sources, such as wind energy, solar energy, wave energy and tidal energy, draw increasing attention of researchers due to the shortage and their adverse impacts on the environment of fossil fuel. Renewable energy is abundant and environmentally-friendly and is a potentially best alternative to fossil fuel in the near future. Although in the western European countries, China and USA, researchers keep on improving the capability of wind turbines [1] and efficiency of solar panels [2], the cost of using renewable energy is still high because it is difficult to be integrated into the power grid either temporally or spatially.

The main reason of the difficulty is that the renewable energy sources are unpredictable. For example, wind power is naturally intermittent, and furthermore for a wind farm, wind turbines affect the wind intake of each other, deteriorating the overall power output of the wind farm [3]. Solar irradiance is affected by cloud cover and haze effect [4].

The unpredictability causes a large fluctuation in the power outputs and in order to smooth the fluctuation, large amount of battery storage or costly power reserve is required. By improving the forecasting accuracy, these reserves can be reduced. In addition, accurate forecasting can improve the energy conversion efficiency, reduce the risk caused by system overloading and extreme weather conditions, and can improve the unit commitment optimization [5–7].

1.2 Background

There are numerous definitions for forecasting and the following paragraph quotes a typical definition:

"Forecasting is the process of making statements based on past events about events whose actual outcomes have not yet been observed [8]."

The history of forecasting is very long. In the early stage, forecasting is based on intuitive observation and empirical experiences. When statistics is employed in forecasting, the development of forecasting methods speeds up and diverges into different categories. Exponential smoothing and the improved versions have good practical record in finance, economics, etc. Auto regressive moving average family also became main stream forecasting methods in certain period of the history.

Fast development in computing, Internet and satellite technology helps the development of meteorology forecasting based on physical data. Differential equation with strong computing power and fast data acquisition improves the portability and accuracy of meteorology forecasting.

Computational intelligence and machine learning uncovers another possibility of research in forecasting. Neural networks, support vector machines extends the view of forecasting problems from linear to non-linear and from analytic to black box. Optimization algorithms such as evolutionary algorithms speed up the parameter tuning for the machine learning methods.

Ensemble of different forecasting methods further improves the performance of forecasting. The ensemble forecasting methods will be discussed in details in Chapter 3.

In general, a forecasting system is a supervised learning [9] which learns from known observations and predict the unknown future values. It can be partitioned into two phases: training (or modelling) phase and testing (or predicting) phase. In the training phase, the known data and their response values are used to form a forecast model (or a collection of models). Then in the testing phase, the trained forecast models are applied to unknown data to obtain the predicted values. Based on the characteristics of the response values, forecasting can be further divided into classification and regression: if the response is a discrete number or a categorical value, then it is classification and if the response is a continuous value, then it is regression. Tables 1.1 summarizes the characteristics of classification and regression. Table 1.2 gives some examples for classification and regression applied to renewable energy.

Table 1.1 Characteristics of Classification and Regression

	Classification	Regression
Equation	$\hat{Y} = \mathrm{f}(\mathbf{X}, \theta)$	
Input	Features **X**, Parameters θ	
Output	Categorical Data	Ordinal, Numerical Data
Evaluation	ACC, Sensitivity, Specificity, etc.	MAE, MAPE, RMSE, etc.

Table 1.2 Examples of Classification and Regression Applications in Renewable Energy

Classification	Regression
Wind Power Ramp, Gearbox Vibration, Bearing Fault, etc.	Wind Speed, Solar Irradiance, Load Demand, etc.

1.3 Objectives

The objectives of this PhD project are to (i) review the state of the art ensemble time series forecasting methods in the context of wind and solar energy; (ii) investigate the empirical mode decomposition based ensemble forecasting methods; (iii) develop and compare the improved versions of empirical mode decomposition with boosting and noise assisted ensemble; (iv) analyse the effectiveness of random vector functional link network on time series forecasting and (v) on wind power ramp forecasting.

1.4 Scope

There are mainly two research topics presented in this thesis: empirical mode decomposition based time series forecasting and random vector functional link network based forecasting and classification. Other methods are not discussed in details in this thesis.

Wind speed/power and solar irradiance time series data are used in the projects because they are the two commonly used renewable energy sources and a lot of on going researches are focusing on them. Other renewable energy sources are not discussed in this thesis. For wind power ramp forecasting, only two definitions of wind power ramp are employed in the thesis and other definitions are not included because the two definitions are most widely used.

1.5 Major Contributions of the Thesis

In this thesis, three contributions pertaining to the renewable energy time series forecasting are reported.

The first contribution pertains to ensemble forecasting. In the early stage of research, some research on ensemble methods with available public classification datasets from the UCI repository [10] to evaluate four bootstrap aggregation (Bagging) based ensemble methods were reported in [11]. Then a kernel ensemble bagging support vector machine (SVM) that had better ensemble strategy and higher accuracy was implemented in [12]. Then the research focus moves to the ensemble time series (TS) forecasting, and a hierarchical autoregressive integrated moving average (ARIMA)-dynamic evolving neural-fuzzy inference system (DENFIS) for TS forecasting was reported [13] and followed by a bagging based forecasting model [14]. A survey on ensemble TS forecasting and a review on ensemble classification and regression were reported in [15] and [16], respectively.

The survey [15] categorized the ensemble TS forecasting methods into competitive and cooperative ensemble forecasting approaches and the recently published papers were reviewed. Some popular ensemble forecasting methods were compared on wind speed and solar irradiance forecasting tasks. The advantage and limitations of each categories were stated. An expanded version of the literature review on ensemble classification and regression was reported in [16]. In the review, ensemble TS forecasting was further reviewed by including multi-objective optimization based ensemble, fuzzy ensemble and deep learning based ensemble.

In view of the state of the art ensemble forecast methods discussed in the survey, there is a method called empirical mode decomposition (EMD) which is used for TS forecasting. However, studies on EMD did not get attentions by the researchers in the recent years. In this thesis, EMD based forecasting methods and possible improvements on EMD with boosting and noise assisted ensemble were introduced, which forms the second contribution.

In the second contribution, EMD method was extensively studied. EMD-k nearest neighbor (kNN) [17] and EMD-support vector regression (SVR) [18] were reported for wind speed forecasting. The EMD-kNN method focused on the parameter tuning of the kNN for forecasting, including the optimal number of neighbors, the decay function of the feature vector and the weighted distance function of the neighbors. The proposed EMD-SVR method employed EMD as a feature extractor. Based on the extracted features from the original wind speed TS, the forecasting accuracy can be maintained and the computation time can be reduced.

Ensemble versions of EMD were also investigated. Ensemble EMD-artificial neural network (ANN) [19] and EMD-adaptive boosting (AdaBoost)-ANN [20] were reported. The addition of ensemble structures were proved to enhance the EMD decomposition and improved the performance on TS forecasting.

The studies on EMD unveiled the feasibility for TS forecasting, especially on wind speed and solar irradiance. However, it also showed the limitations of EMD based ANN on TS forecasting, which is not emphasised in the literature. An EMD-SVR with input vector reconstruction is a novel method aiming for fast computing time and relatively high accuracy. For ensemble of EMD, there exist several versions but no comparisons exist in the literature. In this thesis, a boosting based EMD was proposed to improve the performance of EMD-ANN and a extensive comparisons of different noise assisted ensemble EMD-SVR were reported.

The third contribution is the research on random vector functional link (RVFL) network. The RVFL network has a unique weight assignment and simple least square training that can reduce the training time significantly compared with conventional ANN with backpropagation (BP). It also has direct input output connections that retain some advantage from linear models for TS forecasting. A comparison on different variations of RVFL network for TS forecasting was reported in [21] and an RVFL network for wind power ramp forecasting was studied as well. The advantages in terms of the computation time and the accuracy were shown in the performance analysis.

In this thesis, comparisons on RVFL network with different configurations (input layer bias, hidden layer bias and direct input output connections) were conducted, which has not done before. The comparisons gave a clear conclusion with statistical results for the researchers to apply to their tasks. The effectiveness of RVFL network was tested by wind and solar TS data as well.

In addition, RVFL network was applied for classification task in this thesis. The topic covered an important application: wind power ramp forecasting. Dealing with imbalanced data, missing data, noise and selection of error metrics were addressed in the thesis as well. The performance of RVFL network were compared with state of the art classifiers.

1.6 Organization of the Thesis

The remaining of the thesis is organized as follows: Chapter 2 elaborates the details of wind and solar energy sources; Chapter 3 details the ensemble TS forecasting methods; Chapter 4 describes an EMD-SVR method for TS forecasting; Chapter 5 expands the EMD based forecasting methods with ensemble structure; Chapter 6 depicts some RVFL network based forecasting methods and an RVFL network based wind power ramp forecasting method is discussed in Chapter 7. Finally, Chapter 8 concludes the thesis and recommends some future research directions.

Chapter 2

Background and Characteristics of Wind and Solar Energy

This chapter gives a brief introduction to two renewable energy sources: wind and solar. The importance and challenges of wind speed and solar irradiance forecasting are addressed. The forecasting types are categorized according to the time horizon and approaches. The characteristics of wind speed/power and solar irradiance TS are discussed.

2.1 Wind Energy

Wind is bulk, directed movement of air. It is one of the common renewable energy sources in the world. Ancestors have already used wind as energy sources for milling and sailing. Modern development has converted wind energy to electricity by propelling onto blades of a turbine, rotating a generator, producing electricity. Wind farms consisting of numerous wind turbines have been built and operated all over the world, especially in Europe, North America and Eastern Asia. Offshore wind farms are also drawing lots of attention of the researchers.

The most direct measure of energy inside wind is wind speed, and the most common way to record down the wind speed is wind speed TS. Typical wind speed TS is non-linear and not normally distributed, and the periodic characteristic of wind speed TS is not obvious unless the site has a strong land/sea breeze (diurnal cycle) or trade wind (annual cycle).

The theoretical conversion from wind speed to wind power is shown as [5]:

$$P = \frac{1}{2}\rho_a A_t C_p(\lambda, \beta) v^3 \tag{2.1}$$

where ρ_a is the air density, A_t is the area of the turbine when rotating, $C_p(\lambda,\beta)$ is the efficiency which is affected by two parameters: tip speed ratio λ and blade pitch β, and v is the up-wind speed. If the wind turbine is placed in a wind farm, wake effect should also be considered.

However, a practical wind turbine only has a small operation region that follows equation (2.1). An empirical wind power v.s. speed scatter plot of a wind turbine is shown in Figure 2.1, in which the data is retrieved from National Renewable Energy Laboratory (NREL) [22].

It is shown that there are two regions where the wind power is zero: (i) cut in region when the wind speed is less than the cut in speed v_{in} and (ii) cut out region when the wind speed exceeds the cut out speed v_{out}. When the wind speed is less than v_{in}, the wind turbine cannot gain enough power to rotate thus there is no power output and on the other hand, when the wind speed is larger than v_{out}, the wind turbine has to be halted in order to protect the turbine from over speeding.

There are another two regions where the wind power is non zero: (iii) maximum-output region and (iv) cubic region. The maximum-output region outputs a constant power irrespective of the wind speed due to the maximum output capability of the generator in the turbine. The cubic region shows a cubic power relation between wind speed and wind power as shown in equation (2.1).

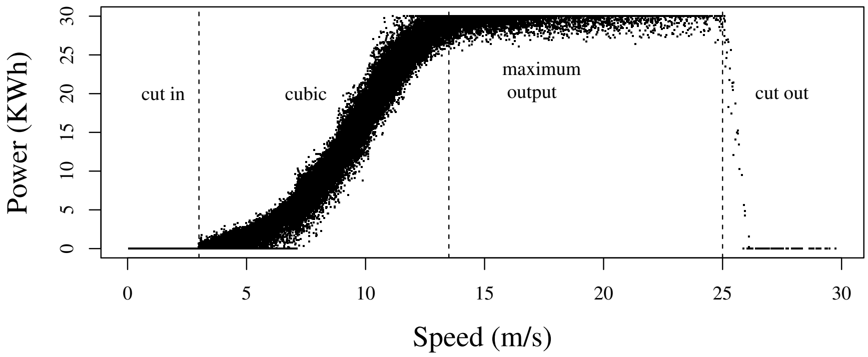


Figure 2.1 Wind Speed to Wind Power Mapping.

In [23, 24], wind power forecasting and wind speed forecasting are considered equivalent if a proper wind speed to wind power conversion such as equation (2.1) and Figure 2.1 are available. Wind speed forecasting is more favorable because the wind speed data is easier to retrieve and it can be obtained without an existing wind turbine/farm [23, 24].

Several wind speed/power forecasting review/survey papers are presented in the literature [5, 23, 25–28]. In the reported literature, wind speed/power forecasting can be categorized according to (i) approaches and (ii) forecasting time horizons. There are mainly three approaches for wind speed/power forecasting: physical approach, statistical approach and learning-based approach. There are mainly four forecasting time horizons: very short term, short term, medium term and long term. The definitions of these two different categorizing schemes are summarized in Tables 2.1 and 2.2. However, to the best knowledge of the author, there is no survey and experimental comparison paper of ensemble methods for renewable energy forecasting.

Table 2.1 Wind Speed/Power Forecasting Categorization based on Approach.

Approach	**Input**	**Examples**
Physical [5, 23, 25, 27]	Meteorological Data	NWP (ECMWF, NCEP)
Statistical [5, 23, 25, 27]	Historical Data	ARIMA, Kalman Filter
Learning-based [23]	Historical Data	ANN, FNN, SVM
Hybrid [5, 25]	Historical Data	ARIMA-ANN/SVM
Spatial Correlation [27]	Historical & Geographical Data	ANN, FNN

Table 2.2 Wind Speed/Power Forecasting Categorization based on Forecasting Time Horizon [5].

Type	**Horizon**
Very Short Term	$<$ 30-min
Short Term	0.5 $\sim$ 6-hr
Medium Term	6 $\sim$ 24-hr
Long Term	1 $\sim$ 7-day

2.2 Solar Energy

In the astronomical point of view, sun is a star at the center of the solar system. All planet including our earth is orbiting around it. In a broad sense, solar energy creates everything on the earth including fossil energy sources. Gasoline, natural gas and coal mine are all converted from animals or forests that were directly or indirectly originated from solar energy via photosynthesis. All other forms of renewable sources are created by solar energy: wind is blowing because of temperature difference, tidal energy is due to moon-earth orbit, etc. In a narrow sense, solar power is the one that can be easily converted to electricity. There are mainly two kinds of conversions: photovoltaic (PV) that converts solar power directly to electricity, and concentrated solar power (CSP) that first converts solar power to heat and then converts to electricity by a gas turbine generator. Despite the variations of conversion, solar irradiance index G is critical for solar power. Solar irradiance is the amount of solar energy received on a surface per unit time per unit area on the earth. It is affected by the location, time and cloud/haze cover. An analytical formula is shown as [29]:

$$\begin{aligned} G &= \alpha G_0(1+0.033\cos\frac{360d}{365})\sin\theta_s \\ \sin\theta_s &= \cos h\cos\delta\cos\phi+\sin\delta\sin\phi \end{aligned} \tag{2.2}$$

where θ_s is the solar elevation angle, h is the solar hour angle, δ is the sun declination, ϕ is the local latitude, d is the date sequence of a year, α is the cloud/haze cover index and G_0 is solar irradiance constant which is recorded as 1367 W/m^2 by World Meteorological Organization [29].

Solar irradiance can be divided into three categories: direct, diffuse and global. Direct irradiance is the solar radiance pans through directly to the earth's surface. Diffuse irradiance is the scattered solar radiance out of the direct beam. Global irradiance is the sum of the above two [30]. Equation (2.2) is for the global irradiance or Global Horizontal Irradiance (GHI) [31].

Similar to wind speed/power forecasting, solar irradiance forecasting can also be categorized according to the approaches [31, 32] as shown in Table 2.3.

Table 2.3 Solar Irradiance Forecasting Categorization based on Approach [31, 32].

Approach	Input	Examples
Physical	Meteorological Data	NWP, TSI
Statistical	Historical Data	ARIMA, Exponential Smoothing
Learning-based	Historical Data	ANN, SVR

In [33], solar irradiance was categorized based on the forecasting time horizon, and the authors also proposed suitable approaches for different time horizons. The categories and the proposed approaches are presented in Table 2.4.

Table 2.4 Solar Irradiance Forecasting Categorization based on Forecasting Time Horizon [33].

Type	Horizon	Example
Very Short Term	$0.5 \sim 6$ hour	TSI
Medium Term	$6 \sim 48$ hour	NWP

2.3 Characteristics of Time Series

A TS is a sequence of values successively measured over a time interval [34, 35]. If the sequence of values is one dimensional, the TS is uni-variate but if the sequence of values is more than one dimension, the TS is multi-variate. For example, a wind speed TS is uni-variate but a wind speed with its associated wind direction is multi-variate.

In statistics, a convenient way to summarize a dataset is to use five numbers (minimum, 25% quantile, median, 75% quantile and maximum) but for visualization purpose, histogram or probability density function (PDF) is usually preferred. Histogram is an empirical and discrete way to estimate the probability of a value in a dataset whereas PDF is a formal and continuous way to describe the distribution. A commonly used method to estimate the PDF for a given dataset is called kernel density estimation [36].

As TS has an extra time information, and some statistical measures for TS may be also time dependent such as mean, variance and so on. For example, white noise is a TS that has constant mean and variance whereas random walk is a TS with varying mean.

An important characteristic is called 'stationarity'. A TS is said to be stationary if its statistical characteristics are independent of the time when it is observed [37]. However, this requirement is difficult to meet, so a weakly stationary TS is mostly used to define the stationarity. In the thesis, a stationary TS by default refers to a weakly stationary TS.

A stationary TS has two properties: (i) the mean of the TS is constant and does not depend on time, and (ii) the autocorrelation function (ACF) depends on s and t only through their difference $|s-t|$. A simple way to tell whether a TS is stationary is to check whether the ACF decays rapidly.

ACF is the ratio of the auto covariance function and the two self-covariances [37]. The auto covariance function is defined as:

$$\gamma(s,t) = \mathrm{E}[(X_s - \mu_s)(X_t - \mu_t)] \tag{2.3}$$

Therefore, the ACF is:

$$\rho(s,t) = \frac{\gamma(s,t)}{\sqrt{\gamma(s,s)\gamma(t,t)}} = \frac{\gamma(s,t)}{|\sigma_s \sigma_t|} \tag{2.4}$$

where μ is the sample mean and σ is the sample standard deviation.

However, ACF shows the correlation between data at time s and t with all the data between time points s and t considered, i.e. ACF has a propagation effect from time s to time t. For some applications, this propagation effect will not be relevant but the sole correlation at time s and t is the only focus. Partial autocorrelation function (PACF) shows the correlation between s and t without considering in-between time points [37]. The PACF is usually used for determining the order of an auto-regressive (AR) model.

A more precise way to determine the stationarity is called unit root test [38, 39]. Unit root test is a statistical method to check the hypothesis that whether a TS is stationary or not. If the TS is non-stationary but the first difference of the TS is stationary, then a unit root is present. This unit root test can be applied iteratively to find higher differencing order. There are several implementations of unit root test such as: (Augmented) Dickey-Fuller test [40, 41], Phillips-Perron test [42], Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test [43] and so on.

For non-stationary TS, there may exist a linear or non-linear trend [37]. A trend is a additive term to the TS which usually slowly varying with respect to time. It is common to observe repeated pattern in a TS, this pattern is called seasonality [37]. Seasonality can be either additive or multiplicative. In certain TS, more than one seasonal components with different frequency bandwidth can be observed.

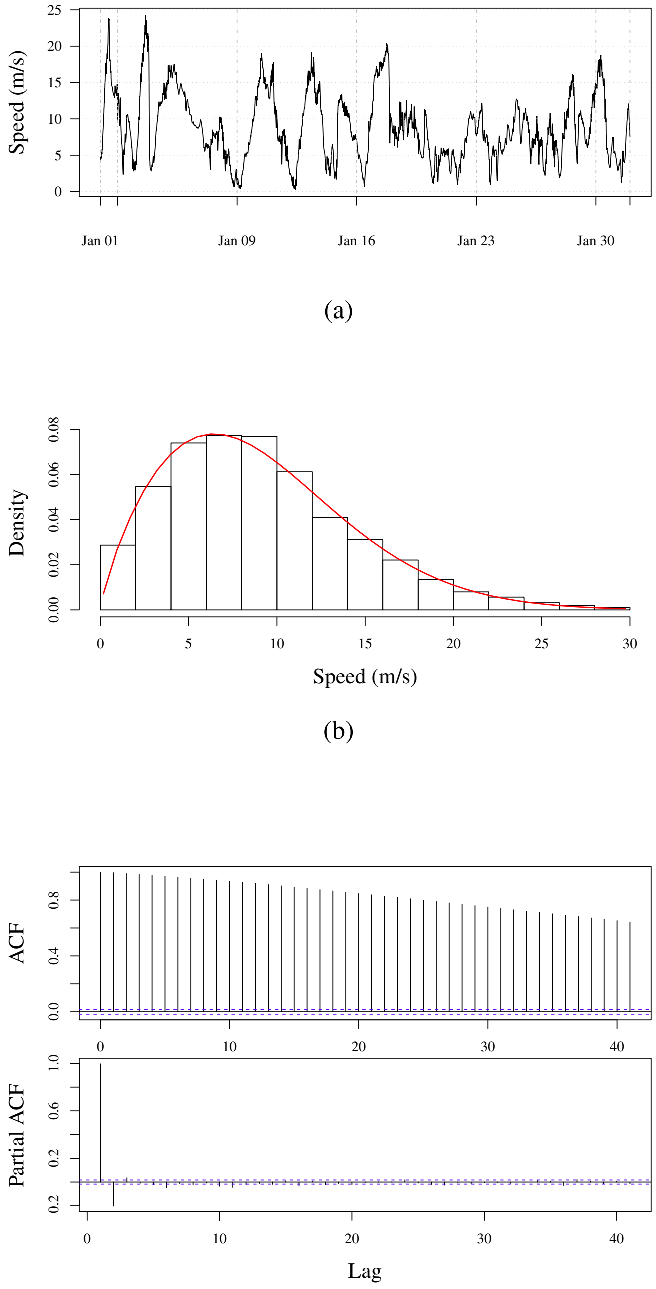
2.4 Characteristics of Wind Speed/Power Time Series

Wind speed is usually recorded by an anemometer and the data is usually averaged over 10 minute interval in order to smooth out the fluctuations due to wind gusts. The wind speed TS is not normally distributed but instead the histogram and the power density estimation shows a Weibull distribution for most wind speed TS. A Weibull distribution has a PDF as [37]:

$$f(x;\lambda,k) = \begin{cases} \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} & x \geq 0, \\ 0 & x < 0 \end{cases} \tag{2.5}$$

where k and λ are parameters to control the shape of the PDF.

A wind speed TS is retrieved from NREL site 12896 [22] during January 2006 and is plotted in Figure 2.2a. As shown, the TS does not show a cyclic characteristic and there is no low frequency trend observed either. The estimated PDF (red smooth curve) via kernel density estimation is superimposed on the histogram (black boxplot) of the TS in Figure 2.2b. The estimated PDF does not show a normal distribution but shows a Weibull distribution. The estimated parameters of the Weibull distribution for this particular wind speed TS are: $k = 10.255$ and $\lambda = 1.777$. The ACF and PACF are plotted in Figure 2.2c. The ACF does not decay rapidly but the PACF does and therefore we may model it with auto-regressive (AR) model [38]. In addition, there is no periodic characteristics shown in the ACF plot and thus we can strongly conclude that there is no seasonal component in this particular wind speed TS.



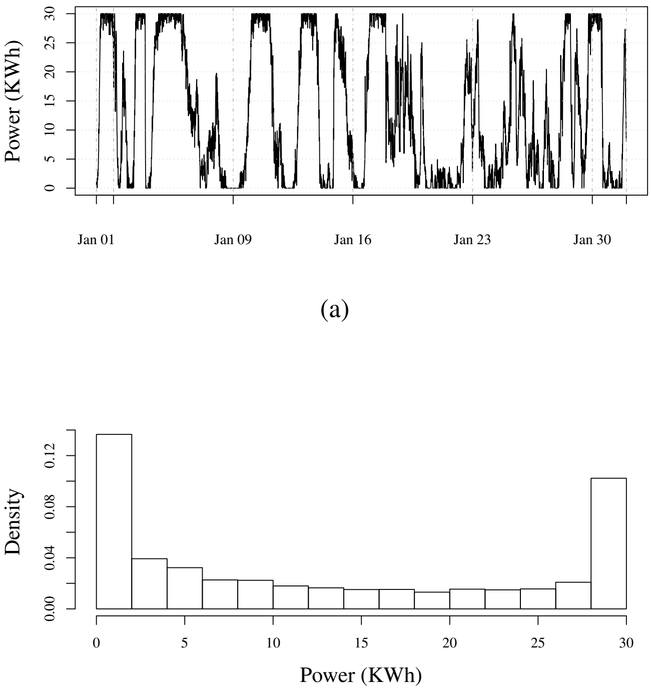
(a)

(b)

(c)

Figure 2.2 Plots of Wind Speed TS: (a) Plot of wind speed TS of NREL site 12896 during January 2006, (b) ACF and PACF plots of the wind speed TS and (c) Histogram and the power density estimation of the wind speed TS.

The wind power TS during January 2006 is plotted in Figure 2.3a and the histogram is shown in Figure 2.3b. We can see that due to the cut in, cut out and saturation regions in the wind speed to power conversion, the wind power TS has a lot of sharp fluctuation (ramps) and the histogram is bipartite (non parametric), introducing more difficulties in forecasting.



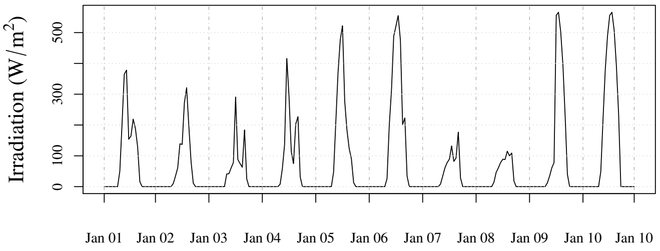
(a)

(b)

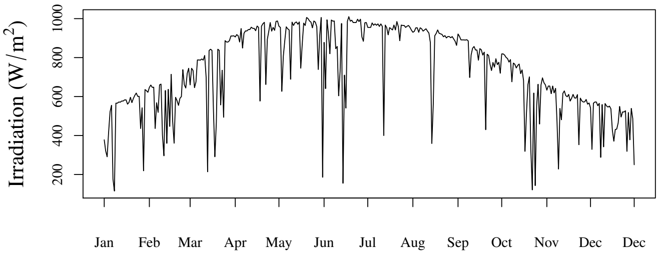
Figure 2.3 Plots of Wind Power TS: (a) Plot of wind power TS of NREL site 12896 during January 2006 and (b) Histogram of the wind power TS.

Wind speed plotted in Figure 2.2a shows neither an obvious trend nor a seasonal component. However, for wind speed recorded near tropical regions, there is a strong annual seasonal pattern due to monsoon. For example, Singapore usually has strong wind during December to early March due to northeast monsoon and during June to September due to southwest monsoon [44]. There also exist a strong diurnal seasonal pattern in Singapore due to land sea breeze.

2.5 Characteristics of Solar Irradiance Time Series



(a)



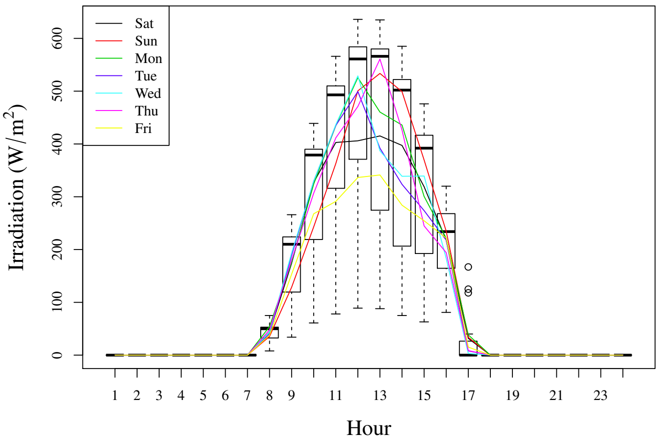
(b)

A direct solar irradiance TS is retrieved from national solar radiation database (NSRDB) site 69160 [45] during 2005 and a fraction of the TS is plotted in Figure 2.4a. As discussed in the previous paragraphs, solar irradiance is a composition of deterministic and stochastic factors. The deterministic factor reflected into the solar irradiance TS is the combination of a daily cycle and an annual cycle. The stochastic factors that affect the solar irradiance TS are cloud cover, haze cover, etc. These factors introduce the main challenge in solar irradiance forecasting.

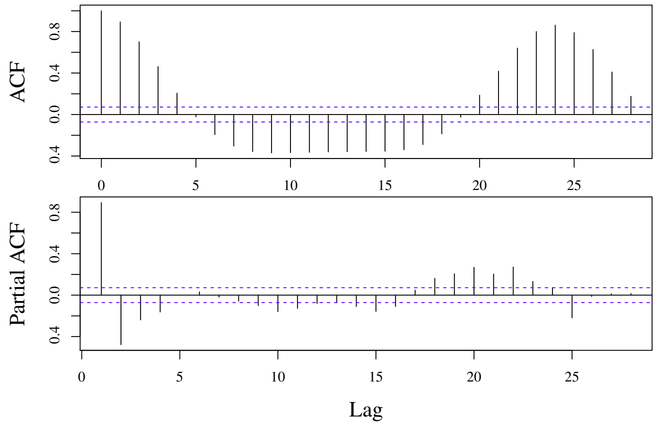
From Figure 2.4a. we can see the obvious daily cycles and the stochastic fluctuations. The daily cycles are separated by zero irradiance during night time when there is no sun light. These zero irradiance data make the TS non-stationary and non-linear. The daily cycles can be visualized on the ACF and PACF plots (Figure 2.3d) as well. From the hourly boxplot shown in Figure 2.3c, we can observe the fluctuations due to the stochastic components.

The deterministic annual cycle can be observed in Figure 2.4b where the daily maximum solar irradiance is plotted with respect to time. We can see that the trend is strongly correlated to the solar elevation angle.

In tropical regions such as Singapore, the seasonal component in solar irradiance is additive because the sun rise and sun down time are almost consistent every day. However, in other regions, especially high latitude regions, the seasonal components are multiplicative, and we can observe longer and stronger solar irradiance during summer whereas shorter and weaker solar irradiance during winter.



(c)



(d)

Figure 2.2 Plots of Solar Irradiance TS: (a) Plot of a fraction of solar irradiance TS of NSRDB site 69160 in January 2005, (b) Plot of the solar irradiance TS during 2005, (c) Hourly boxplot of the solar irradiance TS during January 2005, and (d) ACF and PACF plots of the solar irradiance TS.

Chapter 3

Literature Review of Ensemble Time Series Forecasting

TS forecasting is an important research topic in renewable energy and power systems. It is also very well studied in other disciplines such as finance, manufacturing and engineering, etc. The characteristics of TS is discussed in the previous chapter and example wind and solar TS are discussed. In this chapter, ensemble TS forecasting is discussed. Ensemble TS forecasting is categorized into two categories: competitive ensemble forecasting and cooperative ensemble forecasting. Methods for wind and solar TS forecasting are summarized in this chapter. Strengths and weaknesses of these methods are also addressed. Before introducing ensemble TS forecasting, various conventional TS forecasting methods are briefly discussed.

3.1 Conventional Time Series Forecasting

There are a lot of conventional methods for TS forecasting and most of them can be applied to different renewable energy forecasting applications. The categorizations based on approaches are discussed in Chapter 2. Take wind speed/power forecasting as an example, the methods can be categorized into three approaches: physical approach, statistical approach and learning-based approach [23]. Solar irradiance forecasting can be categorized similarly.

Physical approach uses the meteorological data obtained by physical sensors or satellites to forecast the wind speed of a particular region. A numerical weather prediction (NWP) model is widely applied in the real wind and solar power industry. It fits the real-time measured data into mathematical models based on fluid dynamics, thermo dynamics, etc. and use high performance computers (usually supercomputers) to simulate. NWP has high accuracy due to its advanced hardware configuration and data acquisition system but the trade-off is the high hardware cost and long computation time.

Statistical approach employs the historical data to predict the future data. Additional information such as temperature, pressure and wind direction, etc., can also be included as the statistical model's inputs. Some well developed statistical methods are auto regressive moving average (ARMA) and exponential smoothing, etc. The advantage of statistical approaches is that it requires less computational power and less resources. However, the disadvantage is that it is linear thus not accurate especially for long-term forecast. The statistical approach is usually applied for very-short-term forecast.

Learning-based approach is advantageous over statistical approach because of their nonlinear characteristics. To design a computational intelligence learning-based predictor, there are two phases: (i) supervised learning, and (ii) evaluation on unknown dataset. The most widely used approaches are artificial neural networks (ANN), decision trees, support vector regression (SVR), etc.

Since physical approach requires different domain knowledge and most of the models are patented, it is beyond the scope of this thesis. In the following sections, some typical methods in statistical approach and learning based approach are introduced.

3.1.1 Persistent Model

Persistent model is the simplest model. It assumes the future wind speed is the same as the current speed as shown in equation (3.1). Persistent model is usually used as a benchmark of other approaches.

$$x(t+h) = x(t) \tag{3.1}$$

where $x(t)$ is the TS data at time t and h is the forecasting horizon.

3.1.2 Exponential Smoothing

Exponential smoothing [46] is to model the TS recurrently with a smoothing parameter α which serves as a set of exponentially decreasing weights over time.

$$\hat{x}(t+1) = \alpha x(t) + \alpha(1-\alpha)x(t-1) \tag{3.2}$$

where $\alpha \in (0,1)$.

However, the above mentioned exponential smoothing does not model well on TS with trends or seasonal components. A more generalized Holt-Winters exponential smoothing model [46] was reported to cope with the trend and seasonal component. The details are well documented in [37, 46, 47].

Exponential smoothing is well studied in [48]. A few improved versions of exponential smoothing were employed to forecast one-day ahead electricity load demand. The proposed singular value decomposition (SVD) based exponential smoothing had overall best performance.

3.1.3 Autoregressive Integrated Moving Average

ARIMA is a popular TS modelling and forecasting tool [24, 35, 37, 49] that can be used for TS forecasting. ARIMA consists of three parts: autoregressive (AR), difference (I) and moving average (MA). A generic formula for ARIMA(p,d,q) is shown:

$$\nabla^d x(t) = \underbrace{\sum_{i=1}^{p} \phi_i \nabla^d x(t-i)}_{\text{AR Term}} + w_t + \underbrace{\sum_{j=1}^{q} \theta_j w(t-j)}_{\text{MA Term}} \tag{3.3}$$

where x is the TS, ϕ_i are the AR coefficients, θ_j are the MA coefficients and w is a Gaussian white noise with 0 mean and σ_w^2 variance. The operator ∇^d is to difference the TS data such that:

$$\nabla^d x = [x(t) - x(t-1)] - [x(t-1) - x(t-2)] - \cdots - [x(t-d+1) - x(t-d)] \tag{3.4}$$

If the TS has seasonal components such as solar irradiance TS which has a strong daily cycle ($s = 24$ hour), then seasonal ARIMA (sARIMA) is more suitable [39, 47].

$sARIMA(p,d,q)(P,D,Q)[s]$ is an extension of ARIMA which takes historical data with seasonal lag into consideration. Stationarity is achieved by differencing at lag s (seasonal period). P and Q are the seasonal AR and MA orders which is similar as p and q but with s multiplied. Also, it considers the seasonal difference D as well. The following parameter estimation procedures for ARIMA can be applied to sARIMA in a similar manner.

There are three orders to define an ARIMA model: p, d and q. The first step is to determine the differencing order d. Unit root test is a common way to estimate the value [38, 39].

After unit root test and differencing if necessary, the next step is to determine p and q. In order to determine the orders p and q, ACF and PACF should be calculated [37, 49].

A qualitative order determination method is by inspecting the ACF and PACF. Firstly by inspecting the PACF, the last significant lag is the order p^*; Secondly by inspecting the ACF, the last significant lag is the order q^*. Finally, the candidate ARIMA model are selected between the combination matrix: $[0, p^*] \times [0, q^*]$. The selection is usually based on expert or trial and error.

Based on the literature, there is a selection table (Table 3.1 [35]) that shows a general rule of thumb to select the appropriate model.

Table 3.1 Order Estimation of ARIMA after differencing (*d* is determined)

		PACF	
		Decay	Cut-off at Lag p
ACF	Decay	—	ARMA(p,d,0)
	Cut-off at Lag q	ARIMA(0,d,q)	ARIMA (p,d,q)

Once the order of (p, q) is determined, least square estimation or Yule-Walker estimation [50] is used to find the numerical values of $\phi_1 \ldots \phi_p$. Durbin's estimation is used to find the numerical values of $\theta_1 \cdots \theta_q$.

However, there are usually more than one candidate ARIMA models as mentioned in the previous paragraphs. Therefore, after calculating the values for each candidate models, Akaiki information criterion (AIC) or Bayesian information criterion (BIC) [35, 37] are used to select the best models. The equations to calculate AIC and BIC are shown as [37]:

$$\begin{aligned} \text{AIC} &= \log \frac{\sum_{t=k}^{n} (x(t) - \bar{x})^2}{n} + \frac{n + 2k}{n} \\ \text{BIC} &= \log \frac{\sum_{t=k}^{n} (x(t) - \bar{x})^2}{n} + \frac{k \log n}{n} \end{aligned} \tag{3.5}$$

where k is the number of parameters in the model and n is the sample size.

Equation 3.5 are the Akaiki information criterion (AIC) and Bayesian information criterion (BIC) equations. The cut-offs at ACF/PACF can give a rough range on the order p (AR) and q (MA) so that we can have an 2-d parameter grid for $ARIMA(p, d, q)$ formed by $[0, p] \times [0, q]$. Searching along these $(p+1) \times (q+1)$ candidate models, we can have their AIC or BIC values. Among these values, we select the candidate model whose AIC or BIC value is the smallest.

To forecast a future value $x(t+h)$ based on an ARIMA model, a rolling forecast is used. Firstly, ARIMA model is applied to estimate the one step ahead value $\hat{x}(t+1)$. Secondly, the predicted value, together with the historical value form a new set of input to the ARIMA model to obtain the two step ahead value $\hat{x}(t+2)$. Then the process is repeated until $\hat{x}(t+h)$ is obtained.

ARIMA is usually used as a benchmark method in the recent literature [48, 51]. Some researchers apply some advanced algorithms to optimize the parameters of ARIMA in order to improve the performance. In [52], the authors applied particle swarm optimization (PSO) to obtain the optimal parameters of ARMA with exogenous inputs (ARMAX) for short-term load forecasting and the optimized model performed well. For data-analysis based research, ARIMA is a preferred method because it can provide a simple yet analytic model for site survey. In [24], ARIMA was employed to study the UK wind speed and power data. The ARIMA models was sufficiently accurate and informative for such kind of study.

Learning-based approaches are widely found in the literature. Two most popular methods are SVR and ANN. The following subsections will discuss them in details.

3.1.4 Support Vector Regression

Support vector machine (SVM) is a machine learning tool for classification [53] and one of its variants, SVR, is dedicated for regression. The advantage of SVR is that it can map the input data into a non-linear feature space by an appropriately chosen kernel function and then apply linear regression. The kernel function helps SVR to perform well for non-linear datasets. However, choosing a kernel function and the parameters of the kernel function usually undergoes an exhaustive search process [54]. An example of non-linear regression is shown in Figure 3.1, a hyperplane (solid curve) is constructed to fit the trend of the TS data while allowing some tolerance for mistakes (points outside the dashed margins).

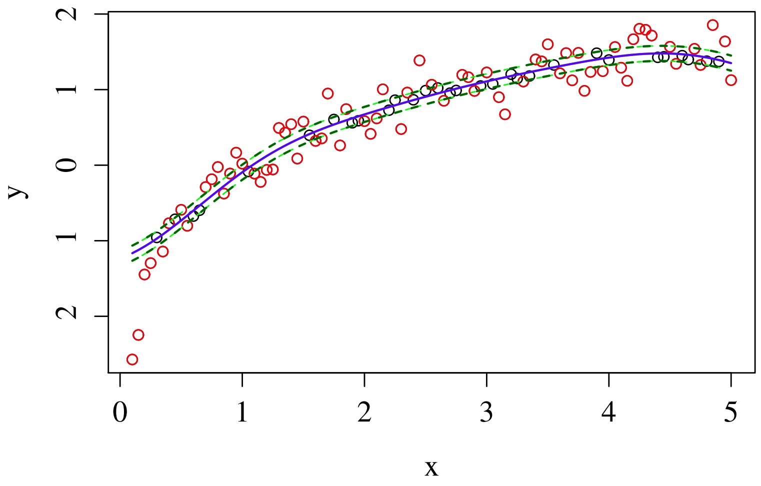


Figure 3.1 An Illustration of SVR, the points are the original data, the solid curve is the SVR regression line, the dashed curves are the margins, and the red points within the margins are the support vectors.

The objective of SVR is to find a maximally flatten hyperplane $f(x) = w^T x + b$ that has at most ε tolerance and minimal slack $\xi^{(*)}$ from the actually obtained targets y_i [55]. The formal SVR optimization problem (Lagrange primal form) is defined as [53]:

$$\min \frac{1}{2}||w||^2 + C\sum_i(\xi_i + \xi_i^*) \tag{3.6}$$

subject to:

$$\begin{aligned} y_i - w^T x_i - b &\leq \varepsilon + \xi_i \\ w^T x_i + b - y_i &\leq \varepsilon + \xi_i^* \\ \xi_i^{(*)} &\geq 0 \end{aligned}$$

where C is the cost factor for trade-off between flatness and tolerance. $\xi_i^{(*)}$ is the slack variable.

To solve (3.6), Lagrange Multiplier Method can be applied:

$$\begin{aligned} \min_{w,b,\xi}\max_{\alpha,\eta} L := \frac{1}{2}||w||^2 + C\sum_i(\xi_i + \xi_i^*) - \sum_i(\eta_i\xi_i + \eta_i^*\xi_i^*) \\ -\sum_i \alpha_i(\varepsilon + \xi_i - y_i + w^T x_i + b) \\ -\sum_i \alpha_i^*(\varepsilon + \xi_i^* + y_i - w^T x_i - b) \\ \alpha_i^{(*)}, \eta_i^{(*)} \geq 0 \end{aligned} \tag{3.7}$$

Equation (3.7) can be solved by quadratic programming algorithms. But an easier way is to solve it with its dual form: $\max\limits_{\alpha,\eta}\min\limits_{w,b,\xi} L$ first. However, only when the KKT conditions are satisfied, the solution of the dual form is equal to the solution of the primal form.

To minimize with respect to w, b, ξ, Lagrange Multiplier method requires $\nabla L = 0$, i.e.:

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_i(\alpha_i^* - \alpha_i) = 0 \tag{3.8}$$

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_i(\alpha_i - \alpha_i^*)x_i \tag{3.9}$$

$$\frac{\partial L}{\partial \xi_i^{(*)}} = 0 \Rightarrow C = \alpha_i^{(*)} - \eta_i^{(*)} \tag{3.10}$$

Additional KKT conditions should also be fulfilled:

$$\alpha_i(\varepsilon + \xi_i - y_i + w^T x_i + b) = 0 \tag{3.11}$$

$$\alpha_i^*(\varepsilon + \xi_i^* + y_i - w^T x_i - b) = 0 \tag{3.12}$$

$$\eta_i^{(*)}\xi_i^{(*)} = 0 \tag{3.13}$$

The dual form is transformed into an optimization function with respect to the Lagrange Multipliers as shown in (3.14). To solve this optimization function, sequential minimization optimization (SMO) [56] is usually applied.

$$\max_{\alpha_i,\alpha_i^*} \sum_i y_i(\alpha_i - \alpha_i^*) - \varepsilon \sum_i (\alpha_i + \alpha_i^*) - \frac{1}{2}\sum_i \sum_j (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) x_i^T x_j \tag{3.14}$$

subject to:

$$\sum_i (\alpha_i - \alpha_i^*) = 0$$

$$0 \leq \alpha^{(*)} \leq C$$

After solving the dual form, the regression function $f(\mathbf{x}) = w^T \mathbf{x} + b$ can be re-written as:

$$f(\mathbf{x}) = \sum_i (\alpha_i - \alpha_i^*) x_i^T \mathbf{x} + b \tag{3.15}$$

By inspecting (3.15), the complexity of $f(\mathbf{x})$ does not depend on the dimension of $\mathbf{x}$ but only on a set of support vectors x_i when $\alpha_i - \alpha_i^* \neq 0$.

In addition, the inner product $x_i^T \mathbf{x}$ can be replaced by a kernel function $K(\cdot)$ without the knowledge of the mapping function $\phi(\cdot)$ provided that $K(\cdot)$ is symmetric and positively semi-definite (Mercer's theorem [56]). Some popularly used kernel functions are:

$$\text{Polynomial Kernel: } K(x_i, x_j) = (\gamma x_i^T x_j)^d \tag{3.16}$$

$$\text{RBF Kernel: } K(x_i, x_j) = e^{\gamma |x_i - x_j|^2} \tag{3.17}$$

$$\text{Sigmoid Kernel: } K(x_i, x_j) = \tanh(\gamma x_i^T x_j) \tag{3.18}$$

where γ is a kernel shape factor and d is the order of the polynomial function.

To solve the intercept variable b, some constraints are derived from the KKT conditions:

$$(C-\alpha_i^{(*)})\xi_i^{(*)}=0 \tag{3.19}$$

$$\alpha_i\alpha_i^*=0 \tag{3.20}$$

The inspection of the above equations shows that: first, only if $\alpha_i^{(*)}=C$ then the points (x_i,y_i) lies outside the bounds; second, α_i and α_i^* cannot be concurrently zero and lastly, $\xi_i^{(*)}=0$ when $\alpha_i^{(*)}\in(0,C)$. Therefore, b can be computed as:

$$b=\begin{cases}-\varepsilon+y_i-w^Tx_i, & \alpha_i\in(0,C)\\ \varepsilon+y_i-w^Tx_i, & \alpha_i^*\in(0,C)\end{cases} \tag{3.21}$$

3.1.5 Artificial Neural Network

ANN is one of the most popular algorithms in machine learning. Neural network can be categorized according to the connectionism: feed-forward and recurrent. Out of the two architectures, feed-forward neural network is the most widely used. It usually has one input layer, one output layer and several hidden layers as shown in Figure 3.2. The neurons in the adjacent layers are connected but there is no interconnection of neurons within the same layer or across non-adjacent layers.

In the input layer, each neuron i_m, $m\in\{1,\dots,M\}$ takes a feature of the input vector and passes to the hidden layer. Each neuron in the nth hidden layer $h_{n,k_n} n\in\{1,\dots,N\}$, $k_n\in\{1,\dots,K_N\}$ is formed by a nonlinear weighted sum of the inputs or the outputs of the preceding hidden layer:

$$h_{1,k_1}=f(\sum_{m=0}^{M}w_{m,k_1}i_m),\forall k_1\in\{1,\dots,K_1\} \tag{3.22}$$

$$h_{n,k_n}=f(\sum_{k_{n-1}=0}^{K_{n-1}}w_{k_{n-1},k_n}h_{n-1,k_{n-1}}),\forall k_n\in\{2,\dots,K_n\} \tag{3.23}$$

where $f(\cdot)$ is a nonlinear activation function, $w_{0,k_n}=1$, $k_n\in\{1,\dots,K_n\}$ denotes the input layer and hidden layer bias, M is the number of input layer neurons, N is the number of hidden layers, K_n is number of nth hidden layer neurons, w_{m,k_1} are the weights between the input and hidden layer neurons, and w_{k_{n-1},k_n} are the weights between the hidden layer neurons.

The commonly used activation functions are (logistic) sigmoid *logsig* function and *tanh* function:

$$logsig(x) = \frac{1}{1+e^{-x}} \tag{3.24}$$

$$tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{3.25}$$

The output neuron is a weighted sum of all the outputs of the hidden neurons:

$$o_l = \sum_{k_N=0}^{K_N} w_{k_N,l} h_{N,k_N}, \forall l \in \{1,\ldots,L\} \tag{3.26}$$

where L is the number of output neurons, and $w_{k_N,l}$ is the weights between the hidden and output layer neurons.

To determine the optimal output value, the weights w_{m,k_1}, $w_{k_N,l}$ and the weights between the hidden neurons should be determined. The most popular way to determine the optimal weights is back-propagation (BP) [57]. However, BP method are usually very slow in computation and can be trapped in local minima.

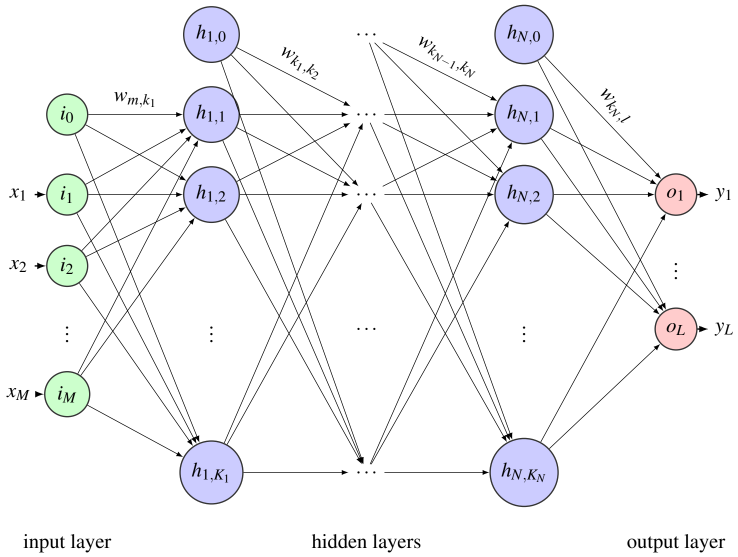


Figure 3.2 Schematic Diagram of an ANN, where M is the number of input layer neurons, N is the number of hidden layers, K_n is number of nth hidden layer neurons, L is the number of output neurons, w_{m,k_1} are the weights between the input and hidden layer neurons, and w_{k_{n-1},k_n} are the weights between the hidden layer neurons and $w_{k_N,l}$ is the weights between the hidden and output layer neurons, x and y are input and output data respectively.

There are on-going researches on SVR and ANN to improve the performance. Variations of SVR and ANN are reported and applied to renewable energy forecasting vastly in the literature such as locally weighted SVR for load forecasting [58], recurrent ANN for TS forecasting [59, 60], etc. Optimization methods were commonly used to tune the parameters of SVR and ANN. Firefly optimization algorithm was applied to tune SVR for short term load forecasting [61], evolutionary programming and PSO was applied to obtain the optimal parameters for SVR on wind speed forecasting [62]. In [63], PSO was used to optimize an ANN for wind power forecasting. In [64], genetic algorithms were employed to optimize an SVR and an ANN for trading TS forecasting. Feature selection assisted forecasting was also reported. In [65], rough set theory was firstly to select the salient features for the SVR to model the short term load demand. The improved SVR had faster training speed and higher accuracy. In [66], fractal theory was applied to select the features to optimize the performance of fuzzy neural network for TS prediction.

3.2 Ensemble Time Series Forecasting

Ensemble method is popular in statistics and machine learning. It uses multiple predictors to obtain an aggregated decision which is better than any of the base predictors [67]. According to Opitz and Maclin [67], there are two kinds of ensemble methods: competitive and cooperative for ensemble classification. Similar to classification, ensemble forecasting can be categorized into competitive and cooperative ensemble forecasting. Competitive ensemble forecast is to train different predictors individually with different datasets or the same dataset but with different parameters and then the prediction is obtained by aggregating the decisions of all the individual predictors (base predictors). On the other hand, cooperative ensemble forecast is to divide the prediction task into several sub-tasks and select appropriate predictors for each sub-task based on the characteristics of the sub-tasks, and the final decision is a sum of all the outputs of the base predictors. The following subsections details the two kinds of ensemble forecasting methods.

3.3 Competitive Ensemble Forecasting

Competitive ensemble forecasting approach uses multiple predictors constructed with slightly different initial conditions or different parameters that are able to construct individual forecast models to form an ensemble forecast model (multi-model ensemble). The forecasting results from all the models or selected models after pruning are aggregated together. The confidence level can be measured by the variations of the spread of individual results which are used during the aggregation [68].

The theory behind the competitive ensemble method is investigated by many researchers. A proof called ambiguity decomposition [69, 70] was reported to show that a proper ensemble predictor can guarantee to have smaller squared error than the average squared error of the base predictors. However, ambiguity decomposition only investigates a single dataset with ensemble methods, for multiple datasets, bias-variance-covariance decomposition is introduced [70, 71] and the equation is shown:

$$\mathrm{E}[\overline{f}-t]^2 = bias^2 + \frac{1}{M}var + (1-\frac{1}{M})covar \tag{3.27}$$

$$bias = \frac{1}{M}\sum_{i=1}^{M}(\mathrm{E}[f_i]-t) \tag{3.28}$$

$$var = \frac{1}{M}\sum_{i=1}^{M}\mathrm{E}[f_i-\mathrm{E}[f_i]]^2 \tag{3.29}$$

$$covar = \frac{1}{M(M-1)}\sum_{i}\sum_{j\neq i}\mathrm{E}[f_i-\mathrm{E}[f_i]](f_j-\mathrm{E}[f_j]) \tag{3.30}$$

where t is the target, M is the ensemble size $\mathrm{E}(\cdot)$ is the expectation function, and f_i is the output from each single model.

There exist several theoretical insights about the soundness of using ensemble methods such as strength-correlation [72], stochastic discrimination [73] and margin theory [74]. They have been shown to be equivalent to bias-variance-covariance decomposition in some sense [75].

From the equation, we can see the term *covar* (covariance) can be negative, which may decrease the expected loss of the ensemble while leaving *bias* (bias) and *var* (variance) unchanged. Beside the *covar*, the number of models also plays an important role. As it increases, the proportion of the variance in the overall loss vanishes whereas the importance of the covariance increases. Overall, this decomposition shows that if we are able to design low-correlated base predictors, we can expect an increase in performance.

Diversity [76, 77] is a way to realize the bias-variance-covariance in real implementation. Low-correlated base predictors means high degree of diversity. For classification, diversity can be further categorized as data diversity, parameter diversity and kernel diversity [11, 12, 67]. For forecasting, kernel diversity is not commonly found in the literature and will be discussed as a future research direction.

A block diagram of competitive ensemble forecasting is shown in Figure 3.3. There are two variations based on data diversity and parameter diversity. Some example competitive ensemble forecasting models are also listed in the figure.

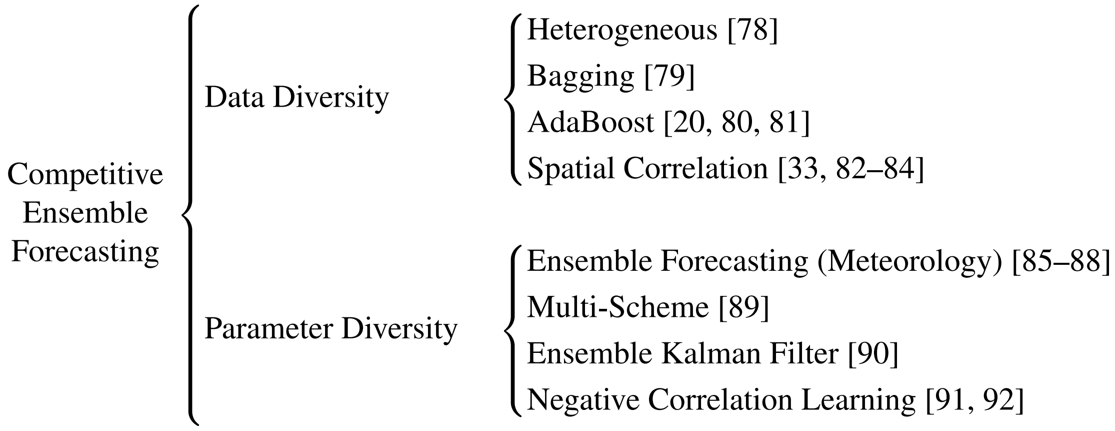

Competitive Ensemble Forecasting
Data Diversity
Heterogeneous [78]
Bagging [79]
AdaBoost [20, 80, 81]
Spatial Correlation [33, 82–84]
Parameter Diversity
Ensemble Forecasting (Meteorology) [85–88]
Multi-Scheme [89]
Ensemble Kalman Filter [90]
Negative Correlation Learning [91, 92]

Figure 3.3 Block Diagram of Competitive Ensemble Forecasting Methods.

3.3.1 Data Diversity

For data diversity-based competitive ensemble forecasting, more than one input datasets are fed into the forecasting system. There are two variations as shown in (3.31) and (3.32). equation (3.31) represents a forecasting approach that applies N predictors $f_1(\cdot)\dots f_N(\cdot)$ for N input datasets $\mathbf{x}_1\dots\mathbf{x}_N$ and the final prediction is a (weighted) average of them all. equation (3.32) represents another forecasting approach that only employs a single predictor $f(\cdot)$ for N input datasets. The example competitive ensemble forecasting approaches are discussed in the following paragraphs.

$$\hat{y}(t+h) = \frac{1}{n}\sum_{i=1}^{N} w_i f_i(\mathbf{x}_i(t)) \tag{3.31}$$

$$\hat{y}(t+h) = f(\mathbf{x}_1(t),\dots,\mathbf{x}_N(t)) \tag{3.32}$$

where $\hat{y}$ is the predicted value and h is the forecast horizon.

Heterogeneous

A study on solar irradiance forecasting was reported in [78]. A heterogeneous competitive ensemble forecasting method was proposed: the first model predicted the solar irradiance using non-linear regression on meteorological data and the second model predicted the solar irradiance based on daily pattern recognition. The authors compared their results with reported results and stated that the heterogeneous competitive ensemble forecasting method had a smaller error than the errors of previously reported methods. The authors also proposed to use an adaptive learning method to update the models on a daily basis.

Bagging

Bootstrap aggregation (Bagging) is a popular method for data diversity [93]. Bagging has two phases: the bootstrap phase to sample the original dataset with replacement to obtain *N* datasets, and the aggregating phase to combine the outputs of *N* base predictors trained by each dataset by averaging. In [79], a Bagging ANN model was introduced for short-term solar irradiance forecasting. The Bagging ANN model consists of three different types of ANNs: MLP, Radial Basis Function ANN (RBFNN) and Recurrent Neural Network (RNN) with historical data after bootstrapping. The results showed that the error measures of the Bagging ANN models were smaller than those of single MLP, RBFNN and RNN.

AdaBoost

Adaptive Boosting (AdaBoost) is an extension of Bagging [94]. It first applies Bagging to create a collection of datasets, introducing data diversity. Then it sequentially trains the predictor with the datasets, and at each iteration, AdaBoost assigns higher weights to the data points with larger error and smaller weights to the data points with smaller error. AdaBoost can boost the prediction accuracy of weak predictors.

Although AdaBoost was originally developed for classification problems [95], there are variants for regression in: gradient boosting [80, 96, 97], big error margin boosting [98], AdaBoost.R [99] (developed from AdaBoost.M2 [100]), AdaBoost.RT [94] (developed from AdaBoost.M1 [95]), AdaBoost+ [101] (modified version of AdaBoost.RT). Some papers reported to apply AdaBoost for TS forecasting, such as gradient boosting with penalised regression splines to forecast a set of load profiles [80], AdaBoost-ANN [81] and EMD-AdaBoost-ANN [20] for short-term wind speed forecasting.

Spatial Correlation

Spatial correlation based forecasting use data from the location of interest and neighborhood locations to form a model. It is advantageous over single location based forecasting because it not only takes terrain and other geographical variations into account but also reveals the correlations among the neighborhood data and thus reduces the error of prediction, which is known as the spatial smoothing effect [27]. Spatial correlation based forecasting is applied to short-term solar irradiance forecasting in [33, 82] and wind speed forecasting in [83, 84].

3.3.2 Parameter Diversity

Parameter diversity approach varies the parameters $\theta_1 \dots \theta_N$ of a predictor to produce N predictors $f_1(\cdot) \dots f_N(\cdot)$ and each predictor will learn from the same dataset $\mathbf{x}$. The predicted value $\hat{y}$ is obtained by averaging the outputs of all predictors as shown in Equation (3.33).

$$\hat{y}(t+h) = \frac{1}{N} \sum_{i=1}^{N} f_i(\mathbf{x}(t), \theta_i) \tag{3.33}$$

where $\hat{y}$ is the predicted value and h is the forecast horizon.

Ensemble Forecasting (Meteorology)

Ensemble forecasting (meteorology) is developed since 1990's for long-term and long-range weather forecasting [85, 102]. The ensemble forecasting methods comprise several numerical predictions integrated from different initial conditions. The initial conditions represent different initial uncertainties of the meteorological analysis. European Center for Medium-Range Weather Forecasts (ECMWF) [86, 88] and National Centers for Environmental Prediction (NCEP) [86, 87, 102] use perturbation theory for creating the initial conditions. Ensemble forecasting enhances the performance of NWP yet requires more computing power.

Multi-Scheme

A multi-scheme ensemble forecasting method was reported in [89]. Multiple predictors were constructed from NWP model with modified physical parameters. The results showed that the reported method has smaller MAE and RMSE than single forecasting and forecasting with simpler training methods [89].

Ensemble Kalman Filter

Kalman filter is a sequential data assimilation algorithm for solving non-linear problems [103]. Ensemble Kalman Filter (EnKF) [90] is an ensemble version of Kalman filter. In [90], the authors introduced a two-stage ARIMA-EnKF for wind speed forecasting. Firstly, $\text{ARIMA}(3,1,0)$ was employed to model the wind speed TS. Secondly, a state vector was formed by concatenating the coefficients of the ARIMA model, the current wind speed and the historical wind speed. Finally, EnKF used the ensemble state vectors with slightly varying parameters for forecasting. The result showed that ARIMA-EnKF had smaller RMSE and MAE than ARIMA alone. Other variations of EnKF were also reported for wind speed/power forecasting such as reduced-rank extended Kalman filter, bred mode EnKF [103].

Negative Correlation Learning

Negative correlation learning (NCL) is a well-known ensemble learning/training algorithm [70, 91, 104]. It introduces strong diversity among different base learners with the same training dataset for each base learners. NCL has a convenient way to balance the bias-variance-covariance trade-off.

In [92], and NCL with random vector functional link (RVFL) neural network ensemble was reported. The advantage of RVFL is that the weights of the neural network is calculated with a closed form, avoiding BP and there is additional input layer to output layer connections. Although the proposed ensemble RVFL method was not tested on renewable energy related TS, it can be applied to wind, solar and load forecasting easily. Detailed discussion on RVFL is in Chapter 6.

3.4 Cooperative Ensemble Forecasting

Cooperative ensemble forecasting divides a prediction task into several sub-tasks and solves each sub-task individually. The overall forecasting results are obtained by aggregating the forecast values of all predictors. It is sometimes referred as ‘hybrid’ approach. A block diagram of cooperative ensemble forecasting is shown in Figure 3.4. There are two variations: one is pre-processing and the other is post-processing. Typical approaches are listed in the figure. These two variations are discussed in the following subsections.

3.4.1 Pre-processing

Pre-processing is to divide the input dataset into several sub-datasets and each sub-dataset is modelled and predicted by a predictor. Usually the predictor is the same for all sub-datasets. The final prediction is a summation of all the outputs of the predictors.

Wavelet Decomposition

A popular TS decomposition method called wavelet decomposition was reported in several literature [105, 106, 118, 119]. Wavelet theory is to study the TS in time-frequency domain

$$\begin{array}{l}\text{Cooperative}\\ \text{Ensemble}\\ \text{Forecasting}\end{array}\left\{\begin{array}{ll}\text{Pre-processing} & \left\{\begin{array}{l}\text{Wavelet Decomposition [7, 105–109]}\\ \text{EMD [17–19, 110, 111]}\\ \text{Complex Value [112]}\end{array}\right.\\ & \\ \text{Post-processing} & \left\{\begin{array}{l}\text{ARIMA-GARCH [113]}\\ \text{ARIMA-ANN/SVR [114, 115]}\\ \text{SVR-SVC [116]}\\ \text{Multi-Stage [117]}\end{array}\right.\end{array}\right.$$

Figure 3.4 Block Diagram of Cooperative Ensemble Forecasting Methods.

as well as time domain. Wavelet decomposition is to decompose the TS into a set of sub-series based on a mother wavelet that can be predicted more accurately [106]. There are two kinds of decompositions: continuous and discrete. For practical applications, discrete wavelet transform (DWT) is usually used for decomposition. The key equations of wavelet decomposition based forecasting are shown in (3.34).

$$\begin{aligned}\{\mathbf{x}_{D_i}(t),\mathbf{x}_{A_j}(t)\} &= \mathrm{DWT}(\mathbf{x}(t))\\ \hat{y}_{D_i}(t+h) &= f(\mathbf{x}_{D_i}(t))\\ \hat{y}_{A_j}(t+h) &= f(\mathbf{x}_{A_j}(t))\\ \hat{y}(t+h) &= \sum_{i=1}^{n}\hat{y}_{D_i}(t+h)+\hat{y}_{A_j}(t+h)\end{aligned} \tag{3.34}$$

where $\mathbf{x}$ is the original dataset, $\mathbf{x}_{D_i}$ is the ith detailed component, $\mathbf{x}_{A_j}$ is the jth abstract component, h is the forecasting horizon, $\hat{y}$ is the predicted value and $f(\cdot)$ is the predictor.

In [105], a number six Daubechies wavelet was used for the discrete wavelet decomposition. After decomposition, there were three detailed decompositions ($\mathbf{x}_{D_i}(t)$, $i = 1,2,3$) and one approximated decomposition ($\mathbf{x}_{A_3}(t)$). ARIMA was selected to be the predictor for each sub-series. Finally the forecast results were aggregated to obtain the final result. The reported wavelet-ARIMA model was evaluated with 3, 5, and 10-hour ahead wind speed forecast, and the performance was better than the conventional ARIMA method.

Cao and Cao [7] presented a wavelet Recurrent BP Network (RBPN) for solar irradiance forecasting. The RBPN is a dynamic ANN that has the option to feedback the outputs totally or partially. The wavelet used was number seven Daubechies wavelets. The authors used cut-and-trial method [7] to determine the number of hidden neurons of a 3-layer

RBPN. The results showed that for one-day ahead forecasting, the proposed wavelet RBPN outperformed the RBPN without wavelet decomposition.

Empirical Mode Decomposition

Empirical Mode Decomposition (EMD) is another TS decomposition method [120]. Unlike wavelet decomposition, it decomposes a TS based on the time domain only without a deterministic function (as mother wavelet in wavelet decomposition). The decomposed TS then undergoes the similar procedures as in wavelet decomposition.

In [17] and [18], the authors proposed to firstly decompose the wind speed TS followed by extracting and forming feature sets from the decomposed data. Then the feature sets were used to train a k-nearest neighbor (kNN) and SVR, respectively. In [19], the authors proposed to use an ensemble version of EMD called complete ensemble EMD with adaptive noise to decompose the wind speed TS data and train ANN and SVR with the decomposed data. The EMD based forecasting methods outperformed the forecasting methods without decomposition.

An EMD-SVR method was employed to forecast the short-term load demand [110]. Another study applied EMD to decompose the load demand TS into high-frequency and low-frequency subsets and applied SVR on the high frequency TS but AR on the low frequency TS as the authors states that the high frequency components are more towards non-linear whereas the low frequency components are more towards linear TS, which can be easily modelled by statistical models [111]. Both of the EMD based methods showed some improvements over the conventional methods.

Complex Value

If wind speed and wind direction are considered as complex numbers, it is possible to convert complex valued wind speed TS from polar form (|speed|∠direction) to Cartesian form ($Re + jIm$). The key equations of complex value transformation based forecasting are shown below:

$$\begin{gathered}\mathbf{a}(t) + j\mathbf{b}(t) = \mathbf{x}(t)\angle\theta(t) \\ \hat{y}(t+h)\angle\hat{\theta}(t+h) = f_{Re}(\mathbf{a}(t)) + jf_{Im}(\mathbf{b}(t))\end{gathered} \tag{3.35}$$

where $\mathbf{x}$ is the wind speed magnitude, θ is the wind direction, $\mathbf{a}$ and $\mathbf{b}$ are the real and imaginary part of the wind vector, $f_{Re}(\cdot)$ and $f_{Im}(\cdot)$ are the prediction functions for real and imaginary parts, respectively, $j = \sqrt{-1}$, h is the forecast horizon, $\hat{y}$ and $\hat{\theta}$ are the predicted magnitude and direction, respectively.

In [112], a Complex Valued Neural Network (CVNN) was employed for wind speed forecasting. CVNN is similar to Real Valued Neural Network (RVNN) but it can handle complex valued data without introducing two RVNNs for real part and imaginary part separately. The authors used CVNN for 24-hour ahead forecasting of a wind speed/direction data from Japan Meteorological Agency during 2009 and compared with RVNN, the MAPE was smaller for every month.

3.4.2 Post-processing

A complex TS may have more than one characteristics and each characteristic is suitable for one particular method. For example, ARIMA model is suitable for modelling linear TS and ANN is more preferred for modelling non-linear TS. Forecasting the TS consecutively by two or more predictors is considered as cooperative ensemble forecasting based on post-processing.

There are several cooperative ensemble forecasting models based on post-processing reported in the literature, such as ARIMA-GARCH [113], ARIMA-ANN, ARIMA-SVM [114] and SVR-SVC [116].

ARIMA-GARCH

ARIMA is a linear TS model that models and forecasts general TS. Generalized Auto-Regressive Conditional Heteroscedasticity (GARCH) is a non-linear model that models the dynamic variance of the residuals after linear regression. As shown in Equation (3.36), ARIMA firstly models the linear portion of the TS and then the residual ε is modelled by a GARCH function.

$$\begin{aligned} \hat{y}(t+1) &= \text{ARIMA}(x(t),\ldots,x(t-p+1)) + \varepsilon(t) \\ \varepsilon(t) &= z(t)\delta(t),\; z(t) \sim \mathcal{N}(0,1) \\ \delta(t)^2 &= \text{GARCH}(\delta^2(t-1),\ldots,\delta^2(t-a), \\ &\qquad \varepsilon^2(t-1),\ldots,\varepsilon^2(t-b)) \end{aligned} \tag{3.36}$$

where $\hat{y}$ is the predicted value, ε is the residual, δ is the standard deviation, p is the order of the AR term in the ARIMA model, a and b are the orders of the δ^2 and ε^2 terms, respectively.

In [113], an ARIMA-GARCH model was presented. The wind speed TS was converted to stationary and normally distributed by one step differencing and Weibull Cumulative Distribution Function (CDF) fitting. Two wind speed TS prediction models namely ARIMA(1,1,4) - GARCH(1,1) and ARIMA(1,0,1) - GARCH(1,1) were used. Although the objective of this paper was to find a proper model for wind speed modelling, extension on forecasting is possible by applying the models to the testing data.

ARIMA-ANN/SVR

Several ARIMA-ANN and ARIMA-SVM models were reviewed in [114] which focused on reviewing the hybrid forecasting approaches. As stated in the paper, ARIMA is a robust and the most widely used model for linear TS whereas ANN and SVM are the typical CI tools for non-linear TS regression.

A wind speed/power TS y_t can be decomposed into two parts: a linear part l_t and a non-linear part n_t. Firstly, the linear part can be modelled by ARIMA. Secondly, a CI model $f(\cdot)$ is applied to the residuals after ARIMA fitting to model the non-linear part. Finally, the two modelled parts are added together to obtain the final predicted result $\hat{y}_t$.

In [115], an ARMA-time delay neural network (TDNN) model was reported. The authors firstly used several statistical de-trending methods to generate a stationary solar irradiance TS and then they used ARMA to predict the linear part of the TS and TDNN to predict the non-linear part. Finally they combined the two predicted results for the final prediction. The author compared the hybrid ARMA-TDNN model with single ARMA and TDNN models and the results showed a smaller normalized RMSE (nRMSE) due to the hybrid model.

SVR-SVC

A wind speed forecasting tool based on SVM regression (SVR) and SVM classification (SVC) was reported in [116]. It is a two-stage forecasting with SVR as the main predictor and SVC as the post-processing tool. A TS was firstly modelled by an SVR to obtain the predicted output $\hat{s}_t$ as well as the residuals e_t. Then the residuals e_t are classified by an SVC into one of three possible classes: $(-\infty, -\varepsilon]$, $(-\varepsilon, +\varepsilon)$, and $[+\varepsilon, +\infty)$ based on a user defined threshold ε. Thirdly, a compensation mapping from the residuals e_t to an offset ω is established such that:

$$\hat{y}(t+h) = \begin{cases} \hat{s}(t) + \omega & e(t) \leq -\varepsilon \\ \hat{s}(t) & -\varepsilon < e(t) < +\varepsilon \\ \hat{s}(t) - \omega & e(t) \geq +\varepsilon \end{cases} \tag{3.37}$$

The authors used an hourly wind speed dataset to test the model against SVR [116]. The reported model had smaller MSE and MAPE than the SVR model. However, the user defined threshold ε and the offset ω were not clearly defined and it is dataset dependent.

Multi-Stage

For solar irradiance forecasting, a multi-stage ANN was reported in [117]. There were three stages: (i) first stage forecasts the average atmospheric pressure based on the historical average atmospheric pressure data, (ii) second stage forecasts the irradiance level based on the output of the first stage and (iii) third stage forecasts the solar irradiance based on the output of the second stage as well as the historical meteorological data. This multi-stage ANN was applied to forecast the heat pump water supply system rather than a PV system. The results showed a reasonable improvement compared with a single-stage ANN.

3.5 Performance Comparison of Ensemble Forecasting Models based on Results in the Literature

Various ensemble forecasting methods are reviewed in the above sections. The applications of the methods are on wind speed/power forecasting and solar irradiance forecasting. Although the methods can be categorized into competitive ensemble forecasting and cooperative ensemble forecasting, the performance cannot be clearly differentiated due the unique characteristics of the datasets. Table 3.2 shows the detailed results of some selected ensemble methods.

As shown in Table 3.2, the forecasting horizons for wind speed/power of the selected reviewed methods fall into the short-term to mid-term categories according to Table 2.2 except for the Multi-scheme model [89] which is a long-term forecasting model. The forecasting horizons for solar irradiation in Table 3.2 of the selected reviewed methods are all 1 day ahead which is mid-term forecasting according to Table 2.4.

Table 3.2 Results Summary of Reviewed Ensemble Forecasting Models on Wind and Solar TS Forecasting.

Methodology	Resolution	Horizon	Benchmark Methods	Improvement*
Spatial Correlation, Fuzzy [121]	20, 10, 5 min	0.5, 1, 2, 4 h	Single site	3.02% (RMSE)
Multi-scheme [89]	1 h	48 h	Single-scheme	17.07% (MAE), 16.96% (RMSE)
EnKF [90]	10 min	10 min	ARIMA	9.49% (RMSE), 27.64% (MAE), 21.54% (MRE)
Wavelet-PSO-ANFIS [106]	15 min	3 h	ARIMA, Wavelet-ANN, Wavelet-Fuzzy	15.67% (MAPE), 15.75% (NMAE)
CVNN [112]	1 h	24 h	RVNN	36.96% (MAPE), 10% (MAE)
ARIMA-ANN/SVR [114]	1 h	1—9 h	ANN, SVR	0.1%–5.5% (RMSE, MAE)
SVR-SVC [116]	1 h	1 h	SVR	32.9% (MAPE), 33.3% (MSE)
Non-linear Regression [78]	1 h	1, 3 h	Regression, ARIMA, ANN	40% (MRE, 1 h), 33.3% (MRE, 3 h)
Bagging ANN [79]	1 h	1 day	MLP, RBFNN, RNN	17.4% (MAE), 2.3% (MAPE)
Wavelet-RBPNN [7]	1 day	1 day	RBPNN	74.5% (MAE), 77.61% (MRE)
Multi-stage ANN [117]	1 day	1 day	ANN	17.43% (MAE)

$*$: The improvement is based on the comparison between the reported method E_p and the best performed benchmark method E_b. Improvement $= \frac{E_b - E_p}{E_b} \times 100\%$.

The ensemble forecasting methods generally outperformed the best alternative methods. The only exception is that the ARIMA-ANN/SVR for wind speed/power forecasting [114] did not outperform the ARIMA model.

3.6 Concluding Remarks

This chapter has discussed the ensemble forecasting methods. They are categorized into competitive and cooperative ensemble forecasting methods. Competitive ensemble forecasting methods usually require a larger number of datasets or a larger collection of parameters for the predictor. Therefore, competitive ensemble forecasting requires highly demanding computing power and computation time. In addition for competitive ensemble forecasting methods based on physical approaches, a large investment on meteorological stations is expected. However, competitive ensemble forecasting improves the forecasting accuracy by diversity. Competitive ensemble forecasting is usually used in medium term and long term forecasting.

Cooperative ensemble forecasting methods usually have lower computing power requirements than competitive ensemble forecasting methods because they do not need to model large number of datasets from different stations or model one dataset using several parameter sets. However, the forecasting time horizon of cooperative ensemble forecasting methods is usually shorter than competitive ensemble forecasting methods. Therefore, cooperative ensemble forecasting is mainly used for very short term or short term forecasting.

Chapter 4

Empirical Mode Decomposition Based Short-term Wind Speed Forecasting

EMD, as discussed in Chapter 3, is a TS decomposition method that is originated from signal processing [120]. It is an adaptive time domain method that does not rely on a deterministic function. The decomposed TS often have narrow bandwidths, which facilitate the modelling by the machine learning methods. In Chapter 3, there are several EMD based forecasting methods reported in the literature and these methods are under the cooperative ensemble method category. In this chapter, an EMD based SVR forecasting method is applied for short-term wind speed forecasting.

Four research questions (RQs) are raised and will be answered in this chapter:

RQ-4.1 How do forecast horizons moderate the relative performance of the EMD based forecasting methods and the benchmarks?

RQ-4.2 Do the EMD based forecasting methods perform better than the non-EMD based counterparts?

RQ-4.3 Does the proposed EMD-SVR with input vector reconstruction perform better than some reported benchmark methods?

RQ-4.4 Is feature selection an important factor to improve the forecasting accuracy for the EMD based methods?

4.1 Empirical Mode Decomposition

EMD decomposes a TS into a collection of intrinsic mode functions (IMFs) and a residue. EMD is based on the local characteristics of the TS such as the local maxima, local minima and zero-crossings. Since the characteristics were determined empirically from the TS, the operation is local and adaptive, which is suitable for non-stationary and non-linear TS modelling. The algorithm of EMD is as follows [120] (shown in Figure 4.1 as well):

1. Identify all local maxima and local minima in the TS $\mathbf{x}(t)$ and interpolate all local extrema to form an upper envelope and a lower envelope $u(t)$ and $l(t)$, respectively.
2. Find the mean of upper and lower envelopes $m(t) = \frac{u(t)+l(t)}{2}$.
3. Subtract the mean from the original TS to obtain a detailed component $d(t) = x(t) - m(t)$.
4. If $m(t)$ and $d(t)$ satisfy one of the stopping criterion, then the first IMF $c_1(t) = m(t)$ and the first residue $r_1(t) = d(t)$. The stopping criteria are: (i) $m(t)$ approaches zero, (ii) the number of local extrema and the number of zero-crossings of $d(t)$ differs at most by one or (iii) the user-defined maximum iteration is reached.
5. Else, repeat steps 1 to 4 for $d(t)$ until $c_1(t)$ and $r_1(t)$ are obtained. These iterative steps are known as *sifting*.
6. For $r_1(t)$, repeat Steps 1 to 5 until all the IMFs and the residue are obtained.

Finally, the original TS is decomposed as:

$$x(t) = \sum_{i=1}^{N} c_i(t) + r_N(t) \tag{4.1}$$

However, in [120], the stopping criteria of sifting is not quantified clearly. Therefore, in [122], two threshold-based stopping criteria were reported:

$$\frac{\#(t|\delta(t) < \theta_1)}{\#(t)} \geq 1 - \alpha \tag{4.2}$$

$$\delta(t) < \theta_2 \tag{4.3}$$

where $\delta(t) = |\frac{u(t)+l(t)}{u(t)-l(t)}|$, α, θ_1, θ_2 are user defined constants and $\#(\cdot)$ is a function to count the numbers in a set.

A wind speed TS that is decomposed by EMD is shown in Figure 4.2.

The decomposed IMFs and the residue has two important characteristics, one is completeness and the other is orthogonality. The completeness is shown in (4.1), which means that the decomposition is loss-less. The orthogonality is obtained empirically [120] although it is not required for signal processing. Orthogonal TS can reduce the modelling complexity and at the same time, can effectively determine the input features of the model.

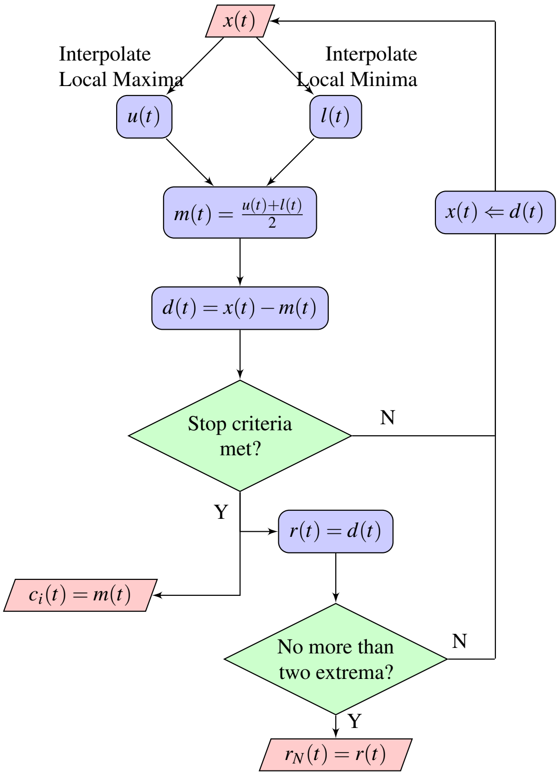


Figure 4.1 Algorithm of Empirical Mode Decomposition

4.2 EMD based Forecasting Methods

The most common way to forecast a TS is to form an AR model which is discussed in Section 3.1.3. An $AR(p)$ model is to predict the future value based on the historical value up to lag p. In machine learning based forecasting, it is known as a feature vector with length p and in signal processing, it is known as a finite impulse response (FIR) filter with tap p [123].

The advantage of applying machine learning based methods to the data of a length p feature vector is that non-linear functions can be applied to the features so as to better approximate the future value. However, as the TS discussed in Chapter 2 are highly non-linear and non-stationary, information are deeply embedded in the TS. Therefore, some decomposition process is required to extract and separate the salient information from the original TS. In this particular thesis, we choose EMD as the decomposition tool and the reason was discussed in the previous section.

There are several EMD based methods for wind speed/power forecasting and they are are listed in Table 4.1. They can be classified into two main approaches: multiple predictors and single predictor.

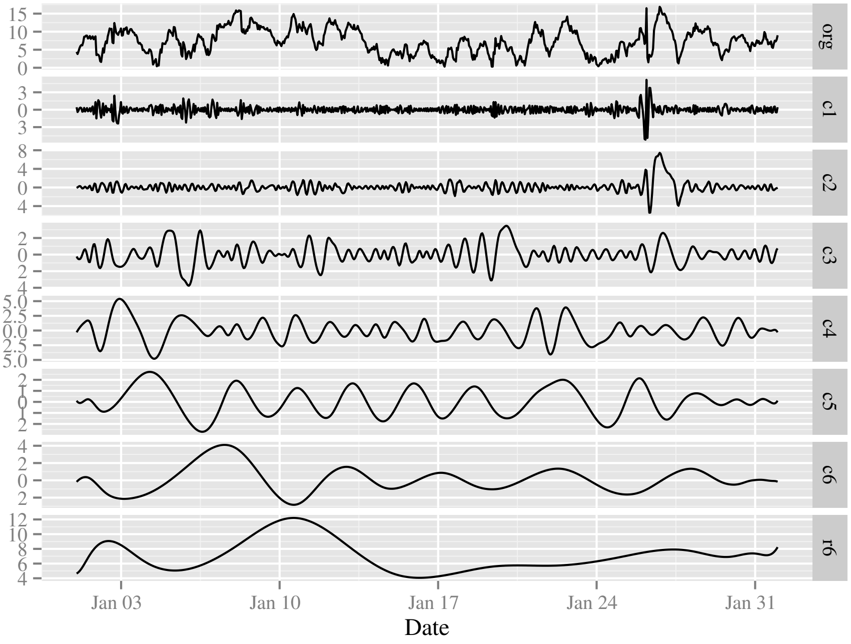


Figure 4.2 A Wind Speed TS Decomposed by EMD, ‘org’ is the original TS, c_1 — c_6 are the IMFs and r_6 is the residue.

The multiple predictors approach [124–126, 128, 129, 131, 134] employs EMD for TS decomposition and applies separate predictors on each IMF. The mathematical representation is shown in (4.4). There are several variations on the multiple predictors approach: (i) fine to coarse grouping [124, 131] is to first classify the IMFs according to their frequency domain behavior and apply different modelling methods to each class; (ii) discarding [128] is to discard noisy or insignificant IMFs before prediction.

$$\hat{x}(t+h) = \sum_{i=1}^{N} f_i(c_i(t), \ldots, c_i(t-p_i+1), \theta_{c_i}) + f(R_N(t), \ldots, R_N(t-l_i+1), \theta_R) \tag{4.4}$$

where $\hat{x}(t+h)$ is the h-step ahead predicted value, $c_i(t)$ is the ith decomposed IMF, $R_N(t)$ is the residue, p_i is the ith lag, $f(\cdot)$ is a prediction function and θ is the parameter set of the prediction function.

The single predictor approach [133] also employs EMD for TS decomposition in the first stage. It then extracts the values at the same time-stamp in each IMF and the residue to create an input vector. This input vector is used for prediction. The mathematical representation is:

$$\hat{x}(t+h) = f(c_i(t), R_N(t), \theta),\ i \in \{1, \ldots, N\} \tag{4.5}$$

Table 4.1 Summary of Selected EMD based wind speed/power forecasting

Citation	Predictor	Post-processing	Remarks
[124]	SVR	Grouping	Group first 3 IMFs to High Frequency Components (HF) and the remaining to Low Frequency Components (LF); Radial Basis Function Kernel (RBF)-SVR to model the HF and Polynomial Kernel (Poly)-SVR to model the LF
[125]	LS-SVR	Grouping	RBF-SVR to model the first 5 IMFs and Poly-SVR to model the remaining
[126]	ANN	–	The aggregation is weighted by a set of aggregation coefficients
[127]	RBFNN	–	Kalman Filter to denoise the TS in advance
[128]	RBFNN	Discarding	Discard the first IMF because it is highly fluctuating yet very small in amplitude
[129]	ARMA	–	ARMA to model the decomposed IMFs and Residue separately
[130]	MA & Persistent	Classification & Grouping	Improved Persistent Approach (IPA) to classify the IMFs into HF and LF; MA to model the HF and Persistent approach to model the LF
[131]	ANN, ARMA	Classification & Grouping	t-test to classify and group IMFs into HF, Medium Frequency (MF) and LF; ANN to model the HF, ARMA to model the MF and the LF
[132]	LEP, & GF	Classification	Phase-space Mutual Information and Wolf method to classify IMFs into chaotic and non-chaotic; LEP to model the chaotic IMFs and GF to model the non-chaotic IMFs
[133]	SVR	–	IMFs are considered as input attributes of the SVR

LS-SVR: least square SVR
LEP: Lyapunov exponent prediction
GF: grey forecast

The multiple predictors approach has to train a forecasting model for each IMF and the residue whereas the single predictor approach needs to train a single forecasting model for the whole decomposed TS up-to lag l. Therefore, the single predictor approach has less complexity than the multiple predictors approach. However, Han *et al.* [133] used IMF with only $l = 1$ to create an input vector which may not provide sufficient information for the forecasting models and the orthogonality property may be suppressed.

4.3 A Novel EMD-SVR Model with Input Vector Reconstruction

In view of the EMD based forecasting methods discussed in the previous section, we can see that for multiple predictors approach, more than one TS are generated and it requires one predictor each for each decomposed TS. Error will be heavily accumulated if the decomposition is sub-optimal. In fact, mode-mixing which will be introduced in Chapter 5 is a common problem that generates sub-optimal decomposed TS and thus degrade the overall performance. The single predictor approach introduced in the previous section focused on reducing the computation time. However, autocorrelation and orthogonality of EMD which are important characteristics of TS are not included in the single predictor approach.

We propose an EMD-SVR model with input vector reconstruction that (i) firstly decomposes the wind speed TS $\mathbf{x}(t)$ by EMD into a collection of IMFs: $c_i(t)$ $i \in \{1,\ldots,N\}$, and a Residue r_N, (ii) then combines the IMFs up-to an optimal lag l based on a feature selection module to create an input vector and (iii) finally uses an SVR to forecast the output from the input vector. The proposed EMD-SVR model is advantageous over the reported methods that (i) it does not require more than one predictor to model the decomposed TS which saves computational power and (ii) it also retains the importance of autocorrelation of the TS. The flow chart of the proposed EMD-SVR model is shown in Figure 4.3.

The feature selection module first determines the lag l from the original TS $\mathbf{x}(t)$ based on the PACF, which is discussed in Section 3.1.3. Then the feature selection module combines $c_i(t),\ldots,c_i(t-l+1)$ and $R_N(t),\ldots,R_N(t-l+1)$ together to form an $(N+1)\times l$ length vector. This vector serves as the input to the predictor.

The proposed EMD-SVR model is superior to the EMD-SVR model reported in [124] because the proposed method requires only one SVR to be trained while [124] requires multiple predictors to be trained. The proposed method is also superior to the EMD-SVR model reported in [133] because the proposed method forms an input vector by using more than one time lagged IMFs where the time lag l, usually larger than one, is obtained from PACF.

4.4 Experiment Setup

The wind speed data sets were retrieved from National Data Buoy Center (NDBC) [135] sites 41004, 44009 and 46077. The datasets were partitioned according to season into 12 TS and each TS is further partitioned into two parts: 70% for training and 30% for testing. The forecasting horizon is 1 to 12 hours ahead. These horizons are typical short-term wind

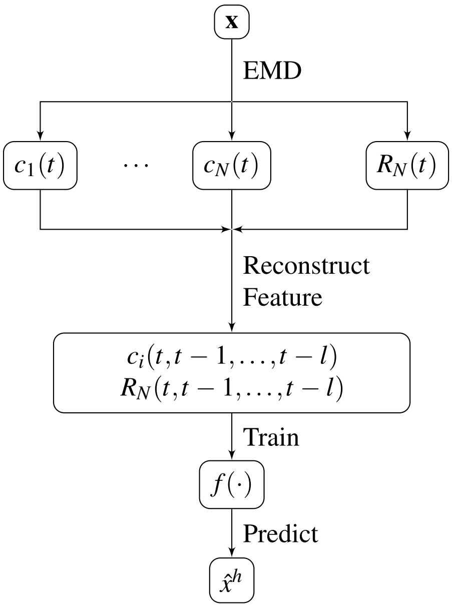


Figure 4.3 Flow Chart of the proposed EMD-SVR Model

speed forecasting horizons. The data in the TS was scaled to $[0,1]$ interval for a unique-scale comparison.

The methods used to evaluate the performance are basic ANN and SVR, EMD based ANN and SVR following multiple predictors approach but without any post-processing, two EMD based SVRs reported in [124, 133] and the proposed EMD-SVR.

To evaluate the performance of the models, three error measures are used in the experiment: normalized root mean square error (nRMSE), normalized mean absolute error (nMAE) and mean absolute scaled error (MASE):

$$\text{nRMSE} = \frac{1}{x_{max} - x_{min}} \sqrt{\mathrm{E}[(\hat{x} - x)^2]} \tag{4.6}$$

$$\text{nMAE} = \frac{1}{x_{max} - x_{min}} \sqrt{\mathrm{E}\,|\hat{x} - x|} \tag{4.7}$$

$$\text{MASE} = \frac{\sum_{t=1}^{n} |\hat{x} - x|}{\frac{n}{n-1} \sum_{i=2}^{n} |x_i - x_{i-1}|} \tag{4.8}$$

where $\hat{x}$ is the predicted data, x is the desired data and n is the number of data points in the TS.

The experiment was implemented on R3.1.3 with Intel Core i5 CPU. The SVM package is ‘e1071’ [136] and the EMD toolbox is ‘EMD’ [137].

4.4.1 Parameter Tuning

To determine whether a data at lag l is significant or not, it is necessary to calculate the PACF value of that particular lag $PACF(l)$. If $PACF(l) >= \pm\frac{1.96}{\sqrt{N}}$, the PACF value of l is considered significant and l is a salient lag, and vise versa. As shown in Figure 4.4, lags 1, 6, 9 and 14 have the PACF value exceeds the threshold $\pm\frac{1.96}{\sqrt{N}}$ (dashed line) and they are considered as salient lags.

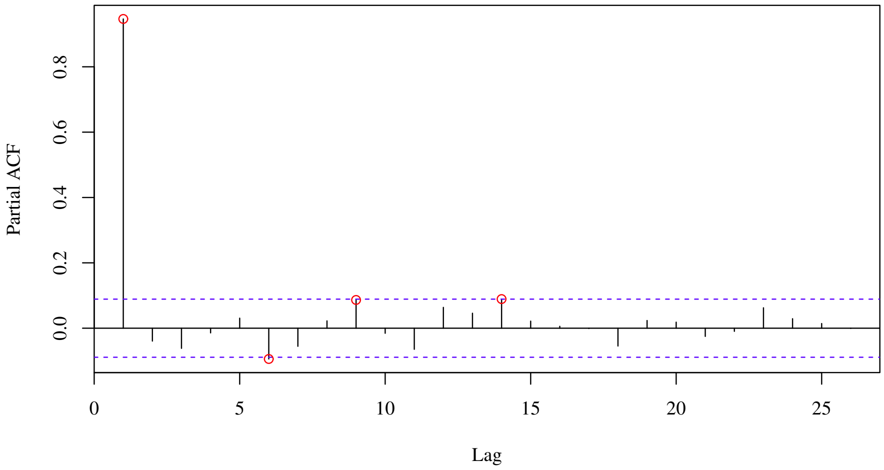


Figure 4.4 PACF Plot of a Wind Speed Time Series, dashed lines bound the threshold and circles annotate the salient lags whose values exceed the threshold.

In order to compare the importance of the feature selection, another two feature sets were formed: one is to use a full 2 days historical data and the other is to apply minimum Redundancy Maximum Relevance (mRMR) [138] feature selection to select the optimal features.

The mRMR feature selection is to maximize the relevance while minimize the redundancy. The relevance and redundancy are both represented by a mutual information (MI) which is defined as:

$$I(x,y) = -0.5\ln(1-\rho(x,y)^2) \tag{4.9}$$

where x is an input feature and y is the output data, I is the MI and ρ is the correlation coefficient between x and y. For continuous features, ρ is usually Pearson's correlation coefficient or Spearman's correlation coefficient [139, 140].

To find the maximum relevance, the input feature with highest MI is selected:

$$x_i = \underset{x_i \in X}{\text{argmax}} I(x_i, y) \tag{4.10}$$

Once the first feature is selected, the following feature should have the highest relevance with respect to y but the lowest relevance (redundancy) with respect to the selected features (store in set S):

$$x_j = \underset{x_j \in X}{\text{argmax}} I(x_j, y) - \frac{1}{|S|} \sum_{x_i \in S} I(x_j, x_i) \tag{4.11}$$

In this particular application, the above mRMR procedure is applied to the wind speed TS so that the 50% features (lags) having highest MI while lowest redundancy are selected from the full 2 days historical data (in total 48 features). That is, a total of 24 lags are selected as salient lags.

The training of SVR undergoes a two-pass grid search for optimal parameter tuning. In order to avoid over-fitting, a 5-fold cross validation (CV) is used in each iteration of the grid search. The parameters of the SVR used in the proposed EMD-SVR are: the penalty factor C, the tolerance ε and the parameter for the RBF kernel γ. The CV for TS is slightly different from general CV for classification and regression. The algorithm of k fold CV for TS is shown in Figure 4.5. The ANN architecture is single hidden layer feed-forward NN which is coded by Matlab's neural network toolbox. The hidden neuron number is selected by 5-fold CV. The BP algorithm is Levenberg-Marquardt.

4.5 Results and Discussion

4.5.1 Error Measure v.s. Forecasting Horizons

The error measures on selected forecasting horizons are tabulated in Tables 4.2, 4.3 and 4.4. The longer the forecasting horizon, the higher the forecasting error is. The nRMSE v.s. the forecasting horizon are plotted in Figure 4.6 on two forecasting methods. The medians of the errors are increasing approximately linearly with respect to the forecasting horizon. This linear positive correlation between error measures and the forecasting horizon was observed in other forecasting methods as well.

4.5.2 Improvement Contributed by EMD

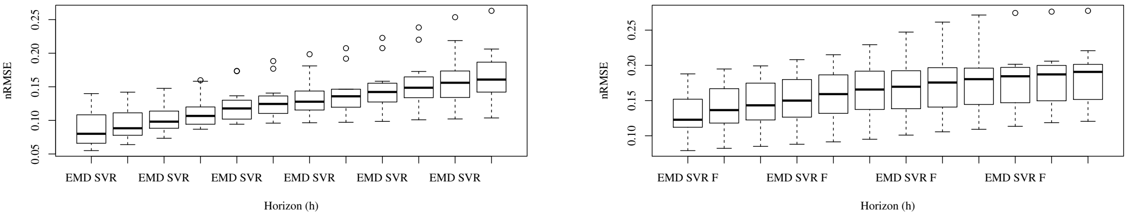
To check whether there is an improvement contributed by EMD, the basic ANN/SVR models without decomposition were compared with the multiple predictors configured EMD-

```
Inputs:
    $\mathbf{Y}$: training data
    $y$: corresponding future value
    $\Theta$: parameter sets of the forecasting model
    $k$: number of folds
Outputs:
    $\varepsilon$: error measure
Functions:
    $P(a,b)$: partition $a$ into $b$ pieces
    $f_t(a,b,c)$: train a model according to historical data $a$,
        future data $b$ and parameters $c$
    $f_s(m,d)$: test a new data $d$ with model $m$
    $\varepsilon(a,b)$: calculate error between $a$ and $b$

    $\{\mathbf{Y}_1,\ldots,\mathbf{Y}_{2k-1}\} = P(\mathbf{Y},2k-1)$
    $\{y_1,\ldots,y_{2k-1}\} = P(y,2k-1)$
Main Algorithm:
    FOR $i = 1$ to $k$
        $\mathbf{Y}_{trn} = \{\mathbf{Y}_i,\ldots,\mathbf{Y}_{i+k-2}\}$; $\mathbf{Y}_{val} = \mathbf{Y}_{i+k-1}$
        $y_{trn} = \{y_i,\ldots,y_{i+k-2}\}$; $y_{val} = y_{i+k-1}$
        FOR $\theta$ in $\Theta$
            $M = f_t(\mathbf{Y}_{trn},y_{trn},\theta)$
            $\varepsilon_{i,\theta} = \varepsilon(y_{val},f_s(M,\mathbf{Y}_{val}))$
        END
    END
    RETURN $\varepsilon$
```

Figure 4.5 Procedures of *k*-fold Cross Validation for Time Series

ANN/SVR models. The comparison was conducted by a Wilcoxon signed rank test [141] and the test statistics on 1, 6 and 12 hour ahead forecasting are shown in Table 4.5. For forecasting horizons up to 12 hour ahead, there are two jitter plots (Figure 4.7) to show the performance according to different error measures. From the comparison between EMD-ANN and ANN, it is shown that there is a significant under-performance for 1–3 hour ahead forecasting of EMD-ANN compared with ANN but the performance is comparable for 4–8 hour ahead except for PACF selected feature set. For 9–12 hour ahead forecasting, we can spot that the EMD-ANN outperformed the ANN with full lag feature set but for the selected feature sets, we cannot observe a performance improvement from ANN to EMD-ANN. The possible reason why EMD does not improve the performance of ANN is because that ANN is often trapped at local optima. For the decomposed TS, each ANN prediction will generate a forecasting error, and these errors are accumulated to become more significant than the ANN's prediction result.



(a) (b)

Figure 4.6 Boxplots of nRMSE v.s. Forecasting Horizons, (a) EMD-SVR, (b) the Proposed Model

Table 4.2 Comparison of Reported Methods and the Proposed EMD-SVR with full Feature Set for 1, 6 and 12 hour ahead Forecasting

		1 Hour							6 Hour							12 Hour						
	Dataset	ANN	SVR	EMD-ANN	EMD-SVR	'HL' [124]	'Single' [133]	Proposed	ANN	SVR	EMD-ANN	EMD-SVR	'HL' [124]	'Single' [133]	Proposed	ANN	SVR	EMD-ANN	EMD-SVR	'HL' [124]	'Single' [133]	Proposed
nRMSE	1	0.135	0.127	0.171	0.116	0.16	0.14	0.167	0.225	0.207	0.229	0.125	0.166	0.178	0.201	0.271	0.256	0.227	0.174	0.203	0.205	0.201
	2	0.158	0.085	0.133	0.066	0.128	0.073	0.079	0.236	0.131	0.249	0.102	0.138	0.101	0.095	0.234	0.137	0.19	0.104	0.148	0.135	0.121
	3	0.105	0.111	0.173	0.085	0.134	0.127	0.115	0.159	0.168	0.246	0.119	0.149	0.147	0.12	0.173	0.185	0.261	0.145	0.159	0.134	0.13
	4	0.093	0.093	0.133	0.08	0.167	0.097	0.118	0.201	0.191	0.172	0.109	0.183	0.178	0.154	0.25	0.249	0.205	0.148	0.205	0.233	0.192
	5	0.106	0.094	0.141	0.14	0.113	0.126	0.188	0.254	0.222	0.26	0.188	0.143	0.176	0.229	0.285	0.252	0.29	0.263	0.186	0.246	0.277
	6	0.133	0.08	0.176	0.055	0.103	0.065	0.109	0.185	0.167	0.217	0.112	0.133	0.112	0.176	0.288	0.188	0.22	0.165	0.161	0.161	0.177
	7	0.142	0.11	0.21	0.08	0.123	0.101	0.127	0.231	0.174	0.34	0.124	0.153	0.182	0.199	0.386	0.194	0.425	0.175	0.196	0.231	0.195
	8	0.172	0.087	0.215	0.067	0.134	0.083	0.136	0.243	0.17	0.229	0.096	0.152	0.156	0.154	0.236	0.192	0.218	0.105	0.161	0.172	0.156
	9	0.109	0.098	0.129	0.076	0.149	0.11	0.154	0.26	0.207	0.148	0.132	0.181	0.193	0.172	0.241	0.225	0.169	0.198	0.2	0.17	0.189
	10	0.091	0.067	0.18	0.1	0.156	0.083	0.117	0.155	0.15	0.131	0.128	0.181	0.127	0.16	0.185	0.194	0.164	0.156	0.199	0.191	0.221
	11	0.11	0.088	0.183	0.059	0.102	0.069	0.095	0.19	0.154	0.204	0.141	0.118	0.141	0.121	0.275	0.226	0.136	0.14	0.159	0.17	0.148
	12	0.137	0.113	0.108	0.123	0.173	0.109	0.15	0.239	0.218	0.212	0.177	0.183	0.206	0.184	0.385	0.24	0.202	0.206	0.2	0.216	0.202
nMAE	1	0.097	0.09	0.139	0.067	0.106	0.084	0.128	0.181	0.159	0.183	0.086	0.121	0.139	0.16	0.219	0.199	0.176	0.133	0.16	0.161	0.16
	2	0.122	0.06	0.107	0.045	0.104	0.053	0.057	0.185	0.105	0.22	0.078	0.112	0.077	0.071	0.195	0.113	0.155	0.076	0.123	0.104	0.095
	3	0.067	0.073	0.142	0.052	0.105	0.077	0.078	0.118	0.128	0.211	0.085	0.118	0.113	0.089	0.139	0.148	0.224	0.107	0.129	0.098	0.096
	4	0.071	0.067	0.115	0.057	0.145	0.07	0.089	0.153	0.149	0.142	0.088	0.157	0.149	0.121	0.198	0.202	0.17	0.113	0.171	0.194	0.159
	5	0.081	0.075	0.111	0.1	0.091	0.093	0.139	0.212	0.176	0.212	0.15	0.114	0.146	0.165	0.231	0.199	0.249	0.213	0.158	0.199	0.193
	6	0.106	0.06	0.153	0.039	0.084	0.051	0.085	0.148	0.135	0.18	0.089	0.109	0.09	0.135	0.245	0.152	0.181	0.136	0.125	0.125	0.14
	7	0.11	0.082	0.175	0.057	0.096	0.077	0.095	0.182	0.144	0.295	0.096	0.12	0.154	0.163	0.345	0.16	0.376	0.132	0.161	0.187	0.159
	8	0.147	0.065	0.176	0.047	0.109	0.061	0.118	0.196	0.13	0.187	0.072	0.122	0.122	0.128	0.188	0.157	0.175	0.083	0.129	0.139	0.129
	9	0.084	0.074	0.096	0.051	0.123	0.081	0.124	0.219	0.152	0.106	0.103	0.144	0.148	0.141	0.184	0.173	0.133	0.147	0.162	0.137	0.145
	10	0.076	0.052	0.15	0.072	0.135	0.066	0.096	0.122	0.115	0.105	0.094	0.154	0.101	0.133	0.141	0.154	0.13	0.118	0.171	0.162	0.184
	11	0.083	0.067	0.166	0.044	0.081	0.053	0.076	0.153	0.124	0.18	0.102	0.091	0.116	0.096	0.221	0.189	0.111	0.113	0.123	0.14	0.117
	12	0.106	0.087	0.088	0.093	0.141	0.08	0.117	0.193	0.166	0.167	0.143	0.149	0.157	0.155	0.321	0.196	0.165	0.171	0.162	0.173	0.166
MASE	1	1.814	1.675	2.582	1.249	1.98	1.556	2.381	3.414	2.996	3.447	1.623	2.274	2.611	3.007	4.123	3.748	3.325	2.511	3.015	3.039	3.01
	2	2.209	1.08	1.922	0.818	1.882	0.951	1.025	3.443	1.949	4.079	1.448	2.087	1.435	1.327	3.667	2.121	2.918	1.434	2.305	1.947	1.786
	3	1.128	1.219	2.383	0.869	1.767	1.295	1.314	1.969	2.14	3.533	1.418	1.972	1.894	1.49	2.501	2.674	4.038	1.927	2.331	1.777	1.734
	4	1.457	1.377	2.382	1.175	3.002	1.444	1.847	3.09	3.005	2.866	1.773	3.174	3.003	2.445	3.994	4.074	3.431	2.281	3.439	3.911	3.208
	5	1.373	1.266	1.873	1.696	1.541	1.571	2.357	3.721	3.1	3.72	2.633	1.994	2.567	2.905	4.081	3.519	4.397	3.761	2.787	3.525	3.409
	6	2.053	1.16	2.953	0.752	1.622	0.981	1.648	2.93	2.691	3.579	1.76	2.163	1.789	2.673	4.744	2.947	3.508	2.636	2.413	2.42	2.703
	7	1.395	1.037	2.213	0.718	1.209	0.972	1.204	2.298	1.82	3.727	1.213	1.513	1.947	2.058	4.357	2.02	4.74	1.661	2.037	2.36	2.01
	8	2.546	1.127	3.06	0.814	1.894	1.064	2.046	3.398	2.253	3.245	1.255	2.121	2.116	2.224	3.279	2.744	3.061	1.455	2.245	2.428	2.253
	9	1.377	1.213	1.575	0.832	2.034	1.332	2.04	3.442	2.396	1.661	1.616	2.265	2.324	2.224	2.888	2.72	2.088	2.31	2.54	2.153	2.276
	10	1.622	1.102	3.183	1.531	2.864	1.407	2.043	2.681	2.526	2.313	2.07	3.401	2.23	2.932	3.105	3.382	2.858	2.587	3.746	3.55	4.03
	11	1.479	1.189	2.961	0.79	1.433	0.936	1.349	2.659	2.167	3.135	1.778	1.588	2.026	1.672	3.828	3.261	1.919	1.954	2.124	2.419	2.023
	12	1.47	1.206	1.225	1.289	1.967	1.117	1.634	2.733	2.353	2.371	2.03	2.11	2.227	2.202	4.699	2.871	2.411	2.498	2.374	2.538	2.432

For EMD-SVR and SVR, the error measures on 1 hour ahead forecasting cannot show significant differences between these two methods but from 2 hour ahead forecasting onwards, there are significant improvements of EMD-SVR over SVR.

The possible reason why the proposed EMD-SVR model had a better performance than the RBF kernel SVR model is that after EMD, the input space expanded because more data points were generated and since the IMFs are orthogonal to each other [120], the output

Table 4.3 Comparison of Reported Methods and the Proposed EMD-SVR with PACF selected Feature Set for 1, 6 and 12 hour ahead Forecasting

		1 Hour							6 Hour							12 Hour						
	Dataset	ANN	SVR	EMD-ANN	EMD-SVR	‘HL’ [124]	‘Single’ [133]	Proposed	ANN	SVR	EMD-ANN	EMD-SVR	‘HL’ [124]	‘Single’ [133]	Proposed	ANN	SVR	EMD-ANN	EMD-SVR	‘HL’ [124]	‘Single’ [133]	Proposed
nRMSE	1	0.128	0.093	0.161	0.116	0.174	0.201	0.142	0.193	0.167	0.2	0.137	0.179	0.222	0.162	0.268	0.215	0.249	0.171	0.199	0.235	0.19
	2	0.121	0.082	0.135	0.053	0.187	0.089	0.064	0.197	0.13	0.19	0.09	0.176	0.099	0.102	0.167	0.135	0.22	0.094	0.166	0.12	0.136
	3	0.121	0.102	0.187	0.079	0.134	0.127	0.127	0.172	0.138	0.316	0.133	0.132	0.147	0.147	0.22	0.152	0.418	0.126	0.157	0.134	0.134
	4	0.085	0.065	0.134	0.096	0.138	0.095	0.095	0.19	0.167	0.266	0.15	0.141	0.157	0.157	0.288	0.225	0.24	0.191	0.169	0.198	0.198
	5	0.085	0.078	0.143	0.115	0.17	0.126	0.126	0.267	0.162	0.296	0.154	0.177	0.176	0.176	0.354	0.21	0.304	0.201	0.21	0.246	0.246
	6	0.093	0.071	0.357	0.061	0.096	0.065	0.065	0.195	0.154	0.48	0.115	0.126	0.112	0.112	0.357	0.172	0.221	0.154	0.155	0.161	0.161
	7	0.113	0.107	0.181	0.086	0.111	0.099	0.099	0.201	0.167	0.46	0.138	0.145	0.189	0.189	0.516	0.192	0.235	0.163	0.186	0.241	0.241
	8	0.129	0.079	0.223	0.068	0.12	0.083	0.083	0.257	0.176	0.33	0.127	0.162	0.156	0.156	0.338	0.203	0.281	0.175	0.194	0.172	0.172
	9	0.087	0.08	0.134	0.081	0.128	0.102	0.102	0.217	0.189	0.371	0.159	0.217	0.167	0.167	0.267	0.228	0.225	0.169	0.223	0.17	0.17
	10	0.075	0.061	0.162	0.085	0.134	0.083	0.08	0.168	0.121	0.204	0.169	0.384	0.127	0.128	0.157	0.189	0.217	0.255	0.419	0.191	0.189
	11	0.104	0.076	0.105	0.06	0.085	0.069	0.069	0.179	0.151	0.238	0.123	0.133	0.141	0.141	0.259	0.197	0.229	0.151	0.164	0.17	0.17
	12	0.099	0.096	0.135	0.083	0.111	0.109	0.11	0.236	0.203	0.243	0.186	0.175	0.206	0.21	0.289	0.24	0.364	0.202	0.177	0.216	0.217
nMAE	1	0.094	0.06	0.132	0.064	0.126	0.154	0.09	0.147	0.134	0.166	0.088	0.136	0.176	0.123	0.212	0.169	0.213	0.121	0.158	0.18	0.149
	2	0.092	0.055	0.106	0.035	0.146	0.065	0.045	0.152	0.099	0.158	0.064	0.145	0.076	0.079	0.132	0.112	0.188	0.074	0.139	0.095	0.107
	3	0.09	0.06	0.156	0.05	0.104	0.077	0.077	0.137	0.096	0.276	0.091	0.103	0.113	0.113	0.182	0.116	0.38	0.089	0.125	0.098	0.098
	4	0.068	0.048	0.113	0.063	0.109	0.069	0.069	0.151	0.131	0.222	0.121	0.108	0.13	0.13	0.237	0.186	0.206	0.159	0.135	0.165	0.165
	5	0.063	0.058	0.113	0.077	0.129	0.093	0.093	0.223	0.132	0.258	0.125	0.147	0.146	0.146	0.286	0.174	0.254	0.171	0.174	0.199	0.199
	6	0.071	0.054	0.314	0.045	0.075	0.051	0.051	0.149	0.126	0.437	0.094	0.101	0.09	0.09	0.308	0.134	0.179	0.115	0.12	0.125	0.125
	7	0.082	0.08	0.14	0.065	0.083	0.074	0.074	0.158	0.136	0.409	0.113	0.116	0.162	0.162	0.469	0.163	0.181	0.132	0.152	0.195	0.195
	8	0.108	0.057	0.186	0.05	0.094	0.061	0.061	0.218	0.137	0.264	0.097	0.124	0.122	0.122	0.293	0.163	0.23	0.138	0.143	0.139	0.139
	9	0.068	0.06	0.101	0.056	0.102	0.07	0.07	0.171	0.14	0.329	0.12	0.167	0.132	0.132	0.208	0.175	0.174	0.137	0.176	0.136	0.136
	10	0.059	0.048	0.136	0.059	0.114	0.066	0.064	0.138	0.092	0.166	0.117	0.253	0.101	0.102	0.123	0.162	0.176	0.172	0.278	0.162	0.16
	11	0.077	0.057	0.086	0.045	0.067	0.053	0.053	0.147	0.121	0.196	0.098	0.107	0.116	0.116	0.218	0.166	0.189	0.123	0.131	0.14	0.14
	12	0.074	0.074	0.099	0.063	0.088	0.08	0.082	0.184	0.162	0.193	0.149	0.14	0.157	0.158	0.224	0.195	0.308	0.165	0.132	0.173	0.173
MASE	1	1.754	1.125	2.451	1.19	2.344	2.871	1.674	2.776	2.526	3.137	1.656	2.571	3.321	2.322	3.996	3.181	4.019	2.28	2.982	3.399	2.809
	2	1.652	0.989	1.913	0.638	2.625	1.18	0.813	2.816	1.832	2.925	1.187	2.7	1.419	1.466	2.489	2.098	3.534	1.39	2.621	1.781	2.003
	3	1.51	1.009	2.615	0.835	1.748	1.295	1.295	2.296	1.607	4.623	1.529	1.727	1.894	1.894	3.289	2.094	6.861	1.613	2.259	1.777	1.777
	4	1.4	0.995	2.341	1.31	2.25	1.423	1.423	3.043	2.647	4.478	2.45	2.181	2.626	2.626	4.78	3.754	4.142	3.202	2.728	3.316	3.316
	5	1.065	0.987	1.917	1.309	2.175	1.571	1.571	3.917	2.322	4.536	2.187	2.58	2.567	2.567	5.05	3.081	4.5	3.031	3.081	3.525	3.525
	6	1.367	1.041	6.085	0.868	1.445	0.981	0.981	2.953	2.502	8.675	1.872	2.013	1.789	1.789	5.959	2.591	3.459	2.225	2.32	2.42	2.42
	7	1.042	1.016	1.766	0.824	1.052	0.935	0.935	1.996	1.716	5.163	1.431	1.47	2.04	2.04	5.912	2.063	2.287	1.66	1.914	2.458	2.458
	8	1.871	0.992	3.238	0.863	1.629	1.064	1.064	3.776	2.378	4.582	1.683	2.149	2.116	2.116	5.11	2.841	4.011	2.413	2.496	2.428	2.428
	9	1.113	0.983	1.656	0.926	1.687	1.157	1.157	2.687	2.207	5.185	1.89	2.627	2.081	2.081	3.259	2.753	2.735	2.153	2.764	2.137	2.137
	10	1.248	1.014	2.888	1.249	2.428	1.407	1.369	3.031	2.019	3.648	2.568	5.562	2.23	2.243	2.707	3.552	3.864	3.771	6.103	3.55	3.507
	11	1.376	1.018	1.533	0.803	1.195	0.936	0.936	2.563	2.11	3.412	1.712	1.86	2.026	2.026	3.763	2.877	3.265	2.124	2.266	2.419	2.419
	12	1.023	1.031	1.381	0.87	1.222	1.117	1.143	2.61	2.292	2.735	2.113	1.988	2.227	2.239	3.284	2.855	4.511	2.417	1.938	2.538	2.525

of the inner product in the regression function is less complex than the SVR without EMD decomposed inputs. Better generalization is usually associated with simpler regression function which has less possibility of over-fitting and therefore, EMD has more positive impact on SVR based forecasting methods.

4.5.3 Comparison on Reported Methods and the Proposed Method

The proposed EMD-SVR with input vector reconstruction was then compared with the conventional EMD-SVR and two reported EMD-SVR methods in [124] (named ‘HL’) and [133] (named ‘Single’).

Friedman rank sum test [141, 142] was employed to check the performance among these four methods. Table 4.6 tabulates the test statistics for 1, 6 and 12 hour ahead forecasting. It is shown that except for 12 hour ahead forecasting on full lag feature, there are significant performance difference among these EMD-SVR based methods.

A post-hoc Nemenyi test was used to distinguish the performance differences among these methods. The test result is shown in Table 4.7 for 1, 6 and 12 hour ahead forecasting. A jitter plot for 1–12 hour forecasting horizon is shown in Figure 4.8. The proposed EMD-

Table 4.4 Comparison of Reported Methods and the Proposed EMD-SVR with mRMR selected Feature Set for 1, 6 and 12 hour ahead Forecasting

		1 Hour							6 Hour							12 Hour						
	Dataset	ANN	SVR	EMD-ANN	EMD-SVR	'HL' [124]	'Single' [133]	Proposed	ANN	SVR	EMD-ANN	EMD-SVR	'HL' [124]	'Single' [133]	Proposed	ANN	SVR	EMD-ANN	EMD-SVR	'HL' [124]	'Single' [133]	Proposed
nRMSE	1	0.161	0.119	0.169	0.118	0.17	0.14	0.16	0.221	0.199	0.238	0.126	0.174	0.178	0.19	0.258	0.247	0.218	0.172	0.199	0.205	0.198
	2	0.162	0.083	0.148	0.063	0.126	0.073	0.078	0.226	0.131	0.184	0.09	0.138	0.101	0.093	0.241	0.143	0.168	0.1	0.148	0.135	0.111
	3	0.204	0.106	0.193	0.082	0.128	0.127	0.119	0.256	0.165	0.279	0.102	0.143	0.147	0.122	0.211	0.185	0.366	0.119	0.148	0.134	0.131
	4	0.081	0.079	0.146	0.087	0.164	0.097	0.104	0.184	0.174	0.157	0.107	0.176	0.178	0.137	0.278	0.217	0.207	0.155	0.2	0.233	0.16
	5	0.093	0.089	0.133	0.144	0.11	0.126	0.173	0.211	0.209	0.323	0.176	0.135	0.176	0.208	0.245	0.246	0.323	0.255	0.183	0.246	0.274
	6	0.125	0.078	0.195	0.053	0.119	0.065	0.105	0.199	0.159	0.174	0.11	0.141	0.112	0.165	0.32	0.191	0.219	0.16	0.162	0.161	0.176
	7	0.147	0.111	0.233	0.078	0.119	0.099	0.114	0.207	0.177	0.276	0.124	0.151	0.189	0.179	0.335	0.19	0.387	0.16	0.199	0.241	0.192
	8	0.148	0.084	0.208	0.063	0.132	0.083	0.125	0.243	0.175	0.251	0.097	0.152	0.156	0.148	0.249	0.186	0.206	0.104	0.163	0.172	0.144
	9	0.1	0.095	0.133	0.074	0.149	0.102	0.134	0.215	0.202	0.164	0.112	0.186	0.167	0.163	0.269	0.227	0.228	0.174	0.204	0.17	0.172
	10	0.075	0.064	0.18	0.1	0.152	0.08	0.108	0.151	0.144	0.147	0.128	0.176	0.128	0.153	0.209	0.19	0.169	0.158	0.193	0.189	0.21
	11	0.107	0.086	0.119	0.054	0.102	0.069	0.085	0.187	0.155	0.159	0.128	0.114	0.141	0.124	0.301	0.221	0.149	0.118	0.158	0.17	0.145
	12	0.103	0.11	0.125	0.115	0.174	0.109	0.141	0.23	0.223	0.212	0.175	0.184	0.206	0.181	0.272	0.243	0.211	0.218	0.199	0.216	0.182
nMAE	1	0.125	0.083	0.138	0.066	0.123	0.084	0.117	0.18	0.156	0.192	0.088	0.134	0.139	0.149	0.206	0.186	0.178	0.133	0.158	0.161	0.158
	2	0.128	0.059	0.121	0.043	0.104	0.053	0.056	0.179	0.103	0.15	0.068	0.111	0.077	0.07	0.205	0.117	0.134	0.073	0.122	0.104	0.086
	3	0.187	0.072	0.161	0.048	0.1	0.077	0.077	0.228	0.131	0.246	0.068	0.111	0.113	0.089	0.179	0.148	0.336	0.086	0.117	0.098	0.097
	4	0.062	0.058	0.12	0.059	0.142	0.07	0.078	0.14	0.13	0.129	0.087	0.152	0.149	0.105	0.222	0.172	0.169	0.129	0.168	0.194	0.126
	5	0.071	0.069	0.106	0.101	0.088	0.093	0.128	0.168	0.17	0.277	0.138	0.108	0.146	0.15	0.191	0.194	0.285	0.2	0.154	0.199	0.189
	6	0.101	0.058	0.155	0.038	0.097	0.051	0.081	0.167	0.128	0.145	0.088	0.113	0.09	0.129	0.282	0.155	0.179	0.131	0.127	0.125	0.14
	7	0.118	0.083	0.202	0.056	0.093	0.074	0.087	0.16	0.146	0.236	0.098	0.118	0.162	0.144	0.296	0.158	0.333	0.129	0.164	0.195	0.153
	8	0.122	0.062	0.174	0.044	0.11	0.061	0.107	0.201	0.133	0.201	0.074	0.122	0.122	0.129	0.199	0.152	0.168	0.082	0.125	0.139	0.119
	9	0.076	0.072	0.099	0.047	0.124	0.07	0.103	0.169	0.152	0.118	0.084	0.148	0.132	0.131	0.207	0.178	0.188	0.129	0.166	0.136	0.135
	10	0.059	0.051	0.15	0.072	0.133	0.064	0.085	0.115	0.112	0.119	0.092	0.149	0.102	0.124	0.164	0.153	0.135	0.12	0.164	0.16	0.172
	11	0.08	0.065	0.103	0.041	0.082	0.053	0.068	0.147	0.125	0.127	0.092	0.091	0.116	0.098	0.243	0.186	0.121	0.098	0.126	0.14	0.114
	12	0.078	0.083	0.096	0.086	0.142	0.08	0.107	0.171	0.173	0.166	0.139	0.149	0.157	0.151	0.207	0.196	0.17	0.184	0.159	0.173	0.148
MASE	1	2.321	1.548	2.573	1.236	2.296	1.556	2.178	3.4	2.936	3.614	1.659	2.526	2.611	2.817	3.877	3.517	3.357	2.516	2.978	3.039	2.979
	2	2.301	1.072	2.174	0.782	1.88	0.951	1.012	3.323	1.912	2.784	1.257	2.063	1.435	1.3	3.849	2.192	2.523	1.379	2.295	1.947	1.624
	3	3.134	1.218	2.704	0.81	1.682	1.295	1.301	3.82	2.184	4.123	1.144	1.859	1.894	1.489	3.239	2.665	6.069	1.545	2.116	1.777	1.746
	4	1.277	1.206	2.481	1.219	2.94	1.444	1.619	2.819	2.617	2.601	1.752	3.062	3.003	2.128	4.472	3.473	3.396	2.592	3.393	3.911	2.541
	5	1.204	1.175	1.799	1.712	1.482	1.571	2.17	2.956	2.983	4.858	2.432	1.904	2.567	2.638	3.381	3.429	5.037	3.53	2.732	3.525	3.351
	6	1.96	1.131	3.004	0.732	1.874	0.981	1.575	3.315	2.55	2.89	1.751	2.252	1.789	2.569	5.46	2.997	3.454	2.536	2.458	2.42	2.698
	7	1.486	1.048	2.55	0.702	1.173	0.935	1.093	2.024	1.845	2.98	1.241	1.494	2.04	1.818	3.736	1.999	4.203	1.628	2.073	2.458	1.929
	8	2.127	1.074	3.031	0.756	1.905	1.064	1.867	3.48	2.31	3.48	1.289	2.112	2.116	2.233	3.484	2.654	2.929	1.429	2.185	2.428	2.073
	9	1.258	1.189	1.626	0.774	2.04	1.157	1.695	2.664	2.39	1.864	1.319	2.328	2.081	2.059	3.253	2.791	2.956	2.032	2.609	2.137	2.124
	10	1.249	1.083	3.193	1.536	2.82	1.369	1.805	2.526	2.473	2.617	2.033	3.275	2.243	2.72	3.599	3.357	2.974	2.627	3.61	3.507	3.786
	11	1.422	1.155	1.831	0.722	1.454	0.936	1.203	2.565	2.181	2.218	1.6	1.592	2.026	1.71	4.199	3.211	2.084	1.694	2.17	2.419	1.977
	12	1.091	1.16	1.338	1.194	1.976	1.117	1.482	2.431	2.452	2.353	1.977	2.109	2.227	2.144	3.034	2.867	2.489	2.688	2.32	2.538	2.166

Table 4.5 Wilcoxon Signed Rank Test for 1, 6 and 12 hour ahead Forecasting to Compare EMD-ANN/SVR against ANN/SVR. (a) EMD-ANN v.s. ANN and (b) EMD-SVR v.s. SVR. p is the p value and z is the votes. If $p < 0.05$, there is 95% confident to conclude that there is a significant better improvement of the EMD based method.

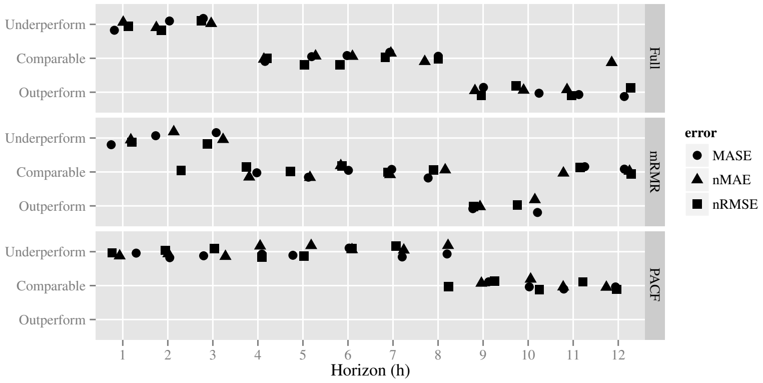
(a)

Horizon (h)		nRMSE		nMAE		MASE	
		p	z	p	z	p	z
Full	1	0.998	5	0.998	5	0.998	5
	6	0.604	36	0.765	30	0.741	31
	12	0.032	63	0.055	60	0.046	61
PACF	1	1	0	1	0	1	0
	6	1	1	1	0	1	0
	12	0.367	44	0.367	44	0.396	43
mRMR	1	0.998	6	0.997	7	0.997	7
	6	0.339	45	0.425	42	0.425	42
	12	0.17	52	0.17	52	0.117	55

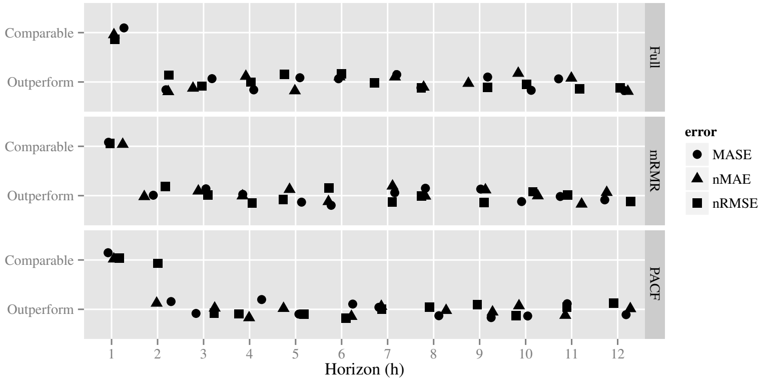
(b)

Horizon (h)		nRMSE		nMAE		MASE	
		p	z	p	z	p	z
Full	1	0.133	54	0.055	60	0.133	54
	6	0	78	0	78	0	78
	12	0	77	0	77	0	77
PACF	1	0.545	38	0.212	50	0.311	46
	6	0.013	67	0.006	70	0.006	70
	12	0.017	66	0.001	76	0.001	76
mRMR	1	0.212	50	0.102	56	0.17	52
	6	0	78	0	78	0	78
	12	0	77	0	77	0	77

SVR with feature reconstruction has comparable performance as the 'HL' method [124] and the 'Single' method [133]. The proposed EMD-SVR has significant performance difference



(a)



(b)

Figure 4.7 Wilcoxon Signed Rank Test to Compare (a) EMD-ANN v.s. ANN and (b) EMD-SVR v.s. SVR. The points are jittered to show whether the EMD-ANN/SVR outperforms/underperforms or is comparable to ANN/SVR with respect to forecasting horizons.

Table 4.6 Friedman Test on the Proposed EMD-SVR Method and three other Reported Methods. p is the p value and χ^2 is the statistics. If $p < 0.05$, there is 95% confident to conclude that there is a significant difference between the contestants.

Horizon (h)		nRMSE		nMAE		MASE	
		p	χ^2	p	χ^2	p	χ^2
Full	1	0	23.3	0	25.1	0	25.1
	6	0.04	8.3	0.021	9.7	0.021	9.7
	12	0.139	5.5	0.151	5.3	0.151	5.3
PACF	1	0	26.464	0	33.107	0	33.107
	6	0.034	8.679	0.016	10.393	0.016	10.393
	12	0.016	10.393	0.014	10.607	0.014	10.607
mRMR	1	0	25.3	0	26.7	0	26.7
	6	0.002	14.5	0	18	0	18
	12	0.021	9.7	0.037	8.5	0.037	8.5

to the conventional EMD-SVR for 1–5 hour ahead forecasting but they are comparable for 9–11 hour ahead.

Table 4.7 p value of Post-hoc Nemenyi Test on the three Reported Methods v.s. the Proposed Method. p is the p value and if $p < 0.05$, there is 95% confident to conclude that there is a significant difference between the contestants.

	Horizon (h)	nRMSE			nMAE			MASE		
		EMD-SVR	'HL' [124]	'Single' [133]	EMD-SVR	'HL' [124]	'Single' [133]	EMD-SVR	'HL' [124]	'Single' [133]
Full	1	0.000	0.999	0.023	0.000	1.000	0.014	0.000	1.000	0.014
	6	0.083	0.989	0.999	0.036	0.989	0.999	0.036	0.989	0.999
	12	0.685	0.922	0.685	0.859	0.922	0.485	0.859	0.922	0.485
PACF	1	0.303	0.009	0.965	0.036	0.023	0.965	0.036	0.023	0.965
	6	0.083	1.000	0.999	0.055	0.999	0.999	0.055	0.999	0.999
	12	0.055	0.999	0.999	0.083	1.000	0.965	0.083	1.000	0.965
mRMR	1	0.009	0.485	0.168	0.005	0.585	0.083	0.005	0.585	0.083
	6	0.055	0.999	0.685	0.003	0.922	0.989	0.003	0.922	0.989
	12	0.389	0.685	0.685	0.778	0.685	0.303	0.778	0.685	0.303

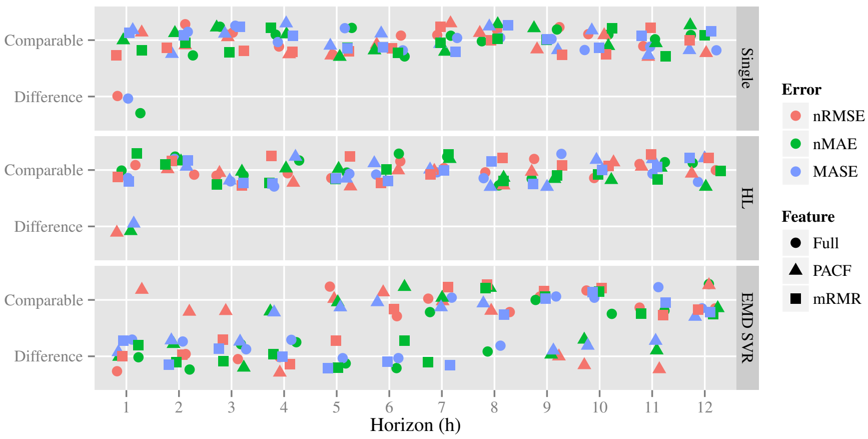


Figure 4.8 Jitter Plots of the Nemenyi Test on the three Reported Methods v.s. the Proposed Method. The points are jittered to show whether the proposed method is comparable or significant differ from the respective reported method.

4.5.4 Comparison on Different Feature Selection Methods

To check whether the performance is affected by the feature selection, the three different feature selection methods were evaluated. Friedman test statistics in Table 4.8 show that a post-hoc Nemenyi is required to further refine the comparison.

As shown in Table 4.9 and Figure 4.9, for shorter horizon forecasting (1–5 hour ahead), PACF selected features resulted better performance than the unselected features and comparable performance as the mRMR selected features. For longer horizon (6–12 hour ahead),

Table 4.8 Friedman Test on the three Feature Sets. p is the p value and χ^2 is the statistics. If $p < 0.05$, there is 95% confident to conclude that there is a significant difference between the contestants.

Horizon (h)	nRMSE		nMAE		MASE	
	p	χ^2	p	χ^2	p	χ^2
1	0	16.667	0	24	0.001	14
6	0.097	4.667	0.125	4.167	0.009	9.5
12	0.097	4.667	0.017	8.167	0.046	6.167

the significance in difference was not always strong but occasionally (9, 11 and 12 hour ahead) PACF outperformed mRMR based on nMAE.

Table 4.9 p value of Post-hoc Nemenyi Test on the Unselected Feature and mRMR Selected Feature sets v.s. the PACF Selected Feature Set. p is the p value and if $p < 0.05$, there is 95% confident to conclude that there is a significant difference between the contestants.

	Horizon (h)	nRMSE		nMAE		MASE	
		Full	mRMR	Full	mRMR	Full	mRMR
PACF	1	0.000	0.102	0.000	0.038	0.001	0.438
	6	0.912	0.232	0.564	0.564	0.006	0.438
	12	0.912	0.232	0.912	0.022	0.977	0.102

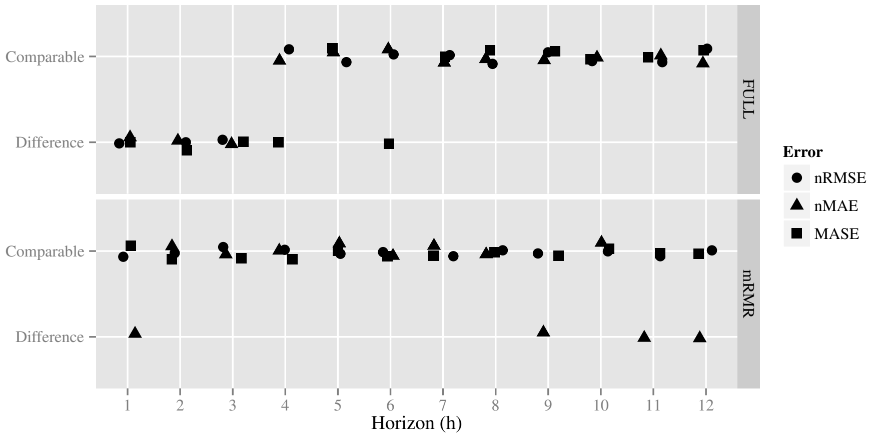


Figure 4.9 Jitter Plots of the Nemenyi Test on the three Feature Sets. The points are jittered to show whether the proposed PACF selected feature is comparable or significant differ from the full lag feature or the mRMR selected feature.

4.6 Concluding Remarks

This chapter has proposed an EMD-SVR model for wind speed TS forecasting. The EMD has been used for TS decomposition. It decomposed the time series into a collection of IMFs and a residue. An input vector consisting l historical data from each IMF and residue has been trained and tested by an SVR. The proposed EMD-SVR model has been evaluated with a collection of wind speed datasets.

The forecasting error is linearly proportional to the forecasting horizon (answer to RQ-4.1).

The EMD has not improved the performance of ANN but it has improved the performance of SVR (answer to RQ-4.2).

The comparisons between the proposed model and some reported models have shown that the proposed model outperformed some reported methods [124, 133] in terms of nRMSE, nMAE and MASE. The proposed model had a comparable performance compared with the EMD-SVR with multiple predictors approach (answer to RQ-4.3). In addition, the proposed method only require one SVR to do the training and prediction instead of multiple SVR modules, the computation time is faster.

The PACF based feature selection has further simplified the model without degrading the performance (answer to RQ-4.4).

Chapter 5

Ensemble Empirical Mode Decomposition based TS Forecasting

As shown in Chapter 4, the proposed EMD-SVR method had a relatively good performance on the short-term wind speed forecasting. The EMD did not improve the performance of ANN method. In this chapter, we propose to enhance the performance of EMD-ANN with an AdaBoost ensemble method.

For other types of TS, or wind speed TS recorded from different regions, there may be a mode mixing problem on the original EMD algorithm as stated by the authors of EMD and other researchers. In order to overcome the mode mixing problem, some ensemble versions of EMD are discussed in this chapter.

Three research questions (RQs) are raised and will be answered in this chapter:

RQ-5.1 Does AdaBoost improve the performance of EMD-ANN on TS forecasting?

RQ-5.2 How is the improvement of noise assisted ensemble EMD-SVR over EMD-SVR?

RQ-5.3 What is the best noise assisted ensemble EMD method among the three methods reviewed in this chapter?

5.1 AdaBoost-Empirical Mode Decomposition (ABEMD)

AdaBoost is discussed in Section 3.1, which is proved to work best with ‘weak learners’. A ‘weak learner’ is a supervised learning method that is unstable and usually give unstable, low accurate result (slightly better than random guess [95]). However, ‘weak learners’ can introduce diversity to the ensemble learning framework, thus enhance the performance.

There are a lot of ensemble forecasting methods based on AdaBoost in the literature [81, 143–146] with applications in face detection [143], debris flow disaster prediction [144] and critical frequency of the ionospheric prediction [145], etc.

AdaBoost employed for TS forecasting was shown in [146] and [81], which reported an AdaBoost-recurrent neural network (RNN) and an AdaBoost-ANN, respectively.

In this section, we propose an EMD based AdaBoost-ANN method for wind speed forecasting. In Chapter 4, we found that EMD did not improve the performance of ANN, therefore we would like to use AdaBoost based ensemble method to improve the performance of EMD.

There are several AdaBoost algorithms discussed in Section 3.3. In this chapter, we propose to use AdaBoost.R2 [147] for the TS forecasting problem. The AdaBoost.R2 algorithm is shown in Figure 5.1.

```
Inputs:
    $\{\mathbf{x}\} := m \times n$: training data
    $y$: corresponding future value
    $I_m$: maximum number of iterations
    $f(\cdot)$: base learning algorithm
    $\mathbf{w} := m \times 1$: weight vector
Initialize:
    $\mathbf{w}^1 = \{w_i^1 | 1/m, 1 \leq i \leq m\}$
    $\overline{L} = 0$
Main Algorithm:
    FOR $t = 1$ to $I_m$ & $\overline{L} < 0.5$
        Sample with replacement from $\mathbf{x}$ with distribution $\mathbf{w} \Rightarrow \mathbf{x}^t$
        Obtain a trained model $h_t = f(\mathbf{x}^t)$
        Calculate maximum loss $L_m = \max |y_i - h_t(x_i)|$
        Calculate individual loss $L_i = \frac{|y_i - h_t(x_i)|}{L_m}$
        Calculate weighted loss $\overline{L} = \sum_{i=1}^{m} L_i w_i^t$
        Update weight vector $w_i^{t+1} = \frac{w_i^t \beta_t^{(1-L_i)}}{Z_t}$,
        where $\beta_t = \frac{\overline{L}}{1-\overline{L}}$ and $Z_t$ is a normalization factor.
    END
    RETURN $\hat{y}^* = \text{median}(h_t(x^*) * \ln \frac{1}{\beta_t})$, $t = 1 \cdots I_m$.
    where $x^*$ is a testing data and $\hat{y}^*$ is the predicted output.
```

Figure 5.1 Algorithm of AdaBoost.R2 [147, 148]

The proposed ABEMD-ANN model for wind speed TS forecasting model first applies EMD to decompose the TS into a collection of IMFs and a residue. Secondly, it employs

ANN with BP as the 'weak learner' to construct a set of AdaBoost models and trains each AdaBoost-ANN model on each IMF and the residue. Then the trained models are applied to the decomposed testing data to obtain the predicted outputs. Finally, the proposed model sums the predicted outputs to obtain the final prediction.

5.2 ABEMD-ANN for Short term Wind Speed Forecasting

Twelve wind speed TS were obtained from NDBC [135]. The wind speed was recorded and averaged over a 1 hour interval. The length of each wind speed TS is around one month. The data were partitioned into 70% for training and 30% for testing. The input vector was formed by 12 hour historical data points. The forecasting horizons were 1, 3 and 5 hour ahead. The wind speed data was normalized into $[0,1]$ interval. Thanks for the comment. There are 12 input neurons and to initialize the NN architecture, I choose to use $2\times 12 = 24$ hidden neurons. I alter the hidden neuron number within 24 ± 12 by step of 2 with 5-fold cross validation, but it does not have significantly better performance as 24.

The ANN is configured as a three layer neural network with 12 neurons in the input layer corresponding to 12 historical data. To determine the number of hidden neurons, an initial $2\times 12 = 24$ hidden neuron number is used and the optimal number is searched within 24 ± 12 by step of 2 with 5-fold cross validation, but others do not have significantly better performance as 24. The output layer has 1 neuron. The BP algorithm used is Levenberg-Marquardt BP. The activation function of the hidden layer neuron is hyperbolic tangent (tanh) as shown in (3.25). The error metric is nRMSE shown in (4.6). For the AdaBoost, the maximum iteration is 100.

The proposed ABEMD-ANN model was compared with several benchmark models: the persistent model (discussed in Section 3.1.1), AdaBoost with regression tree (AdaBoost-RT), ANN, AdaBoost-ANN, EMD-ANN and ABEMD-RT. The nRMSE of the benchmark models and the proposed model are tabulated in Table 5.1.

The performance of the proposed ABEMD-ANN model and the benchmark models were compared by Friedman ranked sum test as shown in Table 5.2a. The p values of the test are much less than 0.05, and therefore it is 95% confident to conclude that the models for comparison are significantly different. For the post-hoc Nemenyi test, it is shown that the proposed ABEMD-ANN model has significantly better performance than the persistence model and AdaBoost-ANN for 1 hour ahead forecasting. It is also significantly better than ANN and EMD-ANN for 5 hour ahead forecasting. Therefore, we can conclude that the employment of AdaBoost to EMD-ANN has improved its performance.

Table 5.1 nRMSE of the Benchmark Models and the Proposed Model. Best record is emphasized in bold.

Dataset	Persistent	AdaBoost -RT	ANN	AdaBoost -ANN	EMD-ANN	ABEMD -RT	Proposed
Horizon=1 h							
1	0.2929	0.0666	0.0670	0.0844	0.0733	**0.0540**	0.0639
2	0.2277	0.0660	0.0662	0.1725	0.0609	0.0623	**0.0466**
3	0.2639	0.0619	0.0810	0.0946	0.3836	**0.0608**	0.0730
4	0.1710	0.0787	0.0731	0.1069	0.0873	0.0822	**0.0621**
5	0.1542	0.0728	0.0751	0.1083	0.0469	0.0454	**0.0397**
6	0.2016	0.0592	0.0589	0.0861	0.0506	**0.0489**	0.0518
7	0.1073	0.0454	0.0459	0.0603	0.0423	0.0322	**0.0316**
8	0.1246	0.0552	0.0488	0.0613	0.0782	0.0562	**0.0343**
9	0.1207	0.0528	0.0541	0.0566	0.2094	0.0484	**0.0423**
10	0.2528	0.0853	0.0628	0.1019	0.0978	0.0847	**0.0568**
11	0.1512	0.0688	0.0777	0.0811	**0.0503**	0.0879	0.0549
12	0.3004	0.0762	0.0817	0.1184	**0.0426**	0.0842	0.0432
Horizon=3 h							
1	0.2882	0.1034	0.1469	0.1236	0.1639	0.0900	**0.0894**
2	0.2264	0.0983	0.1189	0.1262	0.1762	**0.0752**	0.0806
3	0.2611	0.1558	0.1533	0.1927	0.8386	**0.1007**	0.1506
4	0.1794	0.1192	0.1350	0.1333	0.1701	0.1117	**0.1036**
5	0.1506	0.0944	0.1088	0.1229	0.0837	0.0747	**0.0686**
6	0.2018	0.1033	0.1000	0.1105	0.1000	**0.0743**	0.0749
7	0.1048	0.0736	0.0740	0.0900	0.1377	**0.0504**	0.0553
8	0.1221	0.0797	0.0827	0.0995	0.0820	0.0661	**0.0477**
9	0.1137	0.0760	0.0880	0.0773	**0.3042**	0.0785	0.0893
10	0.2617	0.1267	0.1669	0.1290	0.2638	0.1078	**0.0942**
11	0.1519	0.1051	0.1153	0.1086	0.1192	0.1048	**0.0808**
12	0.2990	0.1457	0.1306	0.1759	0.0974	0.1187	**0.0659**
Horizon=5 h							
1	0.2852	0.1262	0.1591	0.1523	0.1613	0.1123	**0.1115**
2	0.2235	0.1079	0.1461	0.1267	0.3060	**0.0933**	0.0946
3	0.2587	0.1728	0.2199	0.1851	0.5618	**0.1449**	0.1776
4	0.1816	0.1434	0.1952	0.1386	0.2435	0.1314	**0.1305**
5	0.1451	0.1264	0.1542	0.1411	0.1271	0.0936	**0.0810**
6	0.2032	0.1122	0.1331	0.1250	0.1100	0.0938	**0.0880**
7	0.1046	0.0843	0.0932	0.0874	0.1097	**0.0651**	0.0693
8	0.1202	0.1032	0.0816	0.1237	0.4012	0.0751	**0.0558**
9	0.1107	0.0903	0.1127	0.0983	0.4744	**0.1003**	0.1023
10	0.2654	0.1729	0.1855	0.1632	0.3918	**0.1193**	0.1260
11	0.1470	0.1284	0.1607	0.1366	0.2412	0.1219	**0.1075**
12	0.2974	0.1856	0.3358	0.1662	0.2351	0.1490	**0.0969**

The comparisons based on Friedman test and Nemenyi test have shown that the proposed ABEMD-ANN model outperformed the benchmark models significantly with exceptions on AdaBoost-RT and ABEMD-RT. Recall the conclusion in Chapter 4, the EMD did not improve the performance of ANN for short term wind forecasting but by introducing the AdaBoost into the forecasting model, the performance of EMD-ANN had been improved. Therefore, we can conclude that AdaBoost has a positive impact on EMD based ANN methods.

Table 5.2 Friedman Test and Nemenyi Test of the Proposed Model against the Benchmark Models, χ^2 is the statistics and p is the p-value.

(a)

Horizon	χ^2	p
1	47.357	1.588e-08
3	52.247	1.663e-09
5	55.321	3.992e-10

(b)

Horizon (h)	Persistent	AdaBoost-RT	ANN	AdaBoost-ANN	EMD-ANN	ABEMD-RT
1	5.5e-08	0.36529	0.17533	7.5e-05	0.11252	0.79266
3	3.0e-07	0.48706	0.03479	0.00856	0.00052	1.00000
5	4.7e-05	0.61549	0.00028	0.14133	7.0e-06	1.00000

5.3 Noise Assisted Ensemble of Empirical Mode Decomposition

Based on inspections in some cases, the IMFs decomposed by the EMD do not have a narrow band instantaneous frequency and this problem is known as mode mixing [149]. It is usually caused by a weak signal that is embedded to a strong signal near the extrema region and misleading the sifting process to pick the small fluctuations up and put them into an IMF instead of separating into two distinct IMFs [149]. Therefore, to determine whether the decomposition encounters this problem, the instantaneous frequency of the IMFs should be checked (usually via Hilbert spectra), and if an IMF consists of signal spanning a wide band of frequency or more than one IMFs contain signals in a similar frequency band, the decomposition is confused.

To solve this problem, some improved versions of EMD were proposed [149–151]. The strategy is to introduce data diversity which is discussed in Section 3.2 into the original TS to dilute the mode mixing sources, which is discussed in Sections 5.3.1, 5.3.2 and 5.3.3.

In the following sections, several noise assisted ensemble EMD are introduced and ensemble EMD-SVR methods are applied for wind speed and solar irradiance forecasting. For more details on ensemble EMD-ANN on time series forecasting, please refer to [19].

5.3.1 Ensemble Empirical Mode Decomposition (EEMD)

EEMD is a multiple trial process and each trial has the similar procedure as EMD except that the input TS is a mixture of the original TS and a finite Gaussian white noise. Although the resultant decompositions are more noisy, the uncorrelated finite white noise will cancel each other when calculating the mean of all trials, preserving the meaningful TS. In addition, EEMD also eliminates largely the mode mixing problem by utilizing the scale separation capability of EMD [149].

The steps of EEMD are:

1. Create a collection of noise added original TS: $x^i(t) = x(t) + \varepsilon^i(t),\ i \in \{1, \ldots, I\}$, where $\varepsilon(t)$ are independent Gaussian white noise.

2. For each $x^i(t)$, apply EMD (discussed in Section 4.1) to obtain the decomposed IMFs and residue: $x^i(t) = \sum_{j=1}^{N} c_j^i + r_N^i$.

3. In order to reconstruct back the original TS, one just needs to average on all trials:

$$x(t) = \frac{1}{I}(\sum_{i=1}^{I}\sum_{j=1}^{N} c_j^i + r_N^i) + \varepsilon_I \tag{5.1}$$

where $\varepsilon_I = \frac{\varepsilon}{\sqrt{I}}$ [149] is the aggregated error introduced by the white noise.

5.3.2 Complementary Ensemble Empirical Mode Decomposition (CEEMD)

Although EEMD greatly reduced the possibility of mode mixing, there raised a new problem: a non-negligible residue noise ε_I will be mixed into the original TS after reconstruction if the number of trails is not large enough (theoretical full completeness is achieved when the number of trials is infinity), thus the completeness is violated. In order to make the reconstructed TS complete and at the same time retain the advantage of EEMD, a CEEMD [150] was reported.

The basic structure of CEEMD is the same as EEMD but instead of fully independently generated Gaussian white noise, CEEMD generated a collection of independent Gaussian white noise and a complementary pair for each white noise to perfectly cancel each other:

$$\varepsilon^i(t) \in \{\varepsilon_+^{i/2}(t), \varepsilon_-^{i/2}(t)\} \tag{5.2}$$

where $\varepsilon_+^{i/2}(t) + \varepsilon_-^{i/2}(t) = 0$, $i \in \{1, \ldots, I\}$.

5.3.3 Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN)

Another problem with EEMD is the high computation cost. As the number of trial increases, the number of sifting process also increases. In order to reduce the number of trials while retaining the ability to solve the mode mixing problem, a CEEMDAN was reported in [151]. The steps of CEEMDAN are:

1. Create a collection of noise-added original TS: $x^i(t) = x(t) + w_0\varepsilon^i(t)$, $i \in \{1,\dots,I\}$, where $\varepsilon(t)$ are independent Gaussian white noise with unit variance and w_0 is a noise coefficient.

2. For each $x^i(t)$, apply EMD (discussed in Section 4.1) to obtain the first decomposed IMF and take average: $c_1(t) = \frac{1}{I}\sum_{i=1}^{I} c_1^i$. Then the first residue is $r_1(t) = x(t) - c_1(t)$.

3. Decompose the noise-added residue $r_1 + w_1E_1(\varepsilon^i(t))$ to obtain the second IMFs.

$$c_2(t) = \frac{1}{I}\sum_{i=1}^{I} E_1(r_1 + w_1E_1(\varepsilon^i(t))) \tag{5.3}$$

   where $E_j(\cdot)$ is a function to extract the jth IMF decomposed by EMD

4. Repeat for the remaining IMFs until there are no more than two extrema of the residue.

CEEMDAN is advantageous over EEMD and CEEMD because (i) it introduces an extra noise coefficient vector w to control the noise level at each and (iii) it requires less trials than EEMD and CEEMD. However, since CEEMDAN is a sequential process, it has an inherent hurdle to parallel computing.

5.4 Ensemble EMD-SVR Method on Wind Speed Forecasting

Twelve wind speed TS were chosen from NDBC [135] sites 41004, 44009 and 46077 on January, April, July and November, respectively. The forecasting horizon is 1–6 hour ahead. The TS were scaled to $[0,1]$ interval and split to 70% training and 30% testing. 5 fold CV for TS was used to choose the best parameters of the SVR. The parameter range of the SVR are: $C \in 10^{\{-3,\dots,4\}}$, $\varepsilon \in 10^{\{-4,\dots,-1\}}$ and $\gamma \in \{1,\dots,24\}/24$. The input feature set was selected by PACF selection as discussed in Chapter 4.

The number of iterations for EEMD and CEEMD is 100 and the number of iterations for CEEMDAN is 20. The standard deviation of the Gaussian noise for EEMD, CEEMD and CEEMDAN is 0.2.

nRMSE, symmetric mean absolute percentage error (sMAPE) and MASE were used to evaluate the forecasting error. The formulae for nRMSE and MASE can be found in equations (4.6) and (4.8), respectively. The formula for sMAPE is shown in (5.4).

$$\text{sMAPE} = \frac{1}{n}\sum_{t=1}^{n}\frac{|x-\hat{x}|}{(|x|+|\hat{x}|)/2}\times 100\% \tag{5.4}$$

where $\hat{x}$ is the predicted data, x is the desired data and n is the number of data points in the TS.

The error measures on 1, 6 and 12 hour ahead wind speed forecasting is shown in Table 5.3. From the table, we can see that the error increases with the increasing forecasting horizon.

Table 5.3 Error Measures of Different Ensemble EMD-SVR Methods for 1, 6 and 12 hour ahead Wind Speed Forecasting

	1 Hour ahead						6 Hour ahead						12 Hour ahead					
Dataset	Persistent	SVR	EMD-SVR	EEMD-SVR	CEEMD-SVR	CEEMDAN-SVR	Persistent	SVR	EMD-SVR	EEMD-SVR	CEEMD-SVR	CEEMDAN-SVR	Persistent	SVR	EMD-SVR	EEMD-SVR	CEEMD-SVR	CEEMDAN-SVR
	nRMSE																	
1	0.075	0.072	0.065	0.057	0.063	0.056	0.161	0.142	0.179	0.120	0.134	0.115	0.215	0.170	0.164	0.131	0.144	0.122
2	0.076	0.059	0.046	0.042	0.045	0.041	0.111	0.101	0.065	0.065	0.065	0.063	0.110	0.099	0.085	0.083	0.083	0.083
3	0.096	0.060	0.053	0.047	0.050	0.049	0.126	0.095	0.080	0.078	0.078	0.076	0.132	0.102	0.083	0.084	0.085	0.084
4	0.099	0.047	0.034	0.033	0.033	0.033	0.161	0.123	0.098	0.094	0.098	0.091	0.178	0.155	0.110	0.115	0.115	0.113
5	0.061	0.061	0.047	0.045	0.047	0.046	0.139	0.135	0.111	0.110	0.109	0.109	0.202	0.176	0.162	0.165	0.164	0.158
6	0.074	0.058	0.042	0.040	0.041	0.041	0.133	0.113	0.100	0.101	0.096	0.106	0.183	0.142	0.134	0.128	0.128	0.138
7	0.080	0.050	0.041	0.037	0.041	0.038	0.101	0.084	0.068	0.066	0.066	0.063	0.120	0.097	0.086	0.082	0.083	0.077
8	0.089	0.060	0.047	0.043	0.046	0.045	0.129	0.095	0.081	0.080	0.081	0.080	0.154	0.108	0.093	0.093	0.094	0.095
9	0.059	0.058	0.044	0.042	0.043	0.041	0.140	0.127	0.102	0.106	0.106	0.104	0.196	0.161	0.126	0.127	0.124	0.124
10	0.065	0.047	0.037	0.035	0.037	0.035	0.100	0.086	0.073	0.068	0.071	0.066	0.110	0.125	0.095	0.096	0.100	0.088
11	0.074	0.044	0.035	0.033	0.035	0.035	0.114	0.093	0.076	0.073	0.075	0.076	0.146	0.120	0.097	0.094	0.096	0.096
12	0.161	0.083	0.061	0.058	0.058	0.059	0.237	0.166	0.138	0.140	0.132	0.134	0.252	0.198	0.133	0.134	0.138	0.125
	sMAPE																	
1	13.641	13.344	14.441	12.613	13.976	12.987	37.471	32.803	34.973	26.925	28.638	26.684	47.560	38.474	35.467	30.935	32.687	29.186
2	26.135	18.829	15.297	14.190	15.072	13.917	39.707	34.858	23.288	23.442	23.712	23.477	45.998	37.757	30.324	29.018	29.138	29.089
3	23.109	14.680	12.677	11.460	12.239	12.046	32.215	25.078	20.669	20.253	20.407	20.359	37.066	27.495	23.020	23.558	23.593	22.509
4	24.804	11.810	8.877	8.520	8.461	8.640	39.298	29.638	24.579	24.113	24.773	22.859	44.889	38.976	30.556	30.384	30.980	30.306
5	24.162	23.238	17.279	16.963	17.079	17.161	49.413	43.932	37.083	35.599	36.331	35.764	61.844	49.296	47.382	46.737	48.132	46.950
6	21.456	17.413	13.218	12.697	13.236	12.954	37.212	32.461	29.958	30.826	28.808	32.078	49.973	37.588	37.395	34.836	36.068	37.119
7	41.608	27.684	24.211	21.847	23.997	22.101	53.421	42.187	35.447	34.299	34.620	33.643	58.641	46.885	41.664	40.353	40.111	37.218
8	23.096	13.537	10.929	10.037	10.730	10.173	31.662	23.797	20.245	19.910	20.067	19.489	37.519	28.316	23.794	23.323	23.794	24.015
9	15.857	14.642	12.007	11.490	11.757	11.495	32.912	27.348	23.779	24.460	24.167	23.905	45.374	35.437	28.243	28.697	28.363	28.707
10	25.653	19.271	15.394	13.926	15.293	14.119	36.871	30.357	26.305	24.502	25.662	24.851	39.556	44.634	34.445	34.125	35.493	32.023
11	29.207	17.882	14.092	12.985	13.466	14.121	43.584	34.933	29.660	28.316	28.592	29.641	51.612	42.859	34.992	33.635	34.674	35.792
12	42.854	23.695	17.869	17.338	17.610	17.161	60.129	39.780	35.446	35.911	33.519	34.319	62.940	47.678	33.335	32.880	34.077	31.536
	MASE																	
1	1.004	0.992	1.032	0.884	0.997	0.901	2.917	2.605	2.789	2.187	2.360	2.163	4.036	3.231	2.892	2.544	2.681	2.405
2	1.347	0.985	0.788	0.717	0.783	0.706	2.077	1.896	1.161	1.168	1.168	1.167	2.266	2.007	1.595	1.531	1.524	1.555
3	1.609	0.948	0.867	0.784	0.827	0.824	2.240	1.717	1.408	1.373	1.382	1.377	2.519	1.873	1.539	1.581	1.590	1.524
4	2.049	0.974	0.720	0.697	0.698	0.685	3.446	2.551	2.107	2.046	2.138	1.952	4.042	3.471	2.516	2.558	2.614	2.552
5	1.002	0.994	0.752	0.739	0.746	0.766	2.469	2.419	1.961	1.908	1.939	1.896	3.446	2.990	2.873	2.826	2.882	2.813
6	1.286	1.021	0.756	0.721	0.756	0.750	2.551	2.257	1.993	2.037	1.916	2.111	3.369	2.649	2.541	2.328	2.425	2.466
7	1.558	1.010	0.838	0.744	0.832	0.760	2.126	1.801	1.431	1.385	1.389	1.353	2.521	2.108	1.800	1.713	1.707	1.576
8	1.701	1.002	0.795	0.736	0.782	0.751	2.496	1.812	1.535	1.502	1.535	1.489	2.955	2.168	1.825	1.790	1.830	1.858
9	1.002	0.995	0.772	0.721	0.743	0.718	2.238	2.046	1.623	1.689	1.682	1.669	3.136	2.602	1.952	2.002	1.968	2.006
10	1.301	0.967	0.761	0.688	0.751	0.694	2.102	1.822	1.575	1.430	1.528	1.456	2.366	2.822	2.108	2.111	2.226	1.925
11	1.689	1.012	0.798	0.737	0.776	0.773	2.617	2.164	1.746	1.663	1.684	1.723	3.357	2.869	2.215	2.108	2.182	2.188
12	1.941	0.986	0.702	0.689	0.695	0.692	3.030	2.047	1.735	1.759	1.629	1.674	3.281	2.603	1.616	1.627	1.680	1.529

The persistent model is the simplest forecasting model which is discussed in Section 3.1.1. The persistent model assume the predicted value is the same as the latest observed value. From the table, we can see that all machine learning based methods outperformed the persistent model. A Wilcoxon signed rank test confirmed this conclusion as tabulated in Table 5.4.

Table 5.4 p value of the Wilcoxon Signed Rank Test between the Persistent Model and the SVR based Methods for Wind Speed Forecasting, $p < 0.05$ means the performance is significantly different.

Horizon	nRMSE					sMAPE					MASE				
(h)	SVR	EMD-SVR	EEMD-SVR	CEEMD-SVR	CEEMDAN-SVR	SVR	EMD-SVR	EEMD-SVR	CEEMD-SVR	CEEMDAN-SVR	SVR	EMD-SVR	EEMD-SVR	CEEMD-SVR	CEEMDAN-SVR
1	0	0.065	0	0.003	0	0	0.003	0	0	0	0	0.026	0	0.001	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0.001	0	0	0.001	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0.076	0	0	0	0	0.013	0	0	0	0	0.235	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The performance difference among the ensemble EMD-SVR methods was tested by a Friedman rank sum test (Table 5.5) first. From the result, we can see that there is significant performance difference among the SVR, EMD-SVR and the ensemble EMD-SVR methods since the p values are much smaller than 0.05. In order to distinguish their performances, a pairwise Wilcoxon signed rank test (Figure 5.2) was implemented. From the plot, we can see that the EMD-SVR method had significant advantage in accuracy compared against the SVR method, which confirms the conclusion drawn in Chapter 4. The ensemble EMD-SVR methods also outperformed the SVR method as expected. The three different ensemble versions of EMD (EEMD, CEEMD, CEEMDAN) improved the performance of EMD-SVR which can be seen from the plot. However, there was no significant difference between EEMD-SVR and CEEMD-SVR. We can infer that the reconstruction completeness from the original TS is not important to affect the wind speed forecasting accuracy. The CEEMDAN-SVR outperformed EEMD-SVR on 2 hour ahead forecasting based on nRMSE and MASE and it outperformed CEEMD-SVR for several forecasting horizons (1, 2, 3, 6, 7 and 12 hour ahead).

The CPU time of the decompositions is plotted in Figure 5.3. As shown, EEMD and CEEMD has similar range of CPU time but CEEMDAN has much smaller CPU time than EEMD and CEEMD. However, due to the different ensemble procedures, the CPU time ratio of CEEMDAN v.s. EEMD/CEEMD (in this paper CEEMDAN:EEMD=1:2.16) is larger than the number of trials ratio (in this paper CEEMDAN:EEMD=20:100=1:5). There-

Table 5.5 Friedman Rank Sum Test among the SVR method and the EMD based SVR Methods. p is the p value and χ^2 is the statistics.

Horizon	nRMSE		sMAPE		MASE	
(h)	p	χ^2	p	χ^2	p	χ^2
1	1.3e-08	42.4	2.9e-08	40.8	4.1e-08	40.0
2	1.9e-07	36.8	2.0e-07	36.7	3.8e-06	30.5
3	2.7e-06	31.2	5.2e-06	29.8	7.8e-06	29.0
4	1.2e-05	27.9	4.7e-06	30.0	3.4e-06	30.7
5	1.9e-06	31.9	1.3e-07	37.5	8.4e-07	33.7
6	5.7e-08	39.4	3.6e-07	35.5	2.4e-07	36.3
7	1.0e-08	43.0	1.2e-08	42.6	1.2e-08	42.6
8	1.1e-06	33.1	1.3e-07	37.6	1.6e-07	37.2
9	2.2e-06	31.6	1.0e-05	28.3	3.6e-06	30.6
10	2.0e-08	41.5	2.9e-08	40.8	1.3e-07	37.6
11	1.2e-07	37.8	1.8e-07	37.0	1.2e-06	32.8
12	3.3e-06	30.8	8.7e-07	33.6	1.1e-06	33.0

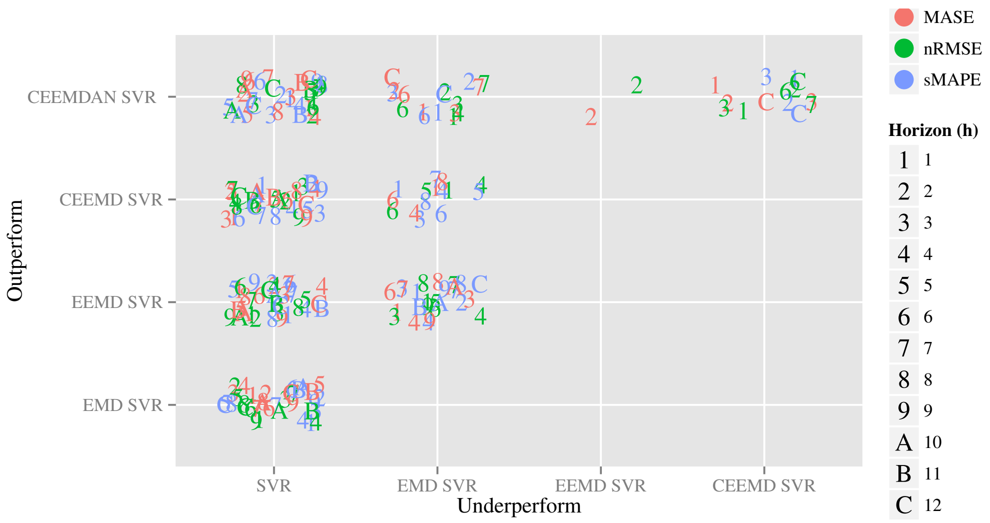


Figure 5.2 Jitter Plot of the Wilcoxon Signed Rank Test among the SVR based Methods. A colored digit show the out-performance of a certain horizon based on a certain error metric, e.g. CEEMDAN-SVR outperformed EEMD-SVR on 2 hour ahead forecasting based on MASE and in the plot a red digit '2' is marked.

fore, CEEMDAN has a much more time consuming ensemble procedure than EEMD and CEEMD.

Based on both accuracy and computation time, CEEMDAN-SVR is the best forecasting method among the EMD based SVR methods for wind speed forecasting.

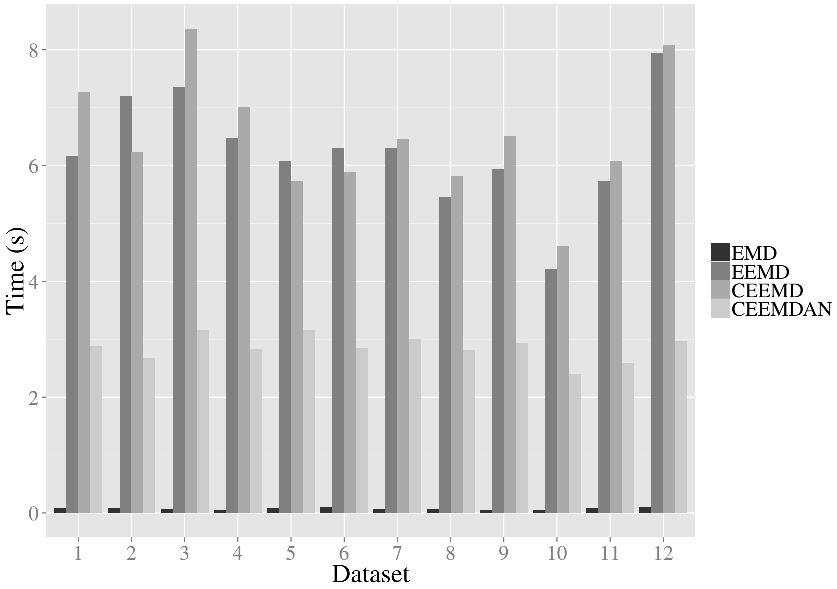


Figure 5.3 CPU Time (s) of EMD, EEMD, CEEMD and CEEMDAN on Wind Speed TS, number of trials for EEMD and CEEMD is 100 and number of trials for CEEMDAN is 20.

5.5 Ensemble EMD-SVR Methods on Solar Irradiance Forecasting

As discussed in Section 2.5, the solar irradiance TS is periodical (daily cycle and annual cycle). The cloud cover and haze effect act as an interruption TS which is superimposed into the ideal solar irradiance TS. This interruption may cause the EMD process suffer from mode mixing problem.

Twelve solar irradiance TS were chosen from NSRDB database [45] sites 690160, 701195 and 702757 on January–February, April–May, July–August, October–November, respectively. The forecasting horizon are 24 and 48 hour (1 and 2 day) ahead. The TS was scaled to $[0,1]$ interval and split to 70% training and 30% testing. 5 fold CV for TS was used to choose the best parameters of the SVR. The parameter range of the SVR are: $C \in 10^{\{-3,\ldots,4\}}$, $\varepsilon \in 10^{\{-4,\ldots,-1\}}$ and $\gamma \in \{1,\ldots,96\}/96$. The input feature set was selected by PACF selection as used in Chapter 4. The number of iterations for EEMD and CEEMD is 100 and the number of iterations for CEEMDAN is 20. The standard deviation of the Gaussian noise for EEMD, CEEMD and CEEMDAN is 0.2.

For solar irradiance forecasting, the error measure is slightly different because the sunrise and sun-down time for each day are deterministic and therefore the evaluation excluded the time during sun-down and sun-rise (intervals without sunlight). Due to the discrete nature of the predicted data, MASE is not suitable. In addition, since the actual irradiance values can be very close to zero, sMAPE will not reflect a proper measure. In this section, nRMSE is used to evaluate the performance. The nRMSE on 24 and 48 hour ahead solar irradiance forecasting is shown in Table 5.6. From the table, we can see that the error increases with the increasing forecasting horizon.

Table 5.6 nRMSE of Different Ensemble EMD-SVR Methods for 24 and 48 hour ahead Solar Irradiance Forecasting

	24 Hour ahead						48 Hour ahead					
Dataset	Persistent	SVR	EMD -SVR	EEMD -SVR	CEEMD -SVR	CEEMDAN -SVR	Persistent	SVR	EMD -SVR	EEMD -SVR	CEEMD -SVR	CEEMDAN -SVR
1	0.166	0.174	0.235	0.139	0.129	0.136	0.180	0.179	0.212	0.143	0.143	0.154
2	0.134	0.110	0.269	0.098	0.099	0.108	0.141	0.115	0.292	0.105	0.109	0.113
3	0.109	0.077	0.163	0.085	0.078	0.088	0.107	0.081	0.127	0.083	0.081	0.087
4	0.110	0.115	0.123	0.094	0.096	0.098	0.128	0.114	0.145	0.104	0.104	0.109
5	0.050	0.039	0.100	0.039	0.056	0.040	0.051	0.045	0.107	0.040	0.053	0.040
6	0.149	0.120	0.148	0.105	0.116	0.106	0.144	0.123	0.139	0.112	0.124	0.113
7	0.131	0.108	0.134	0.084	0.097	0.094	0.140	0.107	0.125	0.091	0.097	0.102
8	0.021	0.014	0.019	0.011	0.013	0.013	0.024	0.0134	0.018	0.012	0.013	0.013
9	0.069	0.063	0.087	0.048	0.051	0.047	0.064	0.099	0.091	0.047	0.048	0.047
10	0.128	0.110	0.148	0.079	0.100	0.096	0.144	0.113	0.151	0.091	0.108	0.101
11	0.132	0.102	0.156	0.092	0.105	0.092	0.126	0.107	0.151	0.098	0.139	0.098
12	0.016	0.013	0.016	0.014	0.013	0.017	0.017	0.013	0.014	0.013	0.014	0.017

The persistent model for solar irradiance forecasting is slightly different from that for wind speed forecasting. For solar irradiance forecasting, the predicted value is assumed to be the same as the most recently observed value at the same time of the day. For example, if the solar irradiance that is going to predict is at 1200PM on Wednesday and the forecasting horizon is 48 hours, then the predicted value is the observed value at 1200PM on the Monday.

Due to the highly periodicity, the persistent model had relatively accurate forecasting results. As shown by the Wilcoxon signed rank test (Table 5.7), the persistent model outperformed the EMD-SVR method in both 24 and 48 hour ahead forecasting. However, the other machine learning methods such as SVR and ensemble EMD-SVR methods still had significant better performance than the persistent model.

Table 5.7 p value of the Wilcoxon Signed Rank Test between the Persistent Model and the SVR based Methods for Solar Irradiance Forecasting, $p < 0.05$ means the performance is significantly different.

Horizon(h)	SVR	EMD-SVR	EEMD-SVR	CEEMD-SVR	CEEMDAN-SVR
24	0.006	0.999	0.000	0.001	0.001
48	0.017	0.992	0.000	0.006	0.001

To compare the performance among the SVR based methods, Friedman rank sum test was first executed and the test statistics showed a significant performance difference among the methods: $p = 4.3 \times 10^{-5}$, $\chi^2 = 25.3$ for 24 hour ahead forecasting, and $p = 2.1 \times 10^{-4}$, $\chi^2 = 21.9$ for 48 hour ahead forecasting. A post-hoc pairwise Wilcoxon signed rank test was performed and the test statistics are tabulated in Table 5.8.

It is shown that EMD-SVR was the worst method among the four EMD based methods and it underperformed the SVR method. The possible reason is that due to the characteristics of the solar irradiance TS, the EMD process suffer from mode mixing problem and the decomposed IMFs were not able to simplify the input space for the SVR but on the other hand degraded the overall performance of SVR. The performance of ensemble EMD-SVR methods (EEMD-SVR, CEEMD-SVR and CEEMDAN-SVR) outperformed the EMD-SVR is because that the ensemble versions attenuated the mode mixing problem.

Comparing among the three ensemble EMD-SVR methods, it is shown that EEMD-SVR is the best. the EEMD-SVR outperformed CEEMD-SVR and CEEMDAN-SVR for both 24 and 48 hour ahead forecasting. This means that maintaining the completeness had no evidence to improve the forecasting accuracy, and the dominant factor is the number of trials. An additional experiment compared the EEMD-SVR and the CEEMDAN-SVR with 100 trials on the same solar irradiance datasets and the Wilcoxon signed rank test did not show evidence on the significant difference between them ($p = 0.99$ for 24 hour ahead forecasting and $p = 0.44$ fore 48 hour ahead forecasting).

Table 5.8 p value of Wilcoxon Signed Rank Test of Ensemble EMD-SVR on Solar Irradiance Forecasting, $p < 0.05$ means the performance is significantly different.

			Underperformance				
			SVR	EMD-SVR	EEMD-SVR	CEEMD-SVR	CEEMDAN-SVR
Outperformance	Horizon=24 h	SVR	–	0.000	0.993	0.935	0.978
		EMD-SVR	1	–	1	1	0.999
		EEMD-SVR	0.008	0.000	–	0.0319	0.017
		CEEMD-SVR	0.075	0.000	0.973	–	0.688
		CEEMDAN-SVR	0.026	0.001	0.986	0.338	–
	Horizon=48 h	SVR	–	0.001	0.997	0.830	0.986
		EMD-SVR	0.999	–	0.999	0.999	0.999
		EEMD-SVR	0.003	0.001	–	0.008	0.000
		CEEMD-SVR	0.190	0.001	0.993	–	0.604
		CEEMDAN-SVR	0.017	0.001	0.999	0.425	–

5.6 Concluding Remarks

This chapter has introduced several ensemble versions of EMD. The first part has applied AdaBoost ensemble framework to EMD-ANN method (ABEMD-ANN) to improve the performance. Compared with the EMD-ANN which has not improved the performance from the ANN method on wind speed forecasting, the ABEMD-ANN has better performance (answer to RQ-5.1). The second part has employed noise-assistance method to create data diversity for EMD. EEMD, CEEMD and CEEMDAN have been discussed in the section. The three ensemble versions of EMD have been combined with SVR for wind speed and solar irradiance forecasting. For wind speed forecasting, the ensemble EMD-SVR methods has outperformed the EMD-SVR method (answer to RQ-5.2) and among the three ensemble versions, CEEMDAN-SVR has the best performance because it has smaller forecasting error and shorter computation time (answer to RQ-5.3). The EEMD-SVR and CEEMD-SVR methods have similar performance and therefore, completeness is not a key factor to wind speed TS forecasting.

However, due to the nature of solar irradiance TS, the fluctuation due to haze effect and cloud cover is deeply embedded in the theoretical solar irradiance TS, making the non-ensemble EMD suffering significantly from the mode mixing problem. Therefore, EMD-SVR has not outperform the SVR method for solar irradiance forecasting (answer to RQ-5.2). The noise-assisted ensemble EMD versions have attenuated the mode mixing problem and the performance has been elevated significantly. Among the three ensemble EMD-SVR methods, EEMD-SVR has the best performance (answer to RQ-5.3), which again emphasized that completeness is not a key factor for solar irradiance forecasting. For TS with worse mode-mixing problem, the number of noise-assisted trials become more important and this is the reason why EEMD-SVR outperformed CEEMD-SVR and CEEMDAN-SVR in the experiment.

Chapter 6

Neural Network with Random Weights and Direct Input-Output Connections

In the previous chapters, SVR and ANN are used for wind speed and solar irradiance forecasting. SVR is time consuming because it requires parameter tuning and it is usually undergoes an exhaustive grid search. The training procedure of ANN is also time consuming because it requires iterative BP to determine the weights of every connection between the neurons.

In this chapter, a family of fast training ANN is introduced. The structure of the ANN is single hidden layer feed-forward, which is a common ANN structure. However, the training procedure is different from the conventional single hidden layer feed-forward neural network (SLFN) that the connections weights between input and hidden layer are set randomly [152], the output neurons are linear and the connection weights between hidden and output layers are estimated using least square estimation [153–155].

In the following sections, several different types of ANN with random weights are discussed and the different configurations are evaluated on several wind speed and solar irradiance TS.

Three research questions (RQs) are raised and will be answered in this chapter:

RQ-6.1 How do the input layer bias, hidden layer bias and direct input output connections affect the performance of RVFL network for TS forecasting?

RQ-6.2 What is the advantage of the RVFL network comparing against the linear–nonlinear hybrid forecasting method?

RQ-6.3 Does the RVFL network outperform some benchmark methods for TS forecasting?

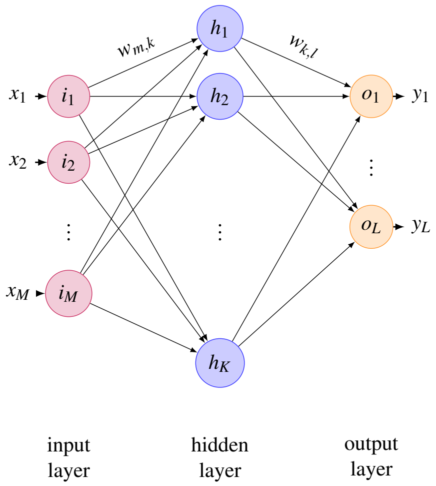


Figure 6.1 Schematic Diagram of an SLFN

6.1 Single Hidden Layer Feed-forward Neural Network

The structure of a feed-forward neural network is discussed in Section 3.1.5. Among the feed-forward neural networks, the most popularly used is single hidden layer. The conventional SLFN has three layers: input layer, hidden layer and output layer as shown in Figure 6.1. The neurons between adjacent layers (input to hidden layer and hidden to output layer) are connected. There is no interconnection of neurons within the same layer or across non-adjacent layers (input to output layer). The hidden neuron has a non-linear activation and the commonly used activation functions are (logistic sigmoid) *logsig* function and (hyperbolic sigmoid) *tanh* function (Section 3.1.5). The value of the output neuron is a weighted summation of the hidden neurons' activation values.

To determine the output value, the input to hidden layer weights and the hidden to output layer weights should be determined. The most widely used method to determine the optimal weights is BP [156]. BP has two phases: forward and backward propagation and weight updating. In the forward and back ward propagation phase, error of each weight is calculated based on the actual input and output and the current network's parameter. In the weight updating phase, the weights are updated based on the error gradient and a learning rate. BP is finished until the error between the ANN's output and the actual output is within a predefined threshold. However, iterative BP is time consuming and the results is likely to be trapped in the local minima.

6.2 SLFN with Random Weights

In 1992, Schmidt *et al.* [152] reported an SLFN with random weights (RWSLFN) assigned to the input to hidden layer connections. The reported RWSLFN has the same structure as a conventional SLFN shown in Figure 6.1. The activation function of the hidden neurons is a *logsig* function.

Training of an SLFN is to minimize the squared output error by finding the optimal hidden layer weights and output layer weights.

$$\min \varepsilon^2 = \sum_{j=1}^{N}(y_j - \hat{y}_j)^2 = \sum_{j=1}^{N}(y_j - \sum_{k=1}^{K} w_{k,l} f(\sum_{m=1}^{M} w_{m,k} i_m^{(j)} + b_{i,k}) + b_{h,j})^2 \tag{6.1}$$

where $f(\cdot)$ is a *logsig* activation function, y_j is a desired output and $\hat{y}_j$ is a predicted output and $\underset{1\times M}{\mathbf{i}^{(j)}}$ is the corresponding input. N is the number of samples, M is the number of input neurons and K is the number of hidden neurons. $w_{m,k}$ are the weights between the input and hidden layer neurons, $w_{k,l}$ are the weights between the hidden and output layer neurons and $b_{i,k}$ and $b_{h,j}$ are the input layer and hidden layer biases, respectively.

The difference between the reported RWSLFN and the conventional SLFN is that for the conventional SLFN, the optimal hidden layer and output layer weights are determined by BP whereas in the reported RWSLFN, the hidden layer weights $\mathbf{W}_h$ are randomly assigned from a uniform distribution within $[-1,1]$ and the output layer weights $\mathbf{W}_o$ are optimized by a least square method.

6.3 Random Vector Functional Link Neural Network

In 1994, Pao *et al.* [153] proposed a random vector functional link (RVFL) neural network that combines the advantage of random weights and functional link [157] together. The RVFL network is an extension of SLFN with additional direct connections from the input layer to the output layer as shown in Figure 6.2 (the dashed arrows show the direct connections between the input neurons and the output neurons).

The RVFL network [153] has a set of nodes called enhancement nodes, which are equivalent to the neurons in the hidden layer in the conventional SLFN. Similar to [152], the weights from the input layer to the enhancement nodes are randomly assigned. In [153], the authors proposed to use a conjugated BP to tune the weights from the input layer to the output layer as well as the weights from the enhancement nodes to the output layer instead of a least square method. However, the authors also stated that applying the least square method is possible if the matrix inversion is feasible [153].

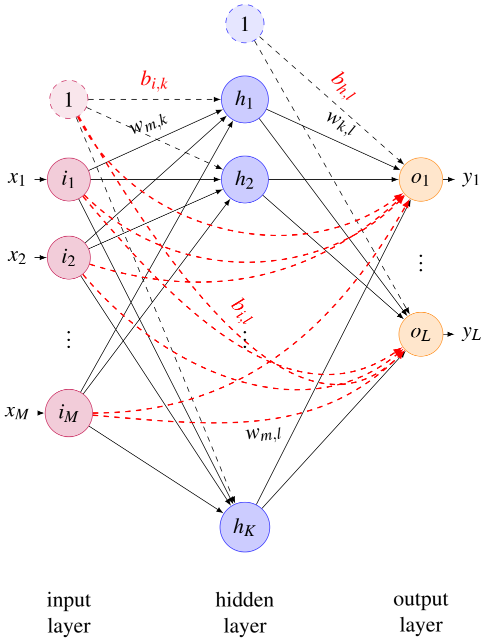


Figure 6.2 Schematic Diagram of an RVFL network, the dashed red arrows show the direct connections between the input neurons and the output neurons, the dashed black arrows show the connections from a bias neuron, $f(\cdot)$ is a *logsig* activation function. N is the number of samples, M is the number of input neurons and K is the number of hidden neurons. $w_{m,k}$ are the weights between the input and hidden layer neurons, $w_{k,l}$ are the weights between the hidden and output layer neurons and $b_{i,k}$ and $b_{h,j}$ are the input layer and hidden layer biases, respectively.

For the range of the random weights, the authors gave a brief rule of thumb: the value of the weights $\mathbf{W_K}$ should constrain the activation function $f(\cdot)$ away from its saturation region most of the time [153].

In 1996, Chen [154] reported an RVFL network for function approximation and interpolation. The structure of the reported RVFL network is the same as [153] with direct connections from the input to the output layers. However, the activation function of the enhancement node in Chen's RVFL network is *tanh* whereas it is *logsig* in Pao's RVFL network. In addition, Chen's RVFL network has an additional activation function applied to the output layer when applied to solve function approximation problems [154] while for TS forecasting, this activation function in the output layer is not-used [155].

Another novelty of Chen's RVFL network is that it has a recursive fast training capability. When adding a new training sample to the existing trained RVFL network model, the output weights $\mathbf{W}_o$ need not be re-trained but can be updated based only on the previously calculated data and the new training sample. Similarly, when a new enhancement node is added, total retraining is avoided by updating the model with only the existing data and the random weights associated with the newly added enhancement node. The details of the fast training procedures can be found in [154, 155]. However, a requisite of this recursive training is that when a new training sample or a new enhancement node is added to the existing RVFL network, the rank of the augmented extended input pattern matrix $[\mathbf{X}|f(\mathbf{W}^T\mathbf{X}+b)]$ should be greater than the extended input pattern matrix before augmentation. These reported RWSLFNs/RVFL network are summarized in Table 6.1 as M2 and M3.

6.4 Variations of RWSLFN and RVFL Network

In the literature, we found that there are four possibilities to modify the SLFN/RVFL network: input layer bias, hidden layer bias, input-output connections and activation function. As the two different sigmoidal activation functions has almost identical shapes but different output interval, different scaling factors are required. In terms of performance, they are equivalent to each other. Therefore, we will use *logsig* as the default activation function for all possible variations. The eight RWSLFN/RVFL network configurations are shown in Table 6.1 [1]. In Sections 6.5 and 6.6, we will compare the performance of these variations on wind speed and solar irradiance TS data.

6.5 Discussions on RWSLFN for Wind Speed Forecasting

The eight different configurations of RWSLFN are used for wind speed TS forecasting. The wind speed TS was retrieved from NDBC [135]. Four datasets were extracted from a recorded TS during 2011 from site 41004. Each dataset has 4 weeks record and the first 3 weeks' data is for training and the remaining 1 week's data is for testing. The forecasting horizon is 1, 4, 8 and 12 hours ahead. The datasets were scaled to $[0,1]$.

The number of enhancement nodes (hidden neurons) were determined by a 5 fold CV in order to avoid overfitting. The evaluation was repeated for 10 times (trials) to average out the variations from the random generator.

The nRMSE metric was used to compare the performances of different configurations, which is tabulated in Table 6.2. It is shown that for all configurations, the nRMSE increases with the increasing forecasting horizon. The performance differences of the different configurations are compared in the following subsections.

[1] Debates on a similar network configuration called Extreme Learning Machine (ELM) can be found in http://elmorigin.weebly.com/ or http://elmorigin.wix.com/originofelm/

Table 6.1 RWSLFN/RVFL Network with Different Configurations, $f(\cdot)$ is a *logsig* activation function. N is the number of samples, M is the number of input neurons and K is the number of hidden neurons. $w_{m,k}$ are the weights between the input and hidden layer neurons, $w_{k,l}$ are the weights between the hidden and output layer neurons and $b_{i,k/l}$ and $b_{h,j}$ are the input layer and hidden layer biases, respectively.

Method	Input Layer Bias	Hidden Layer Bias	Input-Output Connection	Formula
M1	√	√	√	$h_k = f(\sum_{m=1}^{M} w_{m,k} i_m + b_{i,k})$ $o_l = \sum_{k=1}^{K} w_{k,l} h_k + b_{h,l} + \sum_{m=1}^{M} w_{m,l} i_m + b_{i,l}$
M2 [152]	√	√	×	$h_k = f(\sum_{m=1}^{M} w_{m,k} i_m + b_{i,k})$ $o_l = \sum_{k=1}^{K} w_{k,l} h_k + b_{h,l}$
M3 [153]	√	×	√	$h_k = f(\sum_{m=1}^{M} w_{m,k} i_m + b_{i,k})$ $o_l = \sum_{k=1}^{K} w_{k,l} h_k + \sum_{m=1}^{M} w_{m,l} i_m + b_{i,l}$
M4	√	×	×	$h_k = f(\sum_{m=1}^{M} w_{m,k} i_m + b_{i,k})$ $o_l = \sum_{k=1}^{K} w_{k,l} h_k$
M5	×	√	√	$h_k = f(\sum_{m=1}^{M} w_{m,k} i_m)$ $o_l = \sum_{k=1}^{K} w_{k,l} h_k + b_{h,l} + \sum_{m=1}^{M} w_{m,l} i_m$
M6	×	√	×	$h_k = f(\sum_{m=1}^{M} w_{m,k} i_m)$ $o_l = \sum_{k=1}^{K} w_{k,l} h_k + b_{h,l}$
M7	×	×	√	$h_k = f(\sum_{m=1}^{M} w_{m,k} i_m)$ $o_l = \sum_{k=1}^{K} w_{k,l} h_k + \sum_{m=1}^{M} w_{m,l} i_m$
M8	×	×	×	$h_k = f(\sum_{m=1}^{M} w_{m,k} i_m)$ $o_l = \sum_{k=1}^{K} w_{k,l} h_k, \forall k \in \{1,\ldots,K\}, \forall l \in \{1,\ldots,L\}$

Table 6.2 nRMSE of Different RWSLFN Configurations on Wind Speed Forecasting, the values are averaged over 10 trials, the means and the standard deviations are recorded.

	Dataset	M1	M2	M3	M4	M5	M6	M7	M8
1 Hour	D1	0.056±0.002	0.06±0.002	0.056±0.002	0.06±0.003	0.056±0.001	0.063±0.009	0.056±0.001	0.06±0.002
	D2	0.051±0.001	0.054±0.002	0.051±0.001	0.054±0.003	0.051±0.001	0.054±0.003	0.051±0	0.054±0.002
	D3	0.043±0.001	0.046±0.002	0.043±0.001	0.045±0.001	0.043±0.001	0.044±0.001	0.043±0.001	0.045±0.003
	D4	0.058±0.002	0.065±0.003	0.058±0.002	0.065±0.004	0.056±0.002	0.067±0.007	0.056±0.001	0.065±0.004
4 Hour	D1	0.106±0.005	0.108±0.006	0.106±0.005	0.109±0.004	0.104±0.003	0.107±0.005	0.105±0.005	0.106±0.003
	D2	0.1±0.004	0.099±0.002	0.1±0.004	0.098±0.002	0.099±0.002	0.097±0.002	0.099±0.002	0.098±0.003
	D3	0.074±0.002	0.076±0.002	0.074±0.002	0.078±0.003	0.075±0.001	0.083±0.014	0.075±0.002	0.078±0.003
	D4	0.106±0.006	0.109±0.002	0.106±0.006	0.11±0.003	0.105±0.003	0.11±0.003	0.105±0.003	0.11±0.003
8 Hour	D1	0.137±0.005	0.136±0.004	0.137±0.005	0.142±0.009	0.136±0.005	0.14±0.004	0.133±0.002	0.144±0.011
	D2	0.114±0.004	0.112±0.001	0.114±0.004	0.112±0.001	0.114±0.004	0.112±0.002	0.114±0.003	0.111±0.002
	D3	0.093±0.002	0.095±0.002	0.093±0.002	0.095±0.002	0.095±0.003	0.096±0.005	0.094±0.003	0.098±0.005
	D4	0.148±0.004	0.149±0.003	0.148±0.004	0.149±0.004	0.146±0.003	0.151±0.001	0.147±0.003	0.154±0.005
12 Hour	D1	0.147±0.007	0.144±0.003	0.147±0.007	0.147±0.006	0.144±0.003	0.145±0.002	0.143±0.003	0.148±0.007
	D2	0.111±0.002	0.111±0.001	0.111±0.002	0.111±0.001	0.113±0.003	0.112±0.002	0.112±0.002	0.113±0.002
	D3	0.11±0.007	0.109±0.004	0.11±0.007	0.111±0.011	0.113±0.004	0.106±0.003	0.111±0.004	0.108±0.007
	D4	0.17±0.005	0.17±0.003	0.17±0.005	0.168±0.003	0.166±0.003	0.172±0.003	0.165±0.003	0.172±0.003

6.5.1 Influence of Input Layer Bias and Hidden Layer Bias

In order to see the influence of input layer bias, paired Wilcoxon signed rank tests were used to compare four different pairs: M1 v.s. M5, M2 v.s. M6, M3 v.s. M7 and M4 v.s. M8. The

p values are tabulated in Table 6.3. For the longer term forecasting horizons (4, 8 and 12 hours ahead), the RWSLFN with input layer bias has significantly better performance than the RWSLFN without input layer bias.

Table 6.3 Wilcoxon Signed Rank Test Statistics (based on nRMSE) of the RWSLFNs with Input Layer Bias v.s. without Input Layer Bias on Wind Speed Forecasting, alternative hypothesis H_a is 'less than', $p < 0.05$ means significant outperformance.

Comparison	Horizon (h)			
	1	4	8	12
M1 v.s. M5	0.081	**0.013**	**0.011**	0.115
M2 v.s. M6	0.101	0.229	**0.008**	**0.001**
M3 v.s. M7	0.408	**0.007**	**0.002**	0.057
M4 v.s. M8	0.377	0.121	**0.012**	**0.001**

Similarly, paired Wilcoxon signed rank tests were used to compare four different pairs: M1 v.s. M3, M2 v.s. M4, M5 v.s. M7 and M6 v.s. M8, for assessing the influence of hidden layer bias. The p values are tabulated in Table 6.4. We can conclude that the hidden layer bias does not influence on the performance of the wind speed forecasting because almost all p values are larger than 0.05.

Table 6.4 Wilcoxon Signed Rank Test Statistics (based on nRMSE) of the RWSLFNs with Hidden Layer Bias v.s. without Hidden Layer Bias on Wind Speed Forecasting, alternative hypothesis H_a is 'less than', $p < 0.05$ means significant outperformance.

Comparison	Horizon (h)			
	1	4	8	12
M1 v.s. M3	0.121	0.281	0.313	0.637
M2 v.s. M4	0.144	0.964	0.242	0.187
M5 v.s. M7	0.387	0.387	**0.006**	0.096
M6 v.s. M8	0.906	0.281	0.523	0.160

6.5.2 Influence of Direct Input Output Connections

The influence of direct input output connections were assessed by comparing the performances of M1 v.s M2, M3 v.s. M4, M5 v.s. M6 and M7 v.s. M8. It is shown that the direct input output connections improved the performance of RWSLFN for 4, 8 and 12 hour ahead forecasting. However, for 1 hour ahead forecasting, the significance is not obvious.

Based on the comparisons, we saw that the input layer bias and direct input output connections significantly improved the performance of RWSLFN for 4–12 hour ahead forecasting whereas hidden layer bias did not affect the performance. Therefore, we can conclude

Table 6.5 Wilcoxon Signed Rank Test Statistics (based on nRMSE) of the RWSLFNs with Direct Input Output Connections v.s. without Direct Input Output Connections on Wind Speed Forecasting, alternative hypothesis H_a is 'less than', $p < 0.05$ means significant outperformance.

Comparison	Horizon (h)			
	1	4	8	12
M1 v.s. M2	0.290	0.002	7.0e-07	3.5e-05
M3 v.s. M4	0.550	1.8e-04	5.2e-07	1.6e-05
M5 v.s. M6	0.560	0.002	8.4e-07	1.3e-05
M7 v.s. M8	0.632	0.002	3.7e-06	2.5e-05

that the RVFL network (RWSLFN with input layer bias and direct input output connections, M3 in Table 6.1) is the preferred configuration for short term wind TS forecasting. In the following comparison, the RVFL network will be used.

6.5.3 RVFL Network v.s. Hierarchical Methods

It is worthwhile to evaluate the optimization strategy of the RVFL network. In the RVFL network, the outputs from the enhancement nodes (non-linear output) and the outputs directly from the input neurons (linear output) are optimized together by a least square estimation.

To examine the effectiveness of the optimization strategy of the RVFL network, a two-stage hierarchical method is constructed and compared with it. The two-stage method is to firstly train an ARIMA (linear part) and then train an time delayed ANN or an SVR for the residue (non-linear part). The details of the hierarchical methods can be found in Section 3.4.2.

The Wilcoxon signed rank test statistics are shown in Table 6.6. It is shown that the RVFL network outperformed the ARIMA-ANN for all forecasting horizon but it underperformed ARIMA-SVR on 1 and 4 hour ahead forecasting (because $p > 0.95$). For 8 and 12 hour ahead forecasting, the RVFL network had comparable performance as ARIMA-SVR.

Table 6.6 Wilcoxon Signed Rank Test (based on nRMSE) of RVFL Network v.s. ARIMA-ANN and ARIMA-SVR on Wind Speed Forecasting, alternative hypothesis H_a is 'less than', $p < 0.05$ means significant outperformance.

	Horizon (h)	ARIMA-ANN	ARIMA-SVR
RVFL	1	9.09e-13	1.00
	4	9.09e-13	0.99
	8	9.09e-13	0.89
	12	1.18e-13	0.64

6.5.4 RVFL Network v.s. other Methods

The RVFL network was compared with other machine learning methods: ANN, SVR and random forests (RF). These methods are popularly applied to TS forecasting as well as classification. The nRMSE of these methods on wind speed forecasting are shown in Table 6.7. It is shown that ANN has the worst performance among the four methods but the comparison of the RVFL network v.s. SVR and RF are not obvious. Table 6.8 shows the p value of the Wilcoxon signed rank test between RVFL network and the other three methods. It is shown that the RVFL network outperformed ANN for all four horizons and the RVFL network had comparable performance as SVR and RF.

Table 6.7 nRMSE of ANN, SVR, RF and the RVFL Network on Wind Speed Forecasting

Dataset	Horizon=1 h				Horizon=4 h				Horizon=8 h				Horizon=12 h			
	ANN	SVR	RF	RVFL	ANN	SVR	RF	RVFL	ANN	SVR	RF	RVFL	ANN	SVR	RF	RVFL
D1	0.136	0.064	**0.056**	**0.056**	0.142	0.132	0.115	**0.106**	0.179	0.162	0.178	**0.137**	0.197	0.172	0.206	**0.147**
D2	0.150	0.060	0.061	**0.051**	0.167	0.107	0.114	**0.100**	0.167	0.153	0.174	**0.114**	0.194	0.175	0.204	**0.111**
D3	0.137	0.069	0.056	**0.043**	0.170	0.105	0.092	**0.074**	0.197	0.135	0.135	**0.093**	0.218	0.140	0.145	**0.11**
D4	0.085	0.053	**0.045**	0.058	0.123	0.082	0.067	**0.106**	0.160	**0.090**	0.095	0.148	0.205	0.109	**0.104**	0.170

Table 6.8 Wilcoxon Signed Rank Test Statistics (based on nRMSE) of the RVFL Network v.s. other Machine Learning Methods on Wind Speed Forecasting, alternative hypothesis H_a is 'less than', $p < 0.05$ means significant outperformance.

	Horizon (h)	ANN	SVR	RF
RVFL	1	0.006	0.125	0.500
	4	0.006	0.187	0.437
	8	0.006	0.437	0.312
	12	0.006	0.312	0.312

6.6 Discussions on RWSLFN for Solar Irradiance Forecasting

The eight different configurations of RWSLFN were also tested on four solar irradiance TS datasets. The solar irradiance TS was retrieved from NSRDB [45] site 690160. The forecasting horizon is 24 and 48 hour (1 and 2 day) ahead. The TS was scaled to $[0,1]$ interval and split to 70% training and 30% testing. 5 fold CV for TS was used to choose the optimal number of enhancement node of the networks. Each experiment run repeatedly for 10 times and the average and standard deviation of the nRMSE are recorded in Table 6.9.

Notice that the nRMSE is calculated after the values between sun-down and sun-rise period removed.

Table 6.9 nRMSE of Different RWSLFN Configurations on Solar Irradiance Forecasting

	D1		D2		D3		D4	
Method	24 h	48 h	24 h	48 h	24 h	48 h	24 h	48 h
M1	0.198±0.015	0.215±0.018	0.16±0.01	0.159±0.008	0.098±0.004	0.097±0.002	0.088±0.001	0.092±0.001
M2	0.251±0.017	0.258±0.017	0.192±0.011	0.178±0.008	0.11±0.004	0.105±0.004	0.09±0.003	0.094±0.003
M3	0.197±0.016	0.213±0.013	0.161±0.01	0.161±0.008	0.098±0.004	0.097±0.002	0.087±0.001	0.092±0.001
M4	0.251±0.017	0.259±0.015	0.186±0.015	0.181±0.011	0.11±0.005	0.106±0.003	0.09±0.003	0.094±0.003
M5	0.219±0.026	0.203±0.009	0.165±0.005	0.164±0.007	0.098±0.004	0.096±0.003	0.088±0.002	0.094±0.002
M6	0.273±0.024	0.259±0.02	0.187±0.013	0.188±0.012	0.112±0.005	0.105±0.004	0.09±0.004	0.091±0.002
M7	0.218±0.026	0.208±0.021	0.164±0.004	0.162±0.007	0.098±0.003	0.096±0.002	0.088±0.002	0.093±0.002
M8	0.273±0.024	0.259±0.02	0.187±0.013	0.188±0.012	0.112±0.005	0.105±0.004	0.09±0.004	0.091±0.002

To evaluate the influences of input layer bias, hidden layer bias and direct input output connections, paired Wilcoxon signed rank test were implemented and the test statistics are shown in Table 6.10. No evidence showed that the input or hidden layer biases significantly differed the performance whereas the direct input output connections improved the performance of RWSLFN.

Table 6.10 Wilcoxon Signed Rank Test (based on nRMSE) of the RWSLFNs on Solar Irradiance Forecasting (a) with Input Layer Bias v.s. without Input Layer Bias, (b) Hidden Layer Bias v.s. without Hidden Layer Bias, (c) with Direct Input Output Connections v.s. without Direct Input Output Connections, alternative hypothesis H_a is 'less than', $p < 0.05$ means significant outperformance.

(a)

Comparison	Horizon (h)	
	24	48
M1 v.s. M5	0.694	0.952
M2 v.s. M6	0.910	0.608
M3 v.s. M7	0.435	0.333
M4 v.s. M8	0.805	0.201

(b)

Comparison	Horizon (h)	
	24	48
M1 v.s. M3	0.026	0.777
M2 v.s. M4	0.847	0.744
M5 v.s. M7	0.675	0.157
M6 v.s. M8	0.142	0.221

(c)

Comparison	Horizon (h)	
	24	48
M1 v.s. M2	0.029	0.004
M3 v.s. M4	0.010	0.005
M5 v.s. M6	0.029	0.017
M7 v.s. M8	0.004	0.006

The RVFL network (M3 configuration from the eight configurations of RWSLFN) was compared with other methods: ARIMA, SVR, RF and the hybrid ARIMA-SVR methods. The parameters of ARIMA were optimized by a grid search with AIC as the evaluation criterion, the parameters of the SVR were optimized by a grid search with $C \in 10^{\{-2,\dots,4\}}$, $\gamma \in \{1,\dots,48\}/48$ and $\varepsilon \in 10^{\{-4,\dots,0\}}$. The number of trees in RF is 500 and the number of variables randomly sample at each node is 16. The nRMSE of these methods are shown in Table 6.11. It is shown that the RVFL network had comparable performance as the other four methods and the paired Wilcoxon signed rank test shown in Table 6.12) confirmed the statement.

Table 6.11 nRMSE of RVFL Network v.s. Other Machine Learning Methods on Solar Irradiance Forecasting

Method	D1		D2		D3		D4	
	24 h	48 h	24 h	48 h	24 h	48 h	24 h	48 h
RVFL	0.197	0.213	0.161	0.161	0.098	0.097	0.087	0.092
ARIMA	0.355	0.199	0.140	0.106	0.356	0.234	0.185	0.110
SVR	0.233	0.164	0.091	0.089	0.240	0.169	0.090	0.089
RF	0.168	0.158	0.094	0.081	0.174	0.154	0.093	0.082
ARIMA-SVR	0.233	0.165	0.095	0.087	0.240	0.174	0.093	0.091

Table 6.12 Wilcoxon Signed Rank Test Statistics (based on nRMSE) of the RVFL Network v.s. Other Machine Learning Methods on Solar Irradiance Forecasting, alternative hypothesis H_a is ‘less than’, $p < 0.05$ means significant outperformance.

	24 h	48 h
RVFLv.s.ARIMA	0.125	0.437
RVFLv.s.SVR	0.312	0.769
RVFLv.s.RF	0.562	0.812
RVFLv.s.ARIMA-SVR	0.312	0.687

6.7 Concluding Remarks

This chapter has investigated single hidden layer feed-forward neural network with random weights (RWSLFN) reported in [152] and random vector functional link (RVFL) neural network reported in [153, 155]. The structural differences and similarities have been discussed. Six possible variations of RWSLFN/RVFL network has been constructed based on three possible combinations: input layer bias, hidden layer bias and direct input output connections.

The eight configurations have been evaluated on a wind speed time series which has been further partitioned into four different datasets. The best configuration of RWSLFN/RVFL network has been compared with three popular forecasting methods: ANN, RF and SVR and the results have been reported. Based on the experiments, we can conclude that:

- The hidden layer bias does not have significant impact on the performance of RWSLFN/RVFL network on neither wind speed nor solar irradiance forecasting (answer to RQ-6.1).

- The input layer bias has improved the performance of RWSLFN for wind speed forecasting but not for solar irradiance forecasting. The direct input output connections has improved the performance of RWSLFM for both wind speed and solar irradiance forecasting. The RVFL network (RWSLFN with direction input output connections) is the best configuration among the eight configurations for wind speed forecasting (answer to RQ-6.1).
- For wind speed forecasting, the RVFL network has been compared with hierarchical forecasting method with sequential linear and non-linear forecasting. It is shown that the one pass least square estimation based RVFL network has better performance than the two pass ARIMA-ANN method and the RVFL network has comparable performance as ARIMA-SVR method. Therefore, direct input output connections embedded in the ANN have better performance than a linear predictor (such as ARIMA) followed by an ANN predictor (answer to RQ-6.2).
- For wind speed forecasting, the RVFL network has outperformed the ANN method for 1-12 hour ahead forecasting and has comparable performance as SVR and RF (answer to RQ-6.3).

For solar irradiance forecasting, the RVFL network has comparable performance as ARIMA, SVR, RF and ARIMA-SVR. The advantage of the RVFL network in terms of accuracy is not observed.

Chapter 7

Random Vector Functional Link Neural Network for Wind Power Ramp Forecasting

The importance and challenges in wind energy was discussed in Chapters 1 and 2. From the previous discussion, we understand the necessity of wind forecasting. In this chapter, wind power is investigated because it is the final product of wind farms which is injected into the electricity grid. Due to the intermittent nature of the wind, the power generated by wind farms is usually fluctuating. The fluctuations in the wind power is usually compensated by the conventional fossil power generator or battery storage systems. However, if the fluctuation is large and rapid, the compensation will not be effective.

Significant fluctuations are called ramps [158, 159]. The ramps can be classified into up ramp where the power generated significantly increases in a short period and down ramp where the power generated significantly decreases in a short period [160]. Wind power ramp is harmful to the power system planning and scheduling [161, 162] and can cause breakdown to the power transmission and generation system [163]. However, there is no industrial standard to clearly quantify the ramp [164] because it depends on wind farms, power grids and industrial/national standards, etc. Therefore in this chapter, we will provide some power ramp definitions that are commonly used in the literature.

There are different approaches to forecast the wind power ramps in the literature. Some papers focused on identifying power ramps by different definitions [165–167]. Some papers focused on the methodology to forecast or classify the power ramps. ANN was applied to forecast the wind power ramp in [161]. The authors used ANN based stochastic process model to simulate and forecast the probabilities of a future ramp. In [164], SVM, RF and ANN were employed to forecast the future power ramp rate. SVM was also used to forecast the wind power ramp in [159]. In [160], physical model based wind power forecasting methods were used to forecast the power ramp in the Electric Reliability Council of Texas (ERCOT) wind farms. Another physical approach to forecast wind power ramp was reported in [168].

This chapter applies the RVFL network [155] to detect whether or not there will be power ramps in the next 6 or 12 hours based on a classification approach. The details of the RVFL network can be found in Section 6.3.

Three research questions (RQs) are raised and will be answered in this chapter:

RQ-7.1 How to define a wind power ramp and what are the characteristics of wind power ramp?

RQ-7.2 How to deal with imbalanced data and abnormal data?

RQ-7.3 Does the RVFL network outperform some benchmark methods for wind power ramp forecasting?

RQ-7.4 Does the RVFL network have advantage in computation time for wind power ramp forecasting?

7.1 Wind Power Ramp

A wind power ramp is the change of power, either increasing or decreasing in a specific time. We call the increasing power ramp up ramp and the decreasing power ramp down ramp. Wind power ramp is common in wind power generation because the wind speed is fluctuating. In addition, wind power is not a linear function with respect to the wind speed because it has cut in region, cut out region, cubic region and maximum-output region [5] which is discussed in Section 2.4.

In the literature, there are numerous definitions of power ramps, in this chapter we introduce two most popular definitions and employ them in our experiments. The first definition is based on the local extrema within a certain time interval. A significant power ramp occurs when the ramp Γ_{ext} exceeds a certain threshold [166]:

$$\Gamma_{ext} = max(P(t,\ldots,t+\Delta t)) - min(P(t,\ldots,t+\Delta t)) > \Gamma_{val} \tag{7.1}$$

where $P(t)$ is the wind power generated at time t, Δt is the time interval, and Γ_{val} is the threshold to determine whether the ramp is significant or not.

Another definition is based on the two end points of a sliding window [166] and the wind power is denoted as Γ_{end}.

$$\Gamma_{end} = |P(t+\Delta t) - P(t)| > \Gamma_{val} \tag{7.2}$$

Notice that the above-mentioned two definitions do not take into account the ramp directions (up-ramp or down-ramp) and hence the decisions are binary (ramp or no ramp). The multiple decisions (up-ramp, down-ramp or no-ramp) is not discussed in this chapter.

Based on these two definitions, the wind ramp forecasting can be converted to a binary classification problem.

Classification is to predict a decision based on a set of features, and the decision can be either binary class or multiple class. For wind power ramp forecasting based on the definition, the classification is binary. However, wind ramp classification is different from the normal classification tasks because the number of occurrence of wind ramp (minority class) is usually much smaller than that of no ramp (majority class). This kind of data is known as imbalanced data.

Imbalanced data can mislead the training process if we choose inappropriate error measures. For example, if the error measure is accuracy *ACC*, it is sometimes even higher when all predicted decisions are under the majority class and no prediction is under the minority class, which is obviously a wrong prediction.

In order to overcome this problem, the majority class in the training data is under sampled to match the number of minority class, or the minority class in the training data is over sampled to match the number of majority class [169]. In addition, the criteria to measure the performance of the classifier are appropriately selected for imbalanced data as well.

7.2 Wind Power Time Series Pre-processing

The wind power TS is affected by meteorological factors and power system factors and thus it is highly fluctuating. Noise are created when (i) there is a strong wind gust in the environment, (ii) the wind turbine is shut down for maintenance, or (iii) there are sensor fault or noise in the data acquisition and recording system. In order to avoid identifying unnecessary wind ramps from the TS due to noise, the noise embedded in the wind power TS should be removed. There are three types of noise: (i) outliers, (ii) missing values and (iii) white noise corresponding to the above-mentioned possible noise causes. We propose to detect them separately and smooth the outliers and missing values and discard the white noise.

7.2.1 Outlier

To detect the outlier, there is a rolling window with a fixed width to roll along the TS. For each window, there is a segment of TS $\mathbf{X}_w = \{x_i, x_{i+1}, \ldots, x_{i+w}\}$, where w is the window width. For each $\mathbf{X}_w$, the median absolute deviation (MAD) is calculated (equation (7.3)) and the data points $x_{MAD} \leq T_{th} \times \text{MAD}$ is considered as an outlier [170].

$$\text{MAD} = \text{median}_i\left(\left| X_i - \text{median}_j(X_j) \right| \right) \tag{7.3}$$

7.2.2 Missing Data

The missing data is recorded as 'NA', 'NaN', '?', '-' or '-999' in the dataset depend on the data recording standard. These missing data can be easily identified manually.

7.2.3 White Noise

The white noise is usually embedded into the TS. The characteristic of white noise is that it has a constant power spectral density. Depending on the probability distribution, white noise can be Gaussian distributed. or uniformly distributed.

EMD based method can be applied to detect the white noise [171, 172]. As discussed in Chapter 4, a TS can be decomposed into IMFs and a residue. The characteristics of the IMFs are: (i) zero mean, (ii) the number of zero-crossings and the extrema differ at most by one and (iii) the IMFs are (nearly) orthogonal to each other [120].

Viewing from the frequency domain, the energy density of a single IMF is:

$$E_n = \frac{1}{N}\sum_{t=1}^{N} c_n(t)^2 \tag{7.4}$$

where N is the length of the TS, E_n is the nth IMF's energy density and $c_n(t)$ is the nth IMF.

The average period of an IMF $\bar{T}_n$ is calculated based on the Fourier spectrum weighted mean period [172].

After obtaining E_n and $\bar{T}_n$, the particular IMF can be checked because if it is a white noise, the relationship between the two variables are [171]:

$$\ln E_n + \ln \bar{T}_n = 0 \tag{7.5}$$

The derivation of the confidence intervals can be found in [171, 172] and the equation of the confidence interval is:

$$\ln E_n = -\ln \bar{T}_n \pm k\sqrt{\frac{2}{N}\exp\frac{ln\bar{T}_n}{2}} \tag{7.6}$$

where N is the length of the TS and k is the percentiles of a standard normal distribution,

However, if the average period $\bar{T}_n$ is calculated by counting the number of zero-crossings (or extrema), there is an empirical correction [171, 172] which is:

$$\ln E_n + 0.934 \ln \bar{T}_n = 0.12 \tag{7.7}$$

The IMFs whose energy and average period obey the equation is considered as a white noise and the particular IMFs are removed from re-construction back to the TS.

7.3 Proposed Random Vector Functional Link Neural Network based Wind Power Ramp Forecasting Methods

We propose to use RVFL network [155] to classify the wind power ramp. There are two approaches and they are plotted in Figure 7.1. The wind power TS first undergoes outlier removal, missing data removal and de-noising to be a smoothed TS $x(t)$. Firstly, the power ramp Γt is derived from the smoothed TS and the class labels are assigned to the it: '+1' for significant ramp and '-1' for no ramp. Then the power ramp series is partitioned into a training set and a testing set. In addition, the minority class in the training set is oversampled to Γ_t^b so that the amount of the two classes equal [169]. Notice that the testing set is remain the same, i.e. without any over or under sampling. Next the RVFL network is trained with the training set and an optimal number of hidden layers determined by a k-fold CV to obtain a classification model m_c. Finally the trained model is applied to classify the testing data and the predicted power ramp $\hat{\Gamma}_s$ is obtained.

7.4 Results and Discussions

The wind power TS data was retrieved from ELIA wind power website [173]. The wind power TS retrieved covers the period from November 2014 to March 2015. The rated power of the wind farm is 712.9 MWh. The time series was partitioned into five monthly series named as $D1$ to $D5$, and each monthly series is re-scaled from 15 min average to hourly average by a 4 to 1 sub-sampling and scaled to $[0, 1]$ interval. The TS was smoothed by the

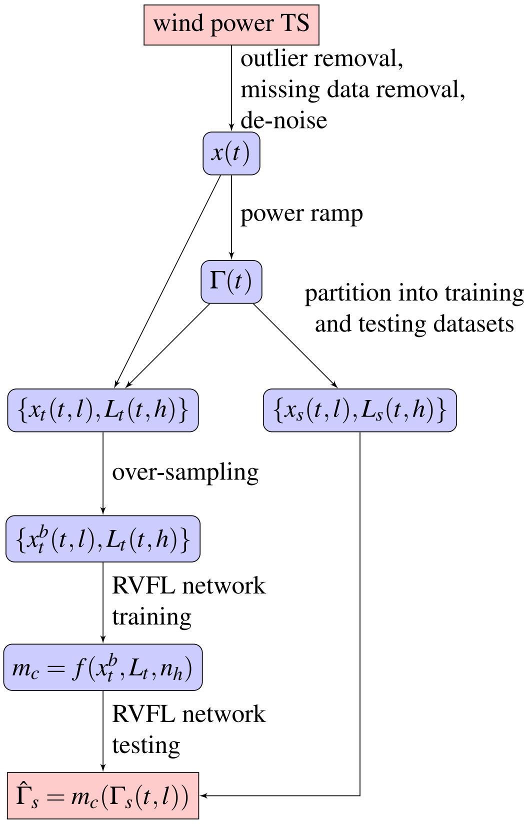


Figure 7.1 Flowchart of the RVFL Network on Wind Power Ramp Forecasting, where the subscript t and s are annotation for training and testing respectively $m_c(\cdot)$ is the trained classification model and $f(\cdot)$ is the RVFL training function.

MAD based outlier removal, missing data removal and moving average smoothing procedure. The TS was then decomposed by an EMD method and the energy and the average period of each IMF was calculated. If the two variables of the IMF followed equation (7.7), the IMF is discarded and the remaining IMFs and the residue is reconstructed back to the de-noised TS. The monthly time series were partitioned into a first 70% training and remaining 30% testing structure.

The $D1$ data is used to illustrate the EMD de-noising procedure. The energy v.s. average period is plotted in Figure 7.2. We can see from that all the decomposed 7 IMFs neither fall onto the empirical white noise line (dashed straight line) and nor within the 1st-99th percentile confidence interval. Therefore, there is no IMFs considered as noise in this particular case.

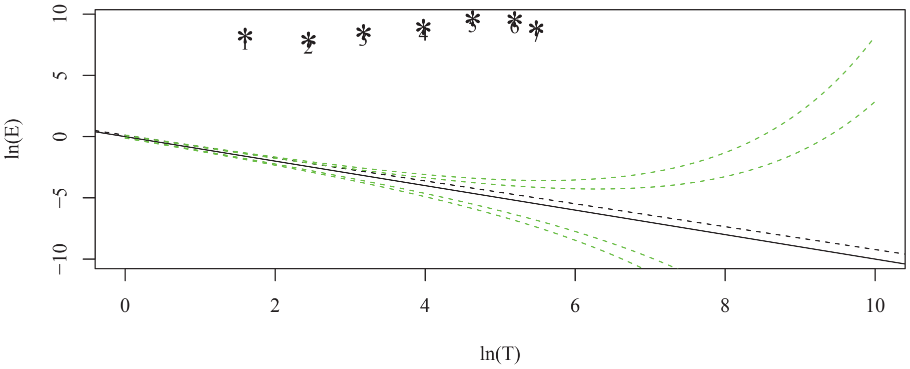


Figure 7.2 The Energy v.s. Average Period Plot for D1 Data. The solid straight line is the theoretical line to determine white noise and the dashed straight line is the empirical line with average period calculated from counting zero-crossings and the dashed curved lines are 1st, 5th, 95th and 99th percentiles.

Based on the wind ramp definition and the literature [165], we set $\Gamma_{val} = 25\%$ of the rated power (712.9 MWh) and $\Delta t = 4h$, we can find the start and end time points of the wind ramps based on the definition of Γ_{ext}. A fraction of wind power time series is plotted in Figure 7.3 and the identified significant ramps are plotted in red segments by joining the start and end time points.

The classification task is to predict whether there exists a significant wind power ramp in the next 6 or 12 hours based on the previous 48 hours historical data. If there exists a wind power ramp, the class is '+1' (has ramp), otherwise the class is '-1' (no ramp). The training and target wind ramp decisions are pre-determined by the two wind ramp definitions: Γ_{ext} and Γ_{end}. The summary of the datasets after partition, pre-processing and labeling is shown in Table 7.1. It is shown that for the next 6 hour ahead forecasting, the datasets are imbal-

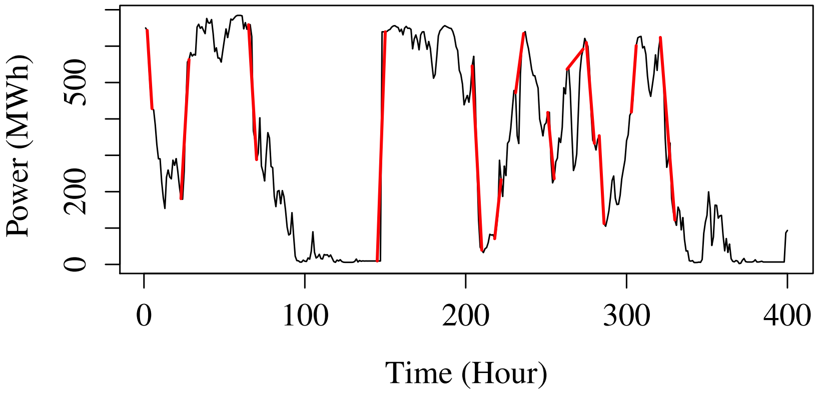


Figure 7.3 Plot of Wind Power Time Series of D1 data, red segments denote the significant power ramps.

anced whereas for the next 12 hour ahead forecasting, the dataset is almost balanced. It is rational because longer time window means higher possibility to have wind power ramps falling inside. Therefore, over-sampling of the minority class in the training set is not necessary for 12 hour ahead forecasting.

Table 7.1 Summary of the Datasets for Wind Ramp Classification

	6 Hour				12 Hour			
	Training		Testing		Training		Testing	
	+1	-1	+1	-1	+1	-1	+1	-1
Γ_{ext}								
D1	64	387	32	184	153	292	71	145
D2	332	136	55	168	252	210	111	112
D3	103	365	54	169	220	242	120	103
D4	77	340	58	144	152	259	75	127
D5	78	389	58	165	164	297	107	116
Γ_{end}								
D1	62	389	26	190	124	321	54	162
D2	130	228	45	178	237	225	85	138
D3	93	375	60	163	167	295	115	108
D4	82	335	78	124	150	261	62	140
D5	84	383	59	164	144	317	117	106

7.4.1 Performance Measures

The binary class classification model is evaluated based on a contingency table (or confusion matrix). The contingency table is a 2 × 2 matrix that reflects the relationship between the target and the predicted values. Derived from the contingency table are the several performance metrics as shown below.

Table 7.2 Contingency Table to Evaluate the Performance of the Binary Class Classification, where x is the obtained class, $\hat{x}$ is the predicted class, TP: true positive, FN: false negative, FP: false positive, TN: true negative.

	$\hat{x} = +1$	$\hat{x} = -1$
$x = +1$	TP	FN
$x = -1$	FP	TN

$$\text{Acc} = \frac{TP+TN}{TP+FN+FP+TN} \tag{7.8}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{7.9}$$

$$\text{Specificity} = \frac{TN}{FP+TN} \tag{7.10}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{7.11}$$

$$F\ \text{Score} = 2\frac{\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}} \tag{7.12}$$

where TP stands for true positive, FN stands for false negative, FP stands for false positive and TN stands for true negative.

However, for imbalanced data, accuracy (*ACC*) is not a good metric to measure the performance and the preferred metrics are Precision, Recall and F score [160, 167]. Therefore, they are used to evaluate the wind power ramp classification methods.

The above-mentioned error metrics are called threshold based metric because the value is associated with a threshold (parameters of a classifier). Another kind of error metric is based on the error distribution. Area under receiver operating characteristic curve (AUC) [174, 175] and H-measure [176] are of such kind. However, In following experiment, the performances of classifiers are compared after parameter optimization, only threshold based error metric is applied.

7.4.2 Parameter Tuning of the Classification Methods

The RVFL network is evaluated with three other benchmark methods: ANN (single layer), SVM (RBF kernel) and RF. The parameters of the RVFL network, ANN and SVM are optimized by a 5-fold CV module and the RF has a build-in out-of-bag error module to select the optimal parameters. Instead of using accuracy as the parameter selection criterion, F score is applied.

The optimal parameters of ANN, SVM, and the RVFL network after 5-fold CV and the parameters of RF are tabulated in Table 7.3.

As the raw outputs of ANN, RF, SVM and RVFL network are continuous values (probabilities/confidence), they are converted to categorical values (classification) by rounding for ANN and RVFL network; majority voting for RF; and since SVM is binary classification, the class labels are determined by the sign function $sgn()$.

Table 7.3 Optimal Parameters of the Classification Methods, 'mtry': number of variables randomly sampled as candidates at each split, 'ntree': number of trees to grow, n_h: number of hidden neurons, C: penalty factor of SVM, σ: RBF kernel's shape parameter.

		6 Hour					12 Hour				
		D1	D2	D3	D4	D5	D1	D2	D3	D4	D5
Γ_{ext}											
RF		mtry=16, ntree=500									
ANN	$n_h =$	96	42	96	54	54	48	54	96	54	30
SVR	$C =$	100	10000	100	100	1000	100	100	1000	100	100
	$\gamma =$	0.604	0.188	0.188	0.937	0.687	0.521	0.875	0.354	0.604	0.187
RVFL	$n_h =$	96	96	96	96	96	96	96	96	96	96
Γ_{end}											
RF		mtry=16, ntree=500									
ANN	$n_h =$	72	72	60	36	54	48	96	60	36	30
SVR	$C =$	100	1000	100	1000	100	100	100	100	100	100
	$\gamma =$	0.771	0.521	0.521	0.687	0.604	0.437	0.104	0.187	0.187	0.187
RVFL	$n_h =$	96	96	96	96	96	96	96	72	72	96

7.4.3 Classification Performance

The performance measures are tabulated in Table 7.4. The performance on 12 hour ahead forecasting is generally higher than that of the 6 hour ahead forecasting which is due to the class balancing in the testing set. The performance measures of the RVFL network is generally higher for both wind power ramp definitions on 12 hour ahead forecasting.

For 6 hour ahead forecasting. There are ‘NaN’ cases in F Score results due to the complete mis-classification of TP (also results a ‘0’ in Precision and Recall). The performance of the RVFL network is comparable with the other three methods.

A Friedman rank sum test in Table 7.5 showed the statistical comparisons among the four classification methods. It is shown that for 6 hour ahead forecasting, the four methods are comparable whereas for 12 hour ahead forecasting, there are significant differences among them. Next a post-hoc Nemenyi test is applied to the 12 hour ahead forecasting and the p value is shown in Table 7.6. From the test statistics, we can see that the RVFL network has comparable performances as ANN and RF and can outperform SVM for wind ramp defined by Γ_{ext}.

7.4.4 Computation Time

The average and standard deviation of the computation time of training and testing over 5 datasets are shown in Table 7.7. ANN has the longest training time among them all due to the iteration of BP. Because RF is regression tree based, the training and testing are fast. SVM requires grid search during training thus it has slow training speed but fast testing speed. The RVFL network has the advantage of fast training because it utilizes least square method to calculate the output layer weights as well as the fast testing because of the matrix calculation. From the table, we can see that in general the RVFL network has the best testing time yet comparable training time compared with RF.

7.5 Concluding Remarks

This chapter has covered an important topic on wind power utilization: wind power ramp forecasting. Two definitions of wind power ramp have been introduced (answer to RQ-7.1). Since the wind power ramp data is imbalanced, over sampling of the minority class is use to balance the training set; F score is used to measure the performance of the classifiers (answer to RQ-7.2). The outliers are detected by median absolute deviation and the white noise is detected by EMD (answer to RQ-7.2). The random vector functional link (RVFL) neural network has been employed to forecast the wind power ramp. The RVFL network has comparable performance as the benchmark methods: artificial neural network (ANN), random forests (RF) and support vector machine (SVM) for 6 hour ahead forecasting but the RVFL network has better performance than the SVM for 12 hour ahead forecasting (answer to RQ-7.3). The computation time of the RVFL network has significant advantage over

Table 7.4 Performance Measures of the Power Ramp Classification in the next 6 and 12 hours, ‘NaN’: not a number due to divide by zero.

		6 Hour				12 Hour			
	Dataset	ANN	RF	SVM	RVFL	ANN	RF	SVM	RVFL
	Γ_{ext}								
F Score	D1	0.031	NaN	0.07	**0.081**	0.17	0.162	0.149	**0.215**
	D2	0.226	0.135	0.146	**0.304**	0.458	**0.508**	0.365	0.478
	D3	0.189	0.123	0.204	**0.33**	0.324	0.512	0.43	**0.581**
	D4	**0.4**	0.304	0.222	0.392	0.579	0.631	0.543	**0.68**
	D5	0.314	**0.465**	0.253	0.296	0.362	**0.593**	0.331	0.585
Precision	D1	0.03	0	**0.08**	0.071	0.171	**0.286**	0.18	0.26
	D2	0.217	0.176	0.122	**0.233**	0.408	**0.473**	0.353	0.434
	D3	0.192	0.185	0.227	**0.234**	0.393	0.492	0.454	**0.531**
	D4	**0.351**	0.412	0.206	0.316	0.63	0.658	0.6	**0.691**
	D5	0.268	**0.714**	0.324	0.32	0.42	**0.683**	0.452	0.612
Recall	D1	0.031	0	0.062	**0.094**	0.169	0.113	0.127	**0.183**
	D2	0.236	0.109	0.182	**0.436**	0.523	**0.55**	0.378	0.532
	D3	0.185	0.093	0.185	**0.556**	0.275	0.533	0.408	**0.642**
	D4	0.466	0.241	0.241	**0.517**	0.535	0.606	0.496	**0.669**
	D5	**0.379**	0.345	0.207	0.276	0.318	0.523	0.262	**0.561**
	Γ_{end}								
F Score	D1	NaN	NaN	NaN	**0.061**	**0.122**	0.022	0.018	0.023
	D2	0.23	0.061	0.143	**0.265**	0.303	0.382	0.353	**0.483**
	D3	0.204	0.277	0.197	**0.329**	0.296	**0.54**	0.336	0.4
	D4	**0.453**	0.067	0.159	0.398	0.594	0.609	0.598	**0.636**
	D5	0.355	**0.509**	0.306	0.273	0.49	**0.578**	0.35	0.574
Precision	D1	0	0	0	**0.05**	**0.115**	0.027	0.017	0.029
	D2	**0.238**	0.095	0.111	0.198	0.312	0.333	0.303	**0.405**
	D3	0.182	**0.382**	0.194	0.272	0.341	**0.594**	0.352	0.419
	D4	**0.525**	0.25	0.183	0.357	0.679	**0.778**	0.713	0.755
	D5	0.338	**0.574**	0.385	0.247	0.51	**0.746**	0.519	0.583
Recall	D1	0	0	0	**0.077**	**0.13**	0.019	0.019	0.019
	D2	0.222	0.044	0.2	**0.4**	0.294	0.447	0.424	**0.6**
	D3	0.233	0.217	0.2	**0.417**	0.261	**0.496**	0.322	0.383
	D4	0.397	0.038	0.141	**0.449**	0.529	0.5	0.514	**0.55**
	D5	0.373	**0.458**	0.254	0.305	0.472	0.472	0.264	**0.566**

SVM and ANN. It has shorter training time than RF and has comparable testing time as RF (answer to RQ-7.4).

Table 7.5 Friedman Rank Sum Test on the Performance Measures of the four Classification Methods, p is the p value and χ^2 is the chi square test statistic, $p < 0.05$ means there is a significant performance difference among the four methods.

	Γ_{ext}				Γ_{end}			
Measure	6 Hour		12 Hour		6 Hour		12 Hour	
	χ^2	p	χ^2	p	χ^2	p	χ^2	p
F Score	4.92	0.177	10.92	**0.0121**	3.7826	0.286	7.8	0.050
Precision	1.08	0.781	12.12	**0.0069**	1.1739	0.759	7.8	0.050
Recall	8.1875	**0.0423**	9.72	**0.021**	7.6957	0.053	4.4667	0.215

Table 7.6 Post-hoc Nemenyi Test on the Performance Measures of the four Classification Methods with 12 hour ahead Forecasting, the p value is recorded, $p < 0.05$ means the RVFL network significantly outperforms the benchmark methods(ANN, RF or SVM).

		Γ_{ext}			Γ_{end}		
		ANN	RF	SVM	ANN	RF	SVM
RVFL	F Score	0.203	0.961	**0.017**	0.20	0.99	0.12
	Precision	0.068	0.995	0.122	0.32	0.99	0.20
	Recall	0.122	0.611	**0.017**	0.53	0.69	0.20

Table 7.7 Computation Time (s) of the Classification Methods over 5 Datasets, ‘Ave’: average/mean, and ‘Sd’: standard deviation

Dataset	6 Hour								12 Hour							
	RF		SVM		ANN		RVFL		RF		SVM		ANN		RVFL	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Γ_{ext}																
D1	1.2	1.13	85.85	0.24	205.42	7.55	1.72	0.09	0.81	0.78	64.36	0.22	151.3	3.95	1.37	0.08
D2	0.88	0.9	75.56	0.2	176.92	4.52	1.7	0.09	0.7	0.67	50.3	0.17	118.95	3.62	1.27	0.06
D3	1.17	1.22	88.48	0.22	181.02	6.27	1.67	0.88	0.67	0.7	57.47	0.16	113.88	4.57	1.22	0.06
D4	0.78	0.81	72.88	0.21	157.63	4.48	1.89	0.08	0.62	0.63	51.01	0.15	122.21	3.71	1.26	0.06
D5	0.98	1.01	92.18	0.33	187.18	5.04	1.73	0.1	0.89	0.85	64.46	0.2	141.37	3.32	1.39	0.08
Ave	1	1	82.99	0.24	181.63	5.57	1.74	0.25	0.74	0.73	57.52	0.18	129.54	3.83	1.30	0.07
Sd	0.182	0.166	8.37	0.05	17.29	1.32	0.09	0.35	0.11	0.09	6.88	0.03	16.00	0.47	0.07	0.01
Γ_{end}																
D1	1.06	1.07	86.03	0.25	214.61	5.55	1.72	0.09	0.98	0.8	70.18	0.21	154.19	4.42	1.55	0.08
D2	0.92	1.12	78.76	0.12	173.92	5.48	1.75	0.08	0.66	0.63	46.34	0.13	115.34	4.64	1.22	0.09
D3	1.14	1.17	89.94	0.25	183.11	5.3	1.76	0.09	0.78	0.98	64.57	0.2	139.76	4.15	1.61	0.05
D4	1	0.78	71.74	0.21	157.38	4.07	1.81	0.08	0.64	0.61	51.28	0.16	124.77	3.2	1.26	0.05
D5	1	1.14	93.96	0.3	176.78	4.92	1.73	0.09	0.85	1.09	71	0.21	146.03	3.52	1.48	0.08
Ave	1.03	1.06	84.09	0.23	181.16	5.06	1.75	0.09	0.78	0.82	60.67	0.18	136.02	3.99	1.42	0.07
Sd	0.08	0.16	8.89	0.07	20.97	0.61	0.04	0.01	0.14	0.31	11.25	0.04	15.80	0.61	0.17	0.02

Chapter 8

Conclusion and Future Work

8.1 Conclusion

This thesis has covered an important topic in renewable energy: time series (TS) forecasting. Renewable energy sources such as wind and solar are usually highly fluctuating, and the fluctuation makes the energy harvesting and grid integration challenging. For example, the fluctuation in the wind speed will cause rapid changes in wind power generated by the wind turbines and these rapid changes, which are known as ramps, may bring instability to the power grid. This thesis has focused on the methodologies to improve the forecasting accuracy for wind speed and solar irradiance forecasting, as well as wind power ramp forecasting.

In this thesis, two most common renewable energy sources: wind speed and solar irradiance have been studied and applied to evaluate the forecasting methods developed in the thesis. In Chapter 2, the nature of wind and solar irradiance have been briefly introduced and the characteristics of wind speed and solar irradiance TS have been analyzed. We can see that both TS are highly non-linear and are not Gaussian distributed. The particular wind speed TS shows neither seasonal (periodic) components nor low frequency trend. Because of the deterministic function of solar elevation angle and sun rise/down time, the solar irradiance TS shows strong daily and yearly cycles. The zero irradiance value in the TS should be exempted out for performance evaluation.

In Chapter 3, state-of-the-art ensemble forecasting methods in the literature have been reviewed. The methods in the literature have been categorized into competitive and cooperative ensemble forecasting methods. In competitive ensemble forecasting, the data and parameter diversity are introduced, which in theory improve the forecasting accuracy. However, the competitive ensemble forecasting requires expensive computing power and time.

In practice, meteorological ensemble forecasting usually updates the prediction in 6 hour interval and only down to mesoscale. Competitive ensemble forecasting is usually used in mid-term and long term forecasting. Cooperative ensemble forecasting is fast in computing and the update interval is shorter. However, since it is mainly based on the historical data, the accuracy is degraded when the forecasting horizon increases. Cooperative ensemble forecasting is usually used in short-term forecasting.

An Empirical Mode Decomposition-Support Vector Regression (EMD-SVR) model for wind speed forecasting has been proposed in Chapter 4. The EMD has been used for feature extraction and the SVR has been trained by the re-constructed feature set. The proposed EMD-SVR model has been evaluated with a collection of wind speed datasets. The comparisons between ANN/SVR and EMD-ANN/EMD-SVR have showed that EMD method has minimal improvement for ANN but has significant improvement for SVR. The comparisons between the proposed model and some reported models have shown that the proposed model outperformed some reported methods. The proposed model had a comparable performance compared with the EMD-SVR with multiple predictor approach. However, the proposed method only require one SVR to do the training and prediction instead of multiple SVR modules, the computation time is faster. The PACF based feature selection has further simplified the model without degrading the performance.

Chapter 5 further researched on the ensemble versions of EMD. An AdaBoost based EMD-ANN has been proposed. It has been shown to improve the performances of ANN and EMD-ANN on wind speed forecasting in terms of accuracy.

In order to solve the mode mixing problem created by EMD, three ensemble EMD methods based on noise-assistance (data diversity) have been discussed as well. The three ensemble versions of EMD have been combined with SVR for wind speed and solar irradiance forecasting. For wind speed forecasting, the ensemble EMD-SVR methods has outperformed the EMD-SVR method and among the three ensemble versions, complete ensemble EMD with adaptive noise (CEEMDAN)-SVR has the best performance because it has smaller forecasting error and shorter computation time. The ensemble EMD (EEMD) and complementary EEMD (CEEMD) based SVR methods have similar performance and therefore, completeness is not a key factor to wind speed TS forecasting. However, due to the nature of solar irradiance TS, the EMD-SVR has suffered from the mode mixing problem and cannot outperform the SVR method. The ensemble EMD versions has attenuated the mode mixing problem embedded in the inherent characteristics of the solar irradiance TS and out of which the EEMD-SVR has the best forecasting accuracy.

Chapters 6 and 7 have introduced a random vector functional link (RVFL) neural network for regression and classification.

The development of the RVFL network has been discussed in Chapter 6 and eight configurations of the network structures have been analyzed with a collection of wind speed TS. It has been shown that the input layer and hidden layer bias do not affect the performance for wind speed forecasting but the direct input output connections have significantly improved the performance. The least square method used in the RVFL network for optimization has an advantage over hierarchical forecasting method which separates the parameter optimization into two passes: linear and non-linear. Compared with other machine learning methods, the RVFL network has outperformed ANN for 1-12 hour ahead wind forecasting, SVR for 1-8 hour ahead forecasting and RF for 4-12 hour ahead wind forecasting.

In the circumstance of solar irradiance forecasting, there is no distinguishable performance difference among the eight configurations of the network. The RVFL network has comparable performance as ARIMA, RF, SVR and ARIMA-SVR methods on solar irradiance forecasting.

The RVFL network has been applied to a classification task: wind power ramp forecasting in Chapter 7. The RVFL network has comparable performance as the benchmark methods: ANN, RF and SVM for 6 hour ahead forecasting but the RVFL network has better performance than the SVM for 12 hour ahead forecasting. The computation time of the RVFL network has significant advantage over SVM and ANN and is comparable as RF.

The forecasting performance on renewable energy sources is highly application sensitive. The characteristics of the TS should be studied first and appropriate methods can be chosen based on the detailed requirements. In this thesis, wind speed and solar irradiance have been evaluated with several machine learning based methods for short-term forecasting. One method that has performed well on wind speed forecasting does not achieve the same performance on solar irradiance forecasting such as CEEMDAN-SVR.

From the results in wind speed forecasting, we can see that the ensemble versions of machine learning methods generally have better performance than the non-ensemble counterparts. For example, AdaBoost-EMD-ANN has better performance than EMD-ANN, and CEEMDAN-SVR has better performance than SVR and EMD-SVR.

For tasks that requires higher accuracy, the ensemble EMD methods are recommended for wind speed and solar irradiance forecasting, in particular, CEEMDAN-SVR for wind speed forecasting and EEMD-SVR for solar irradiance forecasting. Due to the nature of the solar irradiance TS, the EMD-SVR should be avoided since it suffers badly from mode-mixing problem. For wind speed forecasting tasks with fast training, the RVFL network is preferred due to its rapid training and simple structure. Although the random assignment of the weights may cause instability (rank deficiency), it still has advantages over the conventional ANN because of the direct input output connections. The RVFL network can be used for solar irradiance forecasting despite that the direct input output connections do not significantly improve the performance. The simple and rapid training of the RVFL network still makes this method recommended over the SVR, RF and ARIMA methods without sacrificing the forecasting accuracy.

Another advantage of the RVFL network is that it has closed form solution (least square estimation) instead of an iterative solution. In most micro-grid architectures, embedded systems such as ARM or Altmega chips are the main computing power. Closed form algorithm is more preferred in these systems with critical hardware constraint (small memory, low CPU frequency, etc.) due to its simplicity and small computation cost.

8.2 Future Work

The research on renewable energy will be more demanding in the future. The data size and dimensions of the renewable energy sources will increase dramatically and therefore feature extraction and selection will be more and more critical. In this thesis, we applied EMD methods for dataset decomposition and feature extraction. A possible future research area is on adaptive feature selection for individual decomposed components.

There should be more focus on the dataset selection and processing. Instead of using directly related historical dataset for forecasting, multi-variate datasets containing more than one renewable energy sources can be used. Since the renewable energy sources are all originated from the solar energy, and within a certain scaled region and a certain time frame, the TS of different renewable energy sources may have strong correlations. With further study on the correlations, machine learning based forecasting methods can model the renewable energy sources better.

The RVFL network, although it is simple in training and structure, there may be further development on it. The random weight assignment and the range determination of the random weights should be further studied so that the possibility of getting rank deficiency will be lowered. Another future research direction on the RVFL network is the activation function selection. Other than sigmoidal functions, other non-linear function can be applied. In addition, the optimization on the number of hidden neurons should be further studied because the random assignment of the input layer to hidden layer weights may create instability during the cross validation process. Ensemble of the RVFL network is also a potential research area. According to [70, 71], highly unstable methods are naturally suitable for ensembles because the variance of ensemble can be reduced significantly. Therefore, the random weights give rise to the instability of the RVFL network, which is a positive potential for ensemble learning.

Bibliography

[1] (2015) Global wind energy council. [Online]. Available: www.gwec.net/

[2] K. Bullis, “A cheap material boosts solar cells by 50 percent,” *Materials News*, 30 Jan. 2015.

[3] J. B. Bremnes, “A comparison of a few statistical models for making quantile wind power forecasts,” *Wind Energy*, vol. 9, pp. 3–11, 2006.

[4] A. Mellit and S. A. Kalogirou, “Artificial intelligence techniques for photovoltaic applications: A review,” *Progress in Energy and Combustion Science*, vol. 34, pp. 574–632, 2008.

[5] S. Soman, H. Zareipour, O. Malik, and P. Mandal, “A review of wind power and wind speed forecasting methods with different time horizons,” in *Proc. North American Power Symposium (NAPS’10)*, sept. 2010, pp. 1 –8.

[6] S. A. Kalogirou, “Artificial neural networks in renewable energy systems applications: a review,” *Renewable and Sustainable Energy Reviews*, vol. 5, pp. 373–401, 2001.

[7] J. C. Cao and S. H. Cao, “Study of forecasting solar irradiance using neural networks with preprocessing sample data by wavelet analysis,” *Energy*, vol. 31, no. 15, pp. 3435–3445, 2006.

[8] (2015) Forecasting. Wikipedia. [Online]. Available: en.wikipedia.org/wiki/Forecasting

[9] (2015) Supervised learning. Wikipedia. [Online]. Available: en.wikipedia.org/wiki/Supervised_learning

[10] M. Lichman, “UCI machine learning repository,” 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[11] Y. Ren and P. N. Suganthan, “Empirical comparison of bagging-based ensemble classifiers,” in *Proc. International Conference on Information Fusion (Fusion2012)*, Singapore, Jul. 2012, pp. 917–924.

[12] ——, “A kernel-ensemble bagging support vector machine,” in *Proc. International Conference on Intelligent Systems Design and Applications (ISDA2012)*, Brunei, Oct. 2012.

[13] Y. Ren, P. N. Suganthan, N. Srikanth, and S. Sarkar, “A hybrid ARIMA-DENFIS method for wind speed forecasting,” in *Proc. IEEE International Conference on Fuzzy Systems (FUZZ2013)*, Hyderabad, India, Jul. 2013.

[14] Y. Ren, P. N. Suganthan, and N. Srikanth, “Wind speed forecasting: a comparison between statistical approach and learning-based approach,” in *Smart Microgrids: New Advances, Challenges and Opportunities in the Actual Power Systems*, H. M. Khodr, Ed. NY: NOVA, 2013, pp. 1–23.

[15] ——, “Ensemble methods for wind and solar power forecasting: a state-of-the-art review,” *Renewable and Sustainable Energy Reviews*, vol. 50, pp. 82–91, Oct. 2015.

[16] Y. Ren, L. Zhang, and P. N. Suganthan, “Ensemble classification and regression – recent developments, applications and future directions,” *IEEE Comput. Intell. Mag.*, 2016.

[17] Y. Ren and P. N. Suganthan, “EMD-kNN models for wind speed forecasting,” *Journal of Power and Energy Engineering*, vol. 2, no. 4, pp. 176–185, 2014.

[18] Y. Ren, P. N. Suganthan, and N. Srikanth, “A novel empirical mode decomposition with support vector regression for wind speed forecasting,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. PP, Sep. 2014.

[19] ——, “A comparative study of empirical mode decomposition-based short-term wind speed forecasting methods,” *IEEE Trans. Sustain. Energy*, vol. 6, no. 1, pp. 236–244, Jan. 2015.

[20] Y. Ren, X. Qiu, and P. N. Suganthan, “EMD based AdaBoost-BPNN method for wind speed forecasting,” in *Proc. IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL2014)*, Orlando, US, Dec. 2014.

[21] Y. Ren, P. N. Suganthan, N. Srikanth, and G. Amaratunga, “Single hidden layer neural networks with random weights for short-term electricity load demand forecasting,” *Information Sciences*, 2016.

[22] (2015, Apr.) NREL west wind. NREL. [Online]. Available: wind.nrel.gov/Web_nrel/

[23] M. Lydia and S. Kumar, “A comprehensive overview on wind power forecasting,” in *Proc. International Power and Energy Conference (IPEC’10)*, Singapore, 27-29Oct. 2010, pp. 268–273.

[24] D. C. Hill, D. McMillan, K. R. W. Bell, and D. Infield, “Application of Auto-Regressive models to U.K. wind speed data for power system impact studies,” *IEEE Trans. Sustain. Energy*, vol. 3, no. 1, pp. 134–141, 2012.

[25] Y. K. Wu and J. S. Hong, “A literature review of wind forecasting technology in the world,” in *Proc. IEEE Lausanne Power Tech*, 2007, pp. 504–509.

[26] A. Costa, A. Crespo, J. Navarro, G. Lizcano, H. Madsen, and E. Feitosa, “A review on the young history of the wind power short-term prediction,” *Renewable and Sustainable Energy Reviews*, vol. 12, pp. 1725–1744, 2008.

[27] M. Lei, L. Shiyan, J. Chuanwen, L. Hongling, and Z. Yan, “A review on the forecasting of wind speed and generated power,” *Renewable and Sustainable Energy Reviews*, vol. 13, no. 4, pp. 915–920, 2009.

[28] A. M. Foley, P. Leahy, and E. McKeogh, “Wind power forecasting and prediction methods,” in *Proc. International Conference on Environment and Electrical Engineering (EEEIC2010)*, Prague, Czech Republic, 16-19 May 2010, pp. 61–64.

[29] F. Wang, Z. Mi, S. Su, and H. Zhao, "Short-term solar irradiance forecasting model based on artificial neural network using statistical feature parameters," *Energies*, vol. 5, pp. 1355–1370, 2012.

[30] A. Mellit, H. Eleuch, M. Benghanem, C. Elaoun, and A. M. Pavan, "An adaptive model for predicting of global, direct and diffuse hourly solar irradiance," *Energy Conversion and Management*, vol. 51, pp. 771–782, 2010.

[31] H. M. Diagne, M. David, and J. Boland, "Solar irradiation forecasting: state-of-the-art and proposition for future developments for small-scale insular grids," in *Proc. American Solar Energy Society National Solar Conference (SOLAR2012)*, Denver, CO, 2012.

[32] J. Kleissl, "Current state of the art in solar forecasting," in *Proc. California Renewable Energy Forecasting, Resource Data and Mapping*. California Institute for Energy and Environment, 2010, pp. A1–A26.

[33] D. Heinemann, E. Lorenz, and M. Girodo, "Forecasting of solar radiation," in *Solar Resource Management for Electricity Generation from Local Level to Global Scale*, E. Dunlop, L. Wald, and M. Suri, Eds. Nova Science Publishers: NY, 2006, pp. 83–94.

[34] (2015) Time series. Wikipedia. [Online]. Available: en.wikipedia.org/wiki/Time_series

[35] V. Zoonekynd. (2007, Jan.) Time series. [Online]. Available: http://zoonek2.free.fr/UNIX/48_R/15.html

[36] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 832–837, 1956.

[37] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications, with R examples*. NY:Springer, 2011.

[38] R. J. Hyndman and Y. Kh, "Automatic time series forecasting: The forecast package for R," *Journal of Statistical Software*, 2008.

[39] A. I. McLeod, H. Yu, and E. Mahdi, "Time series analysis with R," *Handbook of Statistics*, vol. 30, Jul. 2011.

[40] D. A. Dickey and W. A. Fuller, "Distribution of the estimators for autoregressive time series with a unit root," *Journal of the American Statistical Association*, vol. 74, no. 366, pp. 427–431, 1979.

[41] G. Elliott, T. J. Rothenberg, and J. H. Stock, "Efficient tests for an autoregressive unit root," *Econometrica*, vol. 64, no. 4, pp. 813–836, 1996.

[42] P. C. B. Phillips and P. Perron, "Testing for a unit root in time series regression," *Biometrika*, vol. 75, no. 2, pp. 335–346, 1988.

[43] D. Kwiatkowski, P. C. B. Phillips, P. Schmidt, and Y. Shin, "Testing the null hypothesis of stationarity against the alternative of a unit root," *Journal of Econometrics*, vol. 54, no. 1-3, pp. 159–178, 1992.

[44] (2016) Climate of singapore. [Online]. Available: www.weather.gov.sg/climate-climate-of-singapore/

[45] (2015, Feb.) National solar radiation data base. NREL. [Online]. Available: http://rredc.nrel.gov/solar/old_data/nsrdb/

[46] C. Chatfield, *Time-series forecasting*. CRC Press, 2000.

[47] J. D. Cryer and K.-S. Chan, *Time series analysis with applications in R*, 2nd ed., G. Casella, S. Fienberg, and I. Olkin, Eds. NY: Springer, 2008.

[48] J. W. Taylor, "Short-term load forecasting with exponentially weighted methods," *IEEE Trans. on Power Syst.*, vol. 27, no. 1, pp. 458–464, 2012.

[49] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*. NY:Springer, 2002.

[50] G. Eshel. (2010) The Yule Walker equations for the AR coefficients. [Online]. Available: http://www-stat.wharton.upenn.edu/~steele/Courses/956/ResourceDetails/YWSourceFiles/YW-Eshel.pdf

[51] J. W. Taylor and P. E. McSharry, "Short-term load forecasting methods: An evaluation based on European data," *IEEE Trans. on Power Syst.*, vol. 22, no. 4, pp. 2213–2219, 2007.

[52] C.-M. Huang, C.-J. Huang, and M.-L. Wang, "A particle swarm optimization to identifying the ARMAX model for short-term load forecasting," *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 1126–1133, May 2005.

[53] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[54] R. Lei, X. Kong, and X. Wang, "An ensemble SVM using entropy-based attribute selection," in *Proc. Chinese Control and Decision Conference*, May 2010, pp. 802–805.

[55] A. Smola and B. Schölkopf, "A tutorial on support vector regression," NeuroCOLT, Royal Holloway College, University of London, UK, Tech. Rep. NC-TR-98-030, Oct. 1998.

[56] D. Basak, S. Pal, and D. C. Patranabis, "Support vector regression," *Neural Information Processing Letters and Reviews*, vol. 11, no. 10, pp. 203–224, Sep. 2007.

[57] C.-N. Ko and C.-M. Lee, "Short-term load forecasting using SVR (support vector regression)-based radial basis function neural network with dual extended kalman filter," *Energy*, vol. 49, pp. 413–422, 2013.

[58] E. E. Elattar, J. Goulermas, and Q. H. Wu, "Electric load forecasting based on locally weighted support vector regression," *IEEE Trans. Systems, Man, and Cybernetics–Part C: Applications and Reviews*, vol. 40, pp. 438–447, Jul. 2010.

[59] M. Han, J. Xi, S. Xu, and F.-L. Yin, "Prediction of chaotic time series based on the recurrent predictor neural network," *IEEE Trans. Signal Process.*, vol. 52, no. 12, pp. 3409–3416, 2004.

[60] T. Barbounis, J. Theocharis, M. Alexiadis, and P. Dokopoulos, "Long-term wind speed and power forecasting using local recurrent neural network models," *IEEE Trans. Energy Conv.*, vol. 21, no. 1, pp. 273–284, Mar. 2006.

[61] A. Kavousi-Fard, H. Samet, and F. Marzbani, "A new hybrid modified firefly algorithm and support vector regression model for accurate short term load forecasting," *Expert Systems with Applications*, vol. 41, pp. 6047–6056, 2014.

[62] S. Salcedo-Sanz, E. G. Ortiz-Garcia, A. M. Perez-Bellido, A. Portilla-Figueras, and L. Prieto, "Short term wind speed prediction based on evolutionary support vector regression algorithms," *Expert Systems with Applications*, vol. 38, no. 4, pp. 4052–4057, 2011.

[63] N. Amjady, F. Keynia, and H. Zareipour, "Wind power prediction by a new forecast engine composed of modified hybrid neural network and enhanced Particle Swarm Optimization," *IEEE Trans. Sustain. Energy*, vol. 2, no. 3, pp. 265–276, Jul. 2011.

[64] M. Zhu and L. Wang, "Intelligent trading using support vector regression and multilayer perceptrons optimized with genetic algorithms." in *Proc. International Joint Conference on Neural Networks (IJCNN'10)*, 2010, pp. 1–5.

[65] R. Fang, "A hybrid rough sets and support vector regression approach to short-term electricity load forecasting," in *Proc. IEEE Power and Energy Society General Meeting*, no. 1-5, PA, US, Jul. 2008.

[66] O. Castillo and P. Melin, "Hybrid intelligent systems for time series prediction using neural networks, fuzzy logic, and fractal theory," *IEEE Trans. Neural Netw.*, vol. 13, no. 6, pp. 1395–1408, 2002.

[67] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *Journal of Artificial Intelligence Research*, vol. 11, pp. 169–196, 1999.

[68] P. Manousos. (2006, Jul.) Ensemble prediction systems. Hydrometeorological Prediction Center. [Online]. Available: http://www.wpc.ncep.noaa.gov/ensembletraining/

[69] A. Krogh, J. Vedelsby *et al.*, "Neural network ensembles, cross validation, and active learning," *Advances in neural information processing systems*, pp. 231–238, 1995.

[70] G. Brown, J. Wyatt, R. Harris, and X. Yao, "Diversity creation methods: a survey and categorisation," *Information Fusion*, vol. 6, no. 1, pp. 5–20, 2005.

[71] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural computation*, vol. 4, no. 1, pp. 1–58, 1992.

[72] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[73] E. Kleinberg, "Stochastic discrimination," *Annals of Mathematics and Artificial intelligence*, vol. 1, no. 1, pp. 207–239, 1990.

[74] R. E. Schapire and Y. Freund, "Boosting the margin: a new explanation for the effectiveness of voting methods," *The Annals of Statistics*, vol. 26, pp. 322–330, 1998.

[75] V. Pisetta and D. A. Zighed, "New insights into decision tree ensembles," Ph.D. dissertation, Lyon 2, 2012.

[76] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Machine Learning*, vol. 40, no. 2, pp. 139–157, 2000.

[77] E. K. Tang, P. N. Suganthan, and X. Yao, "An analysis of diversity measures," *Machine Learning*, vol. 65, no. 1, pp. 247–271, 2006.

[78] J. Hall and J. Hall, "Forecasting solar radiation for the Los Angeles basin – phase II report," in *Proc. American Solar Energy Society National Solar Conference (SOLAR2011)*, Raleigh, NC, 2011.

[79] A. Chaouachi, R. M. Kamel, R. Ichikawa, H. Hayashi, and K. Nagasaka, "Neural network ensemble-based solar power generation short-term forecasting," *World Academy of Science, Engineering and Technology*, vol. 54, pp. 54–59, 2009.

[80] S. B. Taieb and R. J. Hyndman, "A gradient boosting approach to the kaggle load forecasting competition," *International Journal of Forecasting*, vol. 30, no. 2, pp. 382–394, 2014.

[81] J. Wu, B. Zhang, and K. Wang, "Application of Adaboost-based BP neural network for short-term wind speed forecast," *Power System Technology*, vol. 36, no. 9, pp. 221–225, Sep 2012.

[82] A. Sozen, E. Arcaklýogcblu, M. Ozalp, and E. G. Kanýt, "Use of artificial neural networks for mapping the solar potential in Turkey," *Applied Energy*, vol. 77, pp. 273–286, 2004.

[83] I. Damousis and P. Dokopoulos, "A fuzzy expert system for the forecasting of wind speed and power generation in wind farms," in *Proc. IEEE Power Engineering Society International Conference on Power Industry Computer Applications (PICA2001)*, 20-24 May 2001, pp. 63–69.

[84] T. Barbounis and J. Theocharis, "Locally recurrent neural networks for wind speed prediction using spatial correlation," *Information Sciences*, vol. 177, no. 24, pp. 5775–5797, 2007.

[85] H. Nielsen, T. Nielsen, H. Madsen, J. Badger, G. Giebel, L. Landberg, K. Sattler, and H. Feddersen, "Wind power ensemble forecasting," in *Proc. Global Windpower Conference*, IL, US, Mar. 2004.

[86] L. Magnusson, M. Leutbecher, and E. Källén, "Comparison between singular vectors and breeding vectors as initial perturbations for the ECMWF ensemble prediction system," *Monthly Weather Review*, vol. 136, no. 11, pp. 4092–4104, Dec. 2007.

[87] E. Kalnay, *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge University Press, 2007.

[88] P. Pinson and R. Hagedorn, "Verification of the ECMWF ensemble forecasts of wind speed against analyses and observations," *Meteorological Applications*, vol. 19, no. 4, pp. 484–500, 2012.

[89] S. Lang. (2007) Wind energy forecasting in the Irish power system using a multi-scheme ensemble prediction system. [Online]. Available: http://www.seai.ie/Publications/Renewables_Publications_/Wind_Power/Wind_Energy_Forecasting_on_the_Irish_Power_Systems-using_a_Multi-Scheme_Ensemble_Prediction_System.5220.shortcut

[90] Z. Wei and W. Weimin, "Wind speed forecasting via ensemble Kalman filter," in *Proc. International Conference on Advanced Computer Control (ICACC2010)*, vol. 2, Shenyang, China, 27-29 Mar 2010, pp. 73–77.

[91] M. De Felice and X. Yao, "Short-term load forecasting with neural network ensembles: A comparative study [application notes]," *IEEE Comput.l Intell. Mag.*, vol. 6, no. 3, pp. 47–56, 2011.

[92] M. Alhamdoosh and D. Wang, “Fast decorrelated neural network ensembles with random weights,” *Information Sciences*, vol. 264, pp. 104–117, 2014.

[93] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, pp. 123–140, 1996.

[94] D. Shrestha and D. Solomatine, “Experiments with AdaBoost.RT, an improved boosting scheme for regression,” *Neural Computation*, vol. 18, no. 7, pp. 1678–1710, 2006.

[95] Y. Freund, R. E. Schapire *et al.*, “Experiments with a new boosting algorithm,” in *Proc. International Conference on Machine Learning (ICML’96)*, vol. 96, 1996, pp. 148–156.

[96] J. H. Friedman, “Stochastic gradient boosting,” *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[97] J. R. Lloyd, “GEFCom2012 hierarchical load forecasting: Gradient boosting machines and gaussian processes,” *International Journal of Forecasting*, vol. 30, no. 2, pp. 369–374, Apr. 2014.

[98] R. Feely, “Predicting stock market volatility using neural networks,” Master’s thesis, Trinity College Dublin, 2000, B.A dissertation.

[99] J. Friedman, T. Hastie, R. Tibshirani *et al.*, “Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors),” *The Annals of Statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[100] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of Computer and System Sciences*, vol. 55, pp. 119–139, 1997.

[101] P. Kankanala, S. Das, and A. Pahwa, “Adaboost^{+} : An ensemble learning approach for estimating weather-related outages in distribution systems,” *IEEE Trans. Power Syst.*, vol. 29, no. 1, pp. 359–367, Jan. 2014.

[102] H. Nielsen, H. Madsen, T. Nielsen, J. Badger, G. Giebel, L. Landberg, K. Sattler, and H. Feddersen, “Wind power ensemble forecasting using wind speed and direction ensembles from ECMWF or NCEP,” Denmark Project PSO, Tech. Rep. NEI-DK-4552, 2005.

[103] T. M. Hamill, C. Snyder, and S. L. Mullen, “Ensemble forecasting in the short to medium range: Report from a workshop,” *Bulletin of the American Meteorological Society*, Nov. 2000.

[104] Y. Liu and X. Yao, “Ensemble learning via negative correlation,” *Neural Networks*, vol. 12, no. 10, pp. 1399–1404, 1999.

[105] H. Liu, H.-Q. Tian, C. Chen, and Y.-F. Li, “A hybrid statistical method to predict wind speed and wind power,” *Renewable Energy*, vol. 35, pp. 1857–1861, 2010.

[106] J. Catalão, H. Pousinho, and V. Mendes, “Hybrid Wavelet-PSO-ANFIS approach for short-term wind power forecasting in Portugal,” *IEEE Trans. Sustain. Energy*, vol. 2, no. 1, pp. 50–59, Jan. 2011.

[107] R.-A. Hooshmand, H. Amooshahi, and M. Parastegari, “A hybrid intelligent algorithm based short-term load forecasting approach,” *Electrical Power and Energy Systems*, vol. 45, pp. 313–324, 2013.

[108] M. Moazzami, A. Khodabakhshian, and R. Hooshmand, "A new hybrid day-ahead peak load forecasting method for Iran's national grid," *Applied Energy*, vol. 101, pp. 489–501, 2013.

[109] C. Guan, P. B. Luh, L. D. Michel, Y. Wang, and P. B. Friedland, "Very short-term load forecasting: Wavelet neural networks with data pre-filtering," *IEEE Trans. Power Syst.*, vol. 28, no. 1, pp. 30–31, 2013.

[110] L. Ghelardoni, A. Ghio, and D. Anguita, "Energy load forecasting using empirical mode decomposition and support vector regression," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 549–556, Mar. 2013.

[111] G. F. Fan, S. Qing, H. Wang, W. C. Hong, and H. J. Li, "Support vector regression model based on empirical mode decomposition and auto regression for electric load forecasting," *Energies*, vol. 6, pp. 1887–1901, 2013.

[112] T. Kitajima and T. Yasuno, "Output prediction of wind power generation system using complex-valued neural network," in *Proc. SICE Annual Conference (SICE2010)*, Taipei, Taiwan, 18-21 Aug 2010, pp. 3610–3613.

[113] A. Lojowska, D. Kurowicka, G. Papaefthymiou, and L. van der Sluis, "Advantages of ARMA-GARCH wind speed time series modeling," in *Proc. IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS2010)*, Singapore, 14-17 Jun 2010, pp. 83–88.

[114] J. Shi, J. Guo, and S. Zheng, "Evaluation of hybrid forecasting approaches for wind speed and power generation time series," *Renewable and Sustainable Energy Reviews*, vol. 16, pp. 3471–3480, 2012.

[115] J. Wu and C. C. Keong, "Prediction of hourly solar radiation using a novel hybrid model of ARMA and TDNN," *Solar Energy*, vol. 85, no. 5, pp. 808–817, 2011.

[116] G.-R. Ji, P. Han, and Y.-J. Zhai, "Wind speed forecasting based on Support Vector Machine with forecasting error estimation," in *Proc. International Conference on Machine Learning and Cybernetics (ICMLC2007)*, vol. 5, Hong Kong, Aug. 2007, pp. 2735–2739.

[117] Y. Kemmoku, S. Orita, S. Nakagawa, and T. Sakakibara, "Daily insolation forecasting using a multi-stage neural network," *Solar Energy*, vol. 66, no. 3, pp. 193–199, 1999.

[118] L. P. Wang, K. Teo, and Z. Lin, "Predicting time series with wavelet packet neural networks," in *Proc. IEEE International Joint Conference on Neural Networks (IJCNN2001)*, 2001, pp. 1593–1597.

[119] N. Pindoriya, S. Singh, and S. Singh, "An adaptive wavelet neural network-based energy price forecasting in electricity markets," *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 1423–1432, 2008.

[120] N. Huang, Z. Shen, S. Long, M. Wu, H. Shih, Q. Zheng, N. Yen, C. Tung, and H. Liu, "The empirical mode decomposition and Hilbert spectrum for nonlinear and nonstationary time series analysis," *Proc. Royal Society London A*, vol. 454, pp. 903–995, 1998.

[121] I. Damousis, M. Alexiadis, J. Theocharis, and P. Dokopoulos, "A fuzzy model for wind speed prediction and power generation in wind parks using spatial correlation," *IEEE Trans. Energy Convers.*, vol. 19, no. 2, pp. 352 – 361, june 2004.

[122] G. Rilling, P. Flandrin, and P. Gonçalvès, "On empirical mode decomposition and its algorithms," in *Proc. IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP'03)*, no. 3, Grado, Italy, 2003, pp. 8–11.

[123] E. A. Wan, "Finite impulse response neural networks for autoregressive time series prediction," in *Proc. NATO Advanced Workshop on Time Series Prediction and Analysis*, Sante Fe, NM, May 1993.

[124] L. Ye and P. Liu, "Combined model based on EMD-SVM for short-term wind power prediction," in *Proc. Chin. Soc. Elec. Eng.*, vol. 31, no. 31, 5 Nov. 2011, pp. 102–108.

[125] X. Wang and H. Li, "One-month ahead prediction of wind speed and output power based on EMD and LSSVM," in *Proc. IEEE International Conference on Energy and Environment Technology (ICEET'09)*, vol. 3, 2009, pp. 439–442.

[126] H. Liu, C. Chen, H. Tian, and Y. Li, "A hybrid model for wind speed prediction using empirical mode decomposition and artificial neural networks," *Renewable Energy*, vol. 48, pp. 545–556, 2012.

[127] Z. Zheng, Y. Chen, X. Zhou, M. Huo, B. Zhao, and M. Guo, "Short-term wind power forecasting using empirical mode decomposition and RBFNN," *International Journal of Smart Grid and Clean Energy*, vol. 2, no. 2, pp. 192–199, May 2013.

[128] Z. Guo, W. Zhao, H. Lu, and J. Wang, "Multi-step forecasting for wind speed using a modified EMD-based artificial neural network model," *Renewable Energy*, vol. 37, pp. 241–249, 2012.

[129] R. Li and Y. Wang, "Short-term wind speed forecasting for wind farm based on empirical mode decomposition," in *Proc. International Conference on Electrical Machines and Systems (ICEMS'08)*, Wuhan, China, 17-20 Oct. 2008, pp. 2521–2525.

[130] C. Sun, Y. Yuan, and Q. Li, "A new method for wind speed forecasting based on empirical mode decomposition and improved persistence approach," in *Proc. International Conference on Power & Energy (IPEC'12)*, Ho Chi Minh City, Vietnam, 2012, pp. 659–664.

[131] X. Liu, Z. Mi, P. Li, and H. Mei, "Study on the multi-step forecasting for wind speed based on EMD," in *Proc. International Conference on Sustainable Power Generation and Supply (SUPERGEN'09)*, Nanjing, China, 6-7 Apr. 2009, pp. 1–5.

[132] X. An, D. Jiang, M. Zhao, and C. Liu, "Short-term prediction of wind power using EMD and chaotic theory," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, pp. 1036–1042, 2012.

[133] Z. Han and X. Zhu, "Training set of support vector regression extracted by empirical mode decomposition," in *Proc. Asia-Pacific Power and Energy Engineering Conference (APPEEC'11)*, Wuhan, China, 25-28 Mar. 2011, pp. 1–4.

[134] X.-J. Liu, Z.-Q. Mi, B. Lu, and W. Tao, "A novel approach for wind speed forecasting based on EMD and time-series analysis," in *Proc. Asia-Pacific Power and Energy Engineering Conference (APPEEC'09)*, Wuhan, China, 27-31 Mar. 2009, pp. 1–4.

[135] (2015, Jan.) National data buoy center historical data. [Online]. Available: http://www.ndbc.noaa.gov/historical_data.shtml

[136] E. Dimitriadou, K. Hornik, F. Leisch, D. Meyer, and A. Weingessel, "Misc functions of the department of statistics (e1071), tu wien," *R package*, pp. 1–5, 2008.

[137] D. Kim and H.-S. Oh, "EMD: A package for empirical mode decomposition and Hilbert spectrum," *The R Journal*, vol. 1, no. 1, pp. 40–46, 2009.

[138] P. Mundra and J. C. Rajapakse, "SVM-RFE with mRMR filter for gene selection," *IEEE Trans. NanoBioscience*, vol. 9, no. 1, pp. 31–37, 2010.

[139] N. De Jay, S. Papillon-Cavanagh, C. Olsen, N. El-Hachem, G. Bontempi, and B. Haibe-Kains, "mRMRe: an R package for parallelized mRMR ensemble feature selection," *Bioinformatics*, pp. 1—4, Jul. 2013.

[140] C. Ding and H. Peng, "Minimum redundancy feature selection from microarray gene expression data," *Journal of bioinformatics and computational biology*, vol. 3, no. 02, pp. 185–205, 2005.

[141] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[142] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Information Sciences*, vol. 180, no. 10, pp. 2044–2064, 2010.

[143] G. D. C. Cavalcanti, J. Magalhaes, R. Barreto, and T. I. Ren, "MLPBoost: A combined adaboost / multi-layer perceptron network approach for face detection," in *Proc. IEEE International Conference on Systems, Man, and Cybernetics (SMC2012)*, Oct 2012, pp. 2350–2353.

[144] P.-F. Pai, L.-L. Li, W.-Z. Hung, and K.-P. Lin, "Using ADABOOST and rough set theory for predicting debris flow disaster," *Water Resources Management*, vol. 28, no. 4, pp. 1143–1155, 2014.

[145] X. Zhao, B. Ning, L. Liu, and G. Song, "A prediction model of short-term ionospheric foF2 based on adaboost," *Advances in Space Research*, vol. 53, no. 3, pp. 387–394, 2014.

[146] M. Assaad, R. Boné, and H. Cardot, "A new boosting algorithm for improved time-series forecasting with recurrent neural networks," *Information Fusion*, vol. 9, no. 1, pp. 41–55, 2008.

[147] H. Drucker, "Improving regressors using boosting techniques," in *Proc. International Conference on Machine Learning (ICML1997)*, vol. 97, 1997, pp. 107–115.

[148] B. Mac Namee, P. Cunningham, S. Byrne, and O. I. Corrigan, "The problem of bias in training data in regression problems in medical decision support," *Artificial intelligence in medicine*, vol. 24, no. 1, pp. 51–70, 2002.

[149] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: a noise-assisted data ananlysis method," *Advances in Adaptive Data Analysis*, vol. 1, no. 1, pp. 1–41, 2009.

[150] J.-R. Yeh, J.-S. Shieh, and N. E. Huang, "Complimentary ensemble empirical mode decomposition: a novel noise enhanced data analysis method," *Advances in Adaptive Data Analysis*, vol. 2, no. 2, pp. 135–156, 2010.

[151] M. Torres, M. Colominas, G. Schlotthauer, and P. Flandrin, “A complete ensemble empirical mode decomposition with adaptive noise,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’11)*, no. 1520-6149, Prague, Czech Republic, 22-27 May 2011.

[152] W. F. Schmidt, M. A. Kraaijveld, and R. P. Duin, “Feedforward neural networks with random weights,” in *Proc. IEEE International Conference on Pattern Recognition Conference B: Pattern Recognition Methodology and Systems*, 1992, pp. 1–4.

[153] Y.-H. Pao, G.-H. Park, and D. J. Sobajic, “Learning and generalization characteristics of the random vector functional-link net,” *Neurocomputing*, vol. 6, no. 2, pp. 163–180, 1994.

[154] C. L. P. Chen, “A rapid supervised learning neural network for function interpolation and approximation,” *IEEE Trans. Neural Netw.*, vol. 7, no. 5, pp. 1220–1230, Sep. 1996.

[155] C. P. Chen and J. Z. Wan, “A rapid learning and dynamic stepwise updating algorithm for flat neural networks and the application to time-series prediction,” *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 29, no. 1, pp. 62–72, 1999.

[156] C.-N. Ko and C.-M. Lee, “Short-term load forecasting using SVR (support vector regression)-based radial basis function neural network with dual extended kalman filter,” *Energy*, vol. 49, pp. 413–422, 2013.

[157] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol. 2, no. 5, pp. 359–366, Jul. 1989.

[158] T. Ouyang, X. Zha, and L. Qin, “A survey of wind power ramp forecasting,” *Energy and Power Engineering*, vol. 5, pp. 368–372, Jan. 2013.

[159] H. Zareipour, D. Huang, and W. Rosehart, “Wind power ramp events classification and forecasting: A data mining approach,” in *Proc. IEEE Power and Energy Society General Meeting*. IEEE, 2011, pp. 1–3.

[160] J. Zhang, A. Florita, B. M. Hodge, and J. Freedman, “Ramp forecasting performance from improved short-term wind power forecasting,” National Renewable Energy Laboratory (NREL), Tech. Rep. NREL/CP-5D00-61730, May 2014.

[161] M. Cui, D. Ke, Y. Sun, D. Gan, J. Zhang, and B.-M. Hodge, “Wind power ramp event forecasting using a stochastic scenario generation method,” *IEEE Trans. Sustain. Energy*, vol. 6, no. 2, pp. 422–433, 2015.

[162] N. Francis, “Predicting sudden changes in wind power generation,” *North American Windpower*, vol. 5, pp. 58–60, 2008.

[163] R. Girard, A. Bossavy, and G. Kariniotakis, “Forecasting ramps of wind power production at different time scales,” in *European Wind Energy Conference*, 2011.

[164] H. Zheng and A. Kusiak, “Prediction of wind farm power ramp rates: A data-mining approach,” *Journal of Solar Energy Engineering*, vol. 131, no. 3, pp. 031 011–1–031 011–8, 2009.

[165] A. Bossavy, R. Girard, and G. Kariniotakis, “Forecasting uncertainty related to ramps of wind power production,” in *European Wind Energy Conference and Exhibition 2010, EWEC 2010*, vol. 2. European Wind Energy Association, 2010, pp. 1–6, hal-00812403.

[166] ——, “A novel methodology for comparison of different wind power ramp characterization approaches,” in *EWEA 2013-European Wind Energy Association annual event*, 2013, pp. 1–6.

[167] C. Ferreira, J. Gama, L. Matias, A. Botterud, and J. Wang, “A survey on wind power ramp forecasting,” Argonne National Laboratory (ANL), Tech. Rep. ANL/DIS-10-13, 2010.

[168] S. Linden, B. Myers, and S. E. Haupt, “Observation-based wind-power ramp forecast system,” in *Proc. American Meteorological Sciety Annual Meeting*, Jan. 22-26 2012.

[169] H. He and E. Garcia, “Learning from imbalanced data,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sept 2009.

[170] P. Rosenmai. (2013, Nov.) Using the median absolute deviation to find outliers. Eurekastatistics. [Online]. Available: eurekastatistics.com/using-the-median-absolute-deviation-to-find-outliers

[171] Z. Wu and N. E. Huang, “A study of the characteristics of white noise using the empirical mode decomposition method,” in *Proc.Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 460, no. 2046. The Royal Society, 2004, pp. 1597–1611.

[172] Z. Wu and N. E. . Huang, *Hilbert-Huang transform and its applications*. World Scientific, 2005, vol. 5, ch. Statistical Significance Test of Intrinsic Mode Functions, pp. 107–125.

[173] (2015, Apr.) Wind power generation data. ELIA. [Online]. Available: www.elia.be/en/grid-data/power-generation/wind-power

[174] D. J. Hand, “Measuring classifier performance: a coherent alternative to the area under the roc curve,” *Machine Learning*, vol. 77, pp. 103–123, 2009.

[175] ——, “Evaluating diagnostic tests: the area under the roc curve and the balance of errors,” *Statistics in Medicine*, vol. 29, pp. 1502–1510, 2009.

[176] D. J. Hand and C. Anagnostopoulos, “A better beta for the H measure of classification performance,” *Statistics in Medicine*, 2012, arXiv:1202.2564v1.