Ensemble Time Series Forecasting with Applications in Power Systems and Financial Markets

Xueheng Qiu

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

Oct 2018

Declaration of Authorship

I, QIU XUEHENG, declare that this thesis titled, 'Ensemble Time Series Forecasting with Applications in Power Systems and Financial Markets' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this report has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the report is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Abstract

In recent years, time series forecasting has obtained significant academic and industrial interest with its significance in various application fields, including power system related applications (electric load, wind power and solar irradiance forecasting, etc.), as well as financial market related applications (stock price, exchange rate and electricity price forecasting, etc.). Many statistics based machine learning models have been proposed to obtain accurate results for time series forecasting in the literature. The methods can be divided into two categories: linear models (such as auto-regressive moving average) and non-linear models (such as artificial neural network and support vector machine). However, due to the highly nonlinear characteristics of real world time series signals caused by various influencing factors, it is very difficult to ensure the performance of machine learning models. Deep learning and ensemble methods are possible solutions to this problem.

This thesis mainly focuses on the state-of-the-art ensemble learning methods and deep learning models for both power system and financial market related time series forecasting. The development of time series forecasting is introduced, and a brief review of existing algorithms is also recorded. Motivated by the attractive advantages of ensemble learning, two deep learning based ensemble methods are presented: (i) ensemble method composed of deep belief networks and support vector machines, (ii) empirical mode decomposition (EMD) based ensemble deep learning model. The performance of the proposed methods is evaluated by real world time series datasets. On the other hand, the ensemble methods based on fast learning models are also investigated in this thesis, such as decision tree ensembles and random vector functional link (RVFL) network based hybrid models. Specifically, a novel decomposition method composed of discrete wavelet transform and EMD is combined with incremental RVFL for electric load forecasting. Finally, the advantages and potential future developments of deep learning and ensemble methods are discussed.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Ponnuthurai Nagaratnam Suganthan, for motivation, guidance and support throughout my research program and for the opportunity to undertake this study work at Nanyang Technological University. I feel honored to be able to work under his guidance.

I would like to thank the Cambridge Centre for Carbon Reduction in Chemical Technology (C4T) project for funding and scholarship.

I also express my gratitude to my committee members of School of Electrical and Electronic Engineering for their helpful comments and suggestions on this work. I would also like to acknowledge all my friends for their cooperation throughout my studies.

Contents

Declaration of Authorship i

Abstract ii

Acknowledgements iii

List of Figures viii

List of Tables x

Abbreviations xii

1 Introduction 1
- 1.1 Motivations . . . 1
- 1.2 Learning Algorithms for Time Series Forecasting . . . 2
- 1.3 Ensemble Methods . . . 4
- 1.4 Incremental Learning . . . 5
- 1.5 Outline of the Thesis . . . 6

2 Theoretical Background of Forecasting Models 8
- 2.1 Data Preprocessing . . . 8
  - 2.1.1 Data Cleaning . . . 8
  - 2.1.2 Normalization . . . 9
  - 2.1.3 Feature Extraction . . . 9
- 2.2 Forecasting Methods . . . 10
  - 2.2.1 Linear Methods . . . 10
  - 2.2.2 Artificial Neural Network . . . 11
  - 2.2.3 Random Vector Functional Link Neural Network . . . 12
  - 2.2.4 Support Vector Regression . . . 13
  - 2.2.5 Kernel Ridge Regression . . . 16
  - 2.2.6 Random Forest . . . 16
  - 2.2.7 Deep Belief Network . . . 17
- 2.3 Ensemble Learning . . . 19
  - 2.3.1 Competitive Ensemble Learning . . . 20
  - 2.3.2 Cooperative Ensemble Learning . . . 22

I Ensemble Learning with Applications in Power Systems 26

3 Oblique Random Forest Ensemble via Least Square Estimation for Electric Load Forecasting 27
3.1 Characteristics of Electric Load Data . . . 27
3.2 Proposed Ensemble Method with Oblique Random Forest . . . 29
3.2.1 Oblique Random Forest . . . 30
3.3 Experiment Setup . . . 31
3.3.1 Methodology . . . 31
3.3.2 Performance Estimation . . . 32
3.4 Assessment on generic time series datasets . . . 33
3.5 Experiments of electricity load demand time series forecasting . . . 35
3.6 Summary . . . 37

4 Ensemble Deep Learning for Electric Load Time Series Forecasting 39
4.1 Ensemble Deep Belief Network . . . 39
4.1.1 Datasets . . . 40
4.1.2 Results . . . 41
4.1.3 Summary for Ensemble DBN . . . 43
4.2 Empirical Mode Decomposition based Ensemble Deep Belief Network . . . 43
4.2.1 Experiment Setup . . . 44
4.2.2 Results and Comparison . . . 46
4.2.2.1 Performance Comparison for Half-an-hour ahead Load Forecasting . . . 46
4.2.2.2 Performance Comparison for One-day ahead Load Forecasting 47
4.2.3 Comparative Experiments . . . 48
4.2.4 Summary for EMD-DBN . . . 51

5 Ensemble Incremental Learning Random Vector Functional Link Network for Short-term Electric Load Forecasting 52
5.1 Incremental Learning with RVFL . . . 52
5.2 Proposed incremental RVFL based ensemble method . . . 53
5.3 Experiment Setup . . . 54
5.3.1 Datasets . . . 54
5.3.2 Variations of RVFL network . . . 56
5.3.3 Methodology . . . 57
5.3.4 Error Measurement . . . 57
5.4 Results and Discussion . . . 57
5.4.1 Effect of Functional Links and Number of Hidden Neurons . . . 57
5.4.2 Effect of Incremental Learning . . . 59
5.4.3 Performance Comparison with Benchmarks . . . 60
5.4.4 Computation time comparison . . . 64
5.5 Comparative experiment . . . 65
5.6 Summary . . . 65

6 Short-term Wind Power Ramp Forecasting with Empirical Mode Decomposition based Ensemble Learning Techniques 67
6.1 Wind Power Ramp . . . 68

6.2 Proposed Ensemble Method . . . 69
6.3 Experimental Setup . . . 70
6.3.1 Datasets . . . 70
6.3.2 Data Preprocessing . . . 71
6.3.3 Cross Validation . . . 72
6.3.4 Performance Measures . . . 72
6.4 Results and Discussions . . . 73
6.4.1 Wind Power Forecasting . . . 73
6.4.2 Wind Power Ramp Rate Forecasting . . . 74
6.4.3 Wind Power Ramp Classification . . . 76
6.4.4 Computation Time . . . 77
6.5 Summary . . . 77

7 Summary of Part I 79

II Ensemble Learning with Applications in Financial Markets 83

8 Short-term Electricity Price Forecasting with Empirical Mode Decomposition based Ensemble Kernel Machines 84
8.1 Introduction of Electricity Price Forecasting . . . 84
8.2 Proposed Ensemble Method . . . 85
8.3 Experiment setup . . . 86
8.4 Results and Discussion . . . 87
8.4.1 Performance comparison for short-term electricity price forecasting . . . 87
8.4.2 Computation time comparison . . . 88
8.5 Summary . . . 89

9 Fusion of Multiple Indicators with Ensemble Incremental Learning Techniques for Stock Price Forecasting 91
9.1 Literature Review of Stock Price Forecasting . . . 91
9.1.1 Indicators of Stock Market . . . 92
9.2 Proposed Fusion Incremental Learning Method . . . 93
9.3 Experiment setup . . . 95
9.4 Results and Discussion . . . 95
9.4.1 Performance Comparison with Benchmark Models . . . 95
9.4.2 Computation time comparison . . . 97
9.5 Summary . . . 98

10 Summary of Part II 99

III Surrogate for Chemical Plant Process Flow Modelling 102

11 Machine learning approach for constructing surrogates of a biodiesel plant flow sheet model 103
11.1 Introduction . . . 103
11.2 Experiment Setup . . . 106
11.2.1 Data collection . . . 106

11.2.2 Data normalization 106
11.2.3 Performance estimation 107
11.3 Results and Comparison 108
11.3.1 Performance comparison using R^2 values for training data 108
11.3.2 Performance comparison using RMSD values for testing data 109
11.3.3 Performance comparison using residual plots 111
11.3.4 Computation time comparison 112
11.4 Summary for Surrogate Models 113

12 Conclusions and Future Work **116**
12.1 Conclusions 116
12.2 Future Work 119

List of Publications **121**

Bibliography **123**

List of Figures

2.1 Schematic of a Neural Network Model . . . 12
2.2 Schematic Diagram of an RVFL Network. The dashed arrows show the direct connections between the input neurons and the output neurons, whose weights are denoted as w_{io}. . . . 13
2.3 Flowchart of a Deep Belief Network (DBN) . . . 18
2.4 Schematic Diagram of a Restricted Boltzmann Machine (RBM) . . . 18
2.5 A typical framework of ensemble methods. . . . 20

3.1 Time plot of load demand data with a time window of two weeks . . . 28
3.2 Autocorrelation function for electricity load demand data in TAS . . . 29
3.3 Schematic Diagram of the Proposed Oblique Random Forest . . . 32
3.4 Nemenyi testing for generic TS forecasting. The critical distance is 1.8. . . . 34
3.5 Nemenyi testing for electricity load demand forecasting based on RMSE. The critical distance is 2.0. . . . 37
3.6 Nemenyi testing for electricity load demand forecasting based on MAPE. The critical distance is 2.0. . . . 38

4.1 Schematic Diagram of the proposed Ensemble Deep Learning Network . . . 40
4.2 Schematic Diagram of the Proposed EMD based Deep Learning Approach . . . 44
4.3 Example of the obtained IMF components after EMD with a time window of one month. . . . 45
4.4 Nemenyi testing results for half-an-hour ahead load forecasting based on RMSE (left) and MAPE (right). The critical distance is 3.0. . . . 47
4.5 Nemenyi testing results for one-day ahead load forecasting based on RMSE (left) and MAPE (right). The critical distance is 3.0. . . . 48

5.1 Schematic Diagram of the Proposed DWT-EMD based Incremental RVFL Network . . . 55
5.2 Nemenyi test for electric load forecasting based on RMSE. The critical distance is 3.1. . . . 61
5.3 Nemenyi test for electric load forecasting based on MAPE. The critical distance is 3.1. . . . 62
5.4 Comparison of predicted values with actual values for the proposed method. The y-axis represents for electric load power (MW), and each point on x-axis represents for half hour. . . . 63
5.5 Comparison of predicted values with actual values for RVFL. The y-axis represents for electric load power (MW), and each point on x-axis represents for half hour. . . . 63
5.6 Computation time of learning models for electric load forecasting . . . 64

6.1 Schematic Diagram of the Proposed CEEMDAN-KRR-RVFL model 69
6.2 A Fraction of Wind Power Generated in an ELIA wind farm, red segments denote power ramps. 70
6.3 Nemenyi test for wind power forecasting based on NRMSE. The critical distance is 1.7. 75
6.4 Nemenyi test for wind power ramp rate forecasting based on NRMSE. The critical distance is 1.7. 75

7.1 Nemenyi test for electric load forecasting based on RMSE. The critical distance is 3.1. 81
7.2 Nemenyi test for electric load forecasting based on MAPE. The critical distance is 3.1. 82

8.1 Schematic Diagram of the Proposed EMD-KRR-SVR approach 86
8.2 Nemenyi test for electricity price forecasting based on RMSE. The critical distance is 2.6. 89
8.3 Computation time of learning models for electricity price forecasting in Tasmania (TAS) 89

9.1 Schematic Diagram of the Proposed DWT-EMD based Incremental RVFL-SVR Model 94
9.2 Nemenyi test for stock price forecasting based on RMSE. The critical distance is 2.7. 97
9.3 Nemenyi test for stock price forecasting based on MAPE. The critical distance is 2.7. 97
9.4 Computation time of learning models for stock price forecasting 98

10.1 Nemenyi test for stock price forecasting based on RMSE. The critical distance is 3.1. 100
10.2 Nemenyi test for stock price forecasting based on MAPE. The critical distance is 3.1. 101

11.1 Framework of EIP modelling based on Industry 4.0. Adopted from [1]. 104
11.2 Model Development Suite work flow. Adopted from [2]. 106
11.3 Plot of R^2 for the surrogate models 108
11.4 Plot of adjusted R^2 for the surrogate models 109
11.5 Nemenyi testing results for surrogate models based on RMSD. The models within a vertical line whose length is less than or equal to a critical distance have statistically the similar performance. 110
11.6 Plots of RMSD values produce by polynomial fitting, HDMR model and machine learning methods for heat duty (MW) of reactor 10D01 with respect to all 11 inputs. 111
11.7 Plot of residuals against molar flow of tripalmitin oil for heat duty of reactor 10D01 produced for 11 inputs. 112
11.8 Plot of residuals against molar flow of tripalmitin oil for heat duty of heater 10E03 produced for 11 inputs 113
11.9 Plot of residuals against molar flow of tripalmitin oil for heat duty of reactor 10D01 produced for 1 inputs. 114
11.10 Training and evaluation time of learning models for constructing 11-dimensional surrogates of heat duties of reactor 10D01. 114

List of Tables

2.1 Random Forest 17

3.1 Summary of the eight generic TS datasets 33
3.2 Forecasting results for eight generic time series datasets 34
3.3 Forecasting results for one day ahead electricity load demand forecasting . . . 36
3.4 Average computation time of electric load forecasting models 36

4.1 Prediction results for Mackey-Glass Time Series 41
4.2 Prediction results for load demand of New South Wales 41
4.3 Prediction results for load demand of South Australia 42
4.4 Prediction results for load demand of Tasmania 42
4.5 Prediction results for 2D planes dataset 42
4.6 Prediction results for Friedman Artificial Domain dataset 42
4.7 Prediction results for California Housing 43
4.8 Prediction results for half-an-hour ahead load forecasting 47
4.9 Average computation time of electric load forecasting models 48
4.10 Prediction results for one day ahead load forecasting 49
4.11 Forecasting results for monthly electric load demand in Northeastern China as used in [3, 4] 50
4.12 Forecasting results for electric load demand in New South Wales in 2007 [5] . . 50
4.13 Comparative Results with PSF-NNs [6] 51

5.1 Summary of AEMO load datasets 56
5.2 Performance comparison between RVFL variants with and without direct links 58
5.3 Performance comparison between incremental and non-incremental learning. I stands for incremental, and N stands for non-incremental. 59
5.4 Prediction results for one-day-ahead electric load forecasting 61
5.5 Prediction results for different seasons using the load data from NSW of the year 2015 64
5.6 Forecasting results for comparative experiment one. The results of additive model, ANN and Hybrid model are obtained from [7]. 65

6.1 Contingency Table to Evaluate the Performance of the Binary Class Classification 72
6.2 Selected NRMSE of the Wind Power Forecasting over 12 hour Forecasting Horizon. 74
6.3 Selected NRMSE of the Wind Power Ramp Rate Forecasting over 12 hour Forecasting Horizon. 76
6.4 Performance Measures of the Power Ramp Classification in the next 12 hours. . 77
6.5 Average Computation Time (sec) over 5 Datasets. 77

7.1 Summary of AEMO load datasets 80
7.2 Prediction results for one-day-ahead electric load forecasting 81

8.1 Prediction results for half-an-hour ahead electricity price forecasting ($/MWh) . 88

9.1 Technical details of the selected stock market indicators [8] 93
9.2 Prediction results for stock market price forecasting 96

10.1 Prediction results for stock market price forecasting 100

11.1 Definitions and domain bounds of input variables 107
11.2 Definitions of Output variables 107
11.3 Performance evaluation of surrogate models with RMSD 109

Abbreviations

ANN	**A**rtificial **N**eural **N**etwork
RVFL	**R**andom **V**ector **F**unctional **L**ink
SVM	**S**upport **V**ector **M**achine
SVR	**S**upport **V**ector **R**egression
KRR	**K**ernel **R**idge **R**egression
RF	**R**andom **F**orest
DBN	**D**eep **B**elief **N**etwork
RBM	**R**estricted **B**oltzmann **M**achine
RNN	**R**ecurrent **N**eural **N**etwork
CNN	**C**onvolutional **N**eural **N**etwork
EMD	**E**mpirical **M**ode **D**ecomposition
IMF	**I**ntrinsic **M**ode **F**unction
DWT	**D**istrict **W**avelet **T**ransform
ACF	**A**uto**c**orrelation **F**unction
SLFN	**S**ingle hidden **L**ayer **F**eedforward neural **N**etwork
LSTM	**L**ong **S**hort **T**erm **M**emory
RVFL	**R**andom **V**ector **F**unctional **L**ink
MAPE	**M**ean **A**bsolute **P**ercentage **E**rror
RMSE	**R**oot **M**ean **S**quare **E**rror
ARMA	**A**uto **R**egressive **M**oving **A**verage
ARIMA	**A**uto **R**egressive **I**ntegrated **M**oving **A**verage

Chapter 1

Introduction

1.1 Motivations

In recent years, time series forecasting has obtained significant academical and industrial interest with its significance in various application fields, including power system related applications (electric load, wind power and solar irradiance forecasting, etc.), as well as financial market related applications (stock price, exchange rate and electricity price forecasting, etc.).

In modern society, the electricity power market is fast developing and highly competitive. As a result, short-term electric load forecasting has become important in power systems. Improving the accuracy and efficiency of load demand forecasting can help power companies develop reasonable grid construction planning which will lead to the improvement of the economic and social benefits of the systems. Moreover, forecasting results with high accuracy can help predict the potential faults in the power systems so that provide a reliable safety basis for the grid operation. In other words, the goal of load demand forecasting is to provide reliable power supply while making the operating costs optimized. However, electric load forecasting is challenging. There are many influencing issues causing the data to be highly nonlinear (e.g. social activities, climate factors, etc.). [9, 10].

Wind is clean and renewable, which may be treated as a potential energy source. However, the power generated by wind farms fluctuates because of the intermittent nature of the wind [11, 12]. The fluctuations caused by wind ramp is normally solved by the battery storage systems or conventional fossil power generator. But the effectiveness of the compensation will be low if there are large fluctuations. If we want to integrate the wind power into the power grid, accurate wind power ramp forecasting is important to optimize the planning and scheduling of power systems [13, 14]. It is also helpful to protect the power transmission and generation system from a sudden rise and drop in power supply [15].

Except for the applications in power systems, in modern financial markets and industrial fields, big data mining, time series analysis and short term forecasting are also important for companies to optimize their plans and strategies in order to keep themselves competitive. Stock price forecasting predicts the future stock market price by analyzing time series (TS) signal and extracting meaningful features and characteristics [16]. Among the challenging tasks in the field of financial time series forecasting, stock price forecasting is regarded as one of the most difficult, due to the highly non-linear and non-stationary patterns of stock price TS caused by numerous influence factors, such as economy, government, enterprise and investors [17, 18].

Based on the forecasting horizon, time series forecasting problems can be categorized into four types: long-term (years ahead), medium-term (months to a year ahead), short-term (a day to weeks ahead) and very short-term (minutes to hours ahead) [19]. In this thesis, I mainly focus on short term time series forecasting.

1.2 Learning Algorithms for Time Series Forecasting

Time series (TS) analysis is a hot research field, which aims to extract meaningful statistics and other characteristics by analyzing the data itself. Methods of time series analysis can be divided into two categories: univariate and multivariate. For example, Raza has developed exponentially weighted moving average (EWMA) based shift-detection methods for detecting covariate shifts in non-stationary environments [20]. In the testing stage, Kolmogorov-Smirnov statistical hypothesis test is applied for univariate TS, and the Hotelling T-Squared multivariate statistical hypothesis test is used in the case of multivariate TS. Moreover, many TS datasets have cyclical or seasonal characteristic, which influences TS analysis. Many models have been designed to deal with the cyclic characteristics. For example, Gharehbaghi has developed a pattern recognition framework for detecting dynamic changes on cyclic time series, which combines the discriminant analysis and k-means clustering method [21].

The regression of time series is similar to other types of regression, in which we use past observations as input features to forecast the future value. But there still exists two important differences [22]. Firstly, time variables themselves are often useful in predicting the behavior of a time series. Many useful information can be extracted from time series data, such as trend over time, seasonality, business cycles, etc. Secondly, the order of the data points is important. That is because, unlike cross-sectional data, the ordering is not arbitrary but represents the order in which the data were collected. Therefore, we need to consider these differences when we construct regression models for time series data.

Since the 1940s, various statistical based linear time series forecasting approaches have been published. For linear models, generally their objective is to use time series analysis for extrapolating the future values. For example, in [23], the energy consumption and economic growth for Israel were examined by trend analysis. Moreover, the most successful methods are based on Holt-Winters exponential smoothing [24] and Autoregressive Integrated Moving Average (ARIMA) [25], as well as Linear Regression [26].

In recent several decades, with the fast development of artificial intelligence and machine learning, numerous nonlinear forecasting models have been proposed, which include Artificial Neural Network (ANN), fuzzy comprehensive evaluation method, Support Vector Machine (SVM) [27] and so on. The advantages of nonlinear models are demonstrated by many works in the literature. Especially, in a survey paper [28], the authors summarized various works based on both linear and nonlinear models for electric load forecasting.

Kernel machines has become very popular since Support Vector Machine (SVM) being introduced in 1995 [27]. SVM makes use of "kernel tricks" to nonlinearly map the data into a higher dimensional space, in which the transformed data can be linearly separated. SVM has the advantage of giving a single optimized functional solution compared to ANN which is frequently trapped in a local minimum. Many SVM based electricity price forecasting algorithms exist in the literature. For example, in [29], a hybrid model called SVR-ARIMA that combines both SVR and ARIMA models was proposed for short term EPF problems. Besides for SVM, possibly the most elementary algorithm that can be kernelized is ridge regression. Sharing the similar idea of SVR, Kernel Ridge Regression (KRR) employs the kernel trick into Ridge Regression. However, KRR can be trained using closed-form solutions, which is typically faster than SVR for medium-sized datasets [30, 31].

However, there still exists some drawbacks in the nonlinear models. For example, artificial neural network is often trapped in a local minimum which misleads the forecasting results. To achieve better performance, deep learning gained traction in 2006 after the publication by Geoffrey Hinton et al, which introduces Deep Belief Network (DBN) [32]. Deep learning has been widely used in many fields, such as image classification, speech recognition, handwriting recognition and so on [33]. There are also many deep learning papers published for time series forecasting [34, 35]. For example, Busseti conducted simulations to compare the performance for electricity load demand forecasting between deep learning methods and traditional shallow neural networks, which successfully showed the advantages of deep learning architectures [35].

Many papers have been published and show the success of deep learning [36–41]. Takashi Kuremoto has proposed a time series forecasting model using DBN with multiple restricted Boltzmann machines [42]. The Color And Thermal Stereo (CATS) benchmark data has been used in the form of 5 blocks with 20 missing and 980 known in each block. The model was then optimized by the particle swarm optimization (PSO) algorithm. This work has shown

DBN's superiority over conventional MLP neural network model and statistical model ARIMA. Busseti *et al.* also conducted simulations to compare deep learning methods with traditional shallow neural networks [35]. The work successfully showed the advantages of deep learning architectures to the problems of electricity load demand forecasting.

Fast algorithms are commonly used with ensembles, such as decision trees. The ensemble of decision trees is usually called "decision forests", among which Random Forest (RF) [43] performs best in the literature [44]. RF increases the variance of base learning models by combining the concept of bagging and random subspaces [45], thereby improving the performance of this learning model. RF has been widely applied for classification problems in numerous research field, including micro-arrays [46], image segmentation [47] and feature selection [48]. Manuel Fernández-Delgado et al. compared 179 learning models from 17 families using 121 classification datasets, among which RF achieved the best performance [44]. In [49], the authors showed that oblique random forest may lead to a better performance and efficiency than its conventional axis-parallel counterpart. Recently, applications of RF for solving regression and time series forecasting problems have gained some interest. For example, Li et al. employed RF for lake water level time series forecasting and compare with some statistical methods [50]. In [51], very short-range sky condition forecasting methods were developed based on RF and neural networks. However, based on the authors' best knowledge, the investigation of oblique RF for time series forecasting problems (including the electricity load demand forecasting) is seldom mentioned in the literature.

Another example of fast training model is a randomized version of neural network, which was reported in [52, 53], named as Random Vector Functional Link (RVFL) network. RVFL has random weight assignment and functional link between input and output layers, which improves the efficiency of neural network training by randomly generating the weights between the input layer and hidden layer [53, 54]. Another independently developed method, single hidden layer neural network with random weights (RWSLFN), was presented in [55], which is different from RVFL by excluding the functional link. However, some research works have proved that the functional link can significantly benefit the performance of RVFL, especially for time series forecasting [54, 56].

1.3 Ensemble Methods

Ensemble methods, which work on a higher level to improve the performance of "unstable" predictors [57] such as decision tree and neural networks, have been successfully employed for solving pattern classification, regression, time series forecasting and fault prediction problems. Chatterjee developed an ensemble method for reliability forecasting of a mining machine [58].

This algorithm employed least square SVM (LS-SVM) which is optimized by Genetic Algorithm (GA). Turbocharger benchmark data sets were used to evaluate the performance. The outcome successfully showed the advantages of this ensemble method in fault prediction and reliability forecasting applications. Three possible reasons for applying ensemble methods are concluded by Dietterich: statistical, computational and representational [59].

There are several survey papers for ensemble regression methods [60, 61] in the literature. Among the various ensemble methods [34, 62–65], Divide and Conquer [66] is a common ensemble strategy for time series forecasting. Wavelet transform, as a well-known TS decomposition algorithm, works by decomposing the TS signal in frequency domain. For example, Benaouda *et al.* [67] employed wavelet based nonlinear multistate decomposition model for electric load forecasting. Empirical Mode Decomposition (EMD) [68] is another decomposition method suitable for time series forecasting, which is a part of Hilbert-Huang Transform (HHT). The difference between wavelet transform and EMD is that EMD decompose TS signal in time domain. A comparative study on different variations of EMD for wind speed forecasting was reported in [69].

Numerous forecasting methods based on various ensemble methods for TS forecasting have been published in the literature. For example, in [70], an EMD-LS-SVR method was proposed for wind speed forecasting, which outperformed LS-SVR and EMD-Regular Least Square methods. In [71], a wind power forecasting method was proposed based on wavelet-fuzzy predictive adaptive resonance theory (ARTMAP). Moreover, in [72], a solar irradiance forecasting model was implemented by a combination of ARIMA and neural networks. Further, an ensemble deep learning method for regression and time series forecasting was introduced in [34], which was composed of an ensemble of deep belief networks (DBN) and an SVR. However, to the authors' best knowledge, there is little research work focus on decision tree ensembles, especially for oblique decision trees, for time series forecasting.

1.4 Incremental Learning

Incremental learning, or online learning, is a machine learning paradigm where the machine learning model is updated according to the new examples whenever they emerge [73]. Therefore, incremental learning is different from traditional machine learning in the requirement of training dataset. Incremental learning does not totally depend on a sufficient training set in the training phase, but always updates its model as new training examples appear over time. In fact, incremental learning is part of the natural learning and quite common in reality. For example, in [74], a recurrent neural network was constructed for grammar learning, and the author found that "the network fails to learn the task when the entire data set is presented all at once, but succeeds when the data are presented incrementally". For electric load forecasting, electricity loads

can be seen as a stream of incoming data, thereby it is necessary to focus on adaptive methods that are able to learn incrementally. Several incremental learning methods have been proposed for electric load forecasting in the literature. For example, Gabriela Grmanová *et al.* proposed an incremental heterogeneous ensemble model for electric load forecasting [75]. Moreover, in [76], the authors presented an incremental electric load forecasting model based on SVR.

1.5 Outline of the Thesis

This thesis mainly focuses on time series forecasting in power systems and financial markets. The primary objective is to reduce the probability of faults occurring with the help of accurate load forecasting methods. Deep learning and ensemble methods have been utilized to construct new models with better performance.

Chapter 2 introduces data preprocessing, including data cleaning, normalization and feature extraction. The literature review of existing time series forecasting models is also explained in this chapter, which starts from linear models, especially for ARIMA, followed by ANN, RVFL, SVR, KRR and RF, finally goes to deep learning.

The thesis consists of three main parts: ensemble learning with applications in power systems, ensemble learning with applications in financial markets and surrogate for chemical plant process flow modelling.

For the first part of this thesis, ensemble methods for two kinds of power system related time series forecasting are investigated: electric load forecasting and wind power forecasting. Chapter 3 investigates one decision tree ensemble method, oblique random forests, in the context of time series forecasting [77]. Moreover, Chapter 4 describes two ensemble deep learning methods: the first one consists of deep belief network (DBN) and SVR, which is named EDBN [34]; while another one is based on EMD and DBN. For EDBN, the SVR aggregates the outputs of different DBNs [16]. The advantages of the proposed method are demonstrated on real world electric load datasets compared with several benchmark learning algorithms. Furthermore, in Chapter 5, an ensemble incremental learning method based on DWT, EMD and RVFL is presented, which has advantages on both accuracy and efficiency [78]. In Chapter 6, ensemble learning for wind power forecasting is studies [79]. Chapter 7 summarizes the first part of this thesis and demonstrates the advantages and disadvantages of the above ensemble learning models by a overall comparison for electric load forecasting.

In the second part of this thesis, we focus on ensemble learning for financial markets based time series forecasting. Chapter 8 presents an ensemble kernel machine for electricity price forecasting [80]. In Chapter 9, a fusion incremental learning approach is presented for short-term stock price forecasting, which is comprised of Discrete Wavelet Transform (DWT), Empirical

Mode Decomposition (EMD), Random Vector Functional Link (RVFL) network and Support Vector Regression (SVR). Similar with Chapter 7, in Chapter 10, the second part of the thesis is summarized by an overall comparison for stock price forecasting.

For the third part, in Chapter 11, another topic is covered, which focuses on constructing surrogate for chemical plant flow sheet model with numerous machine learning models. Based on the definition of regression, it includes both forecasting and function approximation. In this chapter, the surrogate models are trained as regression models using a number of input variables without considering time sequence dependence. For the future research works, time series forecasting/classification models will also be employed to help improve the surrogate models.

Finally, in Chapter 12, the entire conclusion of all these techniques is done along with the importance of various techniques and the further scope of research.

Chapter 2

Theoretical Background of Forecasting Models

2.1 Data Preprocessing

The performance of machine learning algorithms on a given time series signal can be affected by many factors, amongst which, the quality of training data and input feature is the most important [81]. Data are collected by some data gathering methods, which are often loosely controlled. Therefore, many kinds of irrelevant and redundant information exist in the original data set, such as out-of-range values, impossible data combinations and missing values. Then it is difficult for machine learning algorithms to analyze data that has not been carefully screened, and finally produce misleading results. Data preprocessing includes data cleaning, normalization, feature extraction and selection, etc.

2.1.1 Data Cleaning

Time series data are affected by meteorological factors, which create noise and often infect the performance of machine learning algorithms. Some main types of noise include outliers, missing values and white noise. Different methods should be applied to remove different kinds of noise.

A rolling window with a fixed width can be used to roll along the TS signal for outlier detection. One segment of TS $\mathbf{Y}_w$ can thus been formed by each window. Then the median absolute deviation (MAD) can be calculated and used to detect outliers:

$$MAD = median_i(|Y_i - median_j(Y_j)|) \tag{2.1}$$

Incomplete data is an common problem for the real world datasets. The missing data points are often recorded as 'NA', 'NaN', '?', etc. Mean values computed from existing neighbouring data commonly applicable values or regression results can be used to replace the missing data points [82].

The white noise often appears in TS signal, which has a constant power spectral density. White noise can be uniformly distributed or Gaussian distributed depending on the probability distribution. Many techniques can be used to deal with the white noise, such as Auto-regressive Moving Average (ARMA) and Empirical Mode Decomposition (EMD) [83, 84]. The methods will be discussed later in this chapter.

2.1.2 Normalization

Normalization can scale down the input data to avoid a large range of data distribution. This is important for neural network based algorithms, because it is beneficial for approaching to global minima at error surface efficiently. The two most common methods for normalization are:

1. min-max normalization:

$$v' = \frac{v - min_A}{max_A - min_A}(newmax_A - newmin_A) + newmin_A \tag{2.2}$$

2. z-score normalization:

$$v' = \frac{v - mean_A}{std_A} \tag{2.3}$$

where A is the time series dataset, max_A and min_A are the maximum and minimum values of the dataset A, respectively. Similarly, $newmax_A$ and $newmin_A$ are the maximum and minimum values of the scaled dataset, respectively. v is the original value and v' is the scaled value.

2.1.3 Feature Extraction

Feature extraction can also be called feature construction, which is a process to construct new feature from the basic feature set. More concise and accurate predictors or classifiers may be created under the help of the newly generated features. Moreover, it is helpful to have a better understanding of the learned concept by finding meaningful features.

Many techniques can be used for feature extraction, such as extraction of local features, Principal Component Analysis (PCA) [85], Multidimensional Scaling (MDS) [86], etc. In my research, the Empirical Mode Decomposition has been applied to decompose TS signal, remove the white noise and finally obtain the better forecasting performance.

2.2 Forecasting Methods

In the literature, various kinds of learning models have been published for time series modelling and forecasting which include both linear and nonlinear models.

2.2.1 Linear Methods

Linear TS methods can analyze the covariance relationship within TS signal. By combining the concept of autoregressive and moving average, we can get one of the most successful linear prediction model: autoregressive moving average (ARIMA) model.

For multiple regression models, the forecasting model is constructed by a linear combination of predictors. However, in an autoregression model, a linear combination of past values are used for prediction. The autoregressive model of order p is defined as:

$$y_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + e_t \tag{2.4}$$

where φ_1 to φ_p are the weights for corresponding lag values, e_t represents white noise, and c is a constant. The model can be marked as an AR(p) model.

Time series data collected often contain some form of random variation, which shall be removed to cancel the influence. "Smoothing" is an frequently used technique in industry to reveal more clearly the underlying trend, seasonal and cyclic components. Moving average is one of the "smoothing" methods to summarize the past data by computing the mean of successive smaller sets of numbers of past data. The time series $\mathbf{X}_t$ has a $q^t h$ order moving average representation, denoted by MA(q), if the following condition is satisfied:

$$\mathbf{X}_t = \sum_{j=0}^{q} (\varphi_j \varepsilon_t) - j \tag{2.5}$$

where ε_t is a stochastic process with zero mean and finite variance.

Therefore, by combining both of the AR(p) model and MA(q) model, we can obtain the ARMA model, which should be denoted as ARMA(p,q). This model was first introduced in the 1951 thesis of Peter Whittle [87–89]. He used Laurent series and Fourier analysis, along with statistical inference to describe the ARMA model. Then in 1971, George E. P. Box and Jenkins developed an iterative method, called Box-Jenkins method, for choosing and estimating the ARMA models [22]. Note that this is only useful for low-order polynomials, which means degree three or less [90].

The autoregressive integrated moving average (ARIMA) model is a generalization of an ARMA model. Non-seasonal ARIMA models are normally denoted as ARIMA(p, d, q) where p is the order of the AR model, d is the degree of differencing, and q is the order of the MA model. That is to say, the ARIMA forecasting equation for a stationary time series is a linear equation in which the predictors consist of lags of the dependent variable and/or lags of the forecast errors. The general forecasting equation is:

$$\hat{y}_t = \mu + \varphi_1 y_{t-1} + ... + \varphi_p y_{t-p} - \vartheta_1 e_{t-1} - ... - \vartheta_q e_{t-q} \tag{2.6}$$

where φ_1 to φ_p are the weights for the lagged terms of the dependent variable, while ϑ_1 to ϑ_p represent the lagged terms of the forecast errors.

Similarly, seasonal ARIMA models are usually denoted ARIMA$(p, d, q)(P, D, Q)_m$, where m is the length of the seasonal cycle, and the uppercase P, D, Q stand for the AR, differencing, and MA terms for the seasonal part, respectively.

2.2.2 Artificial Neural Network

ANN is a learning model inspired by the human brain, especially the central nervous system [158]. The simplest model of ANN is called single-hidden layer feedforward neural network (SLFN). Figure 2.1 is an illustration of a three-layer SLFN. There are three fundamental layers in an SLFN: an input layer with the same number of neurons as the dimension of input features; a hidden layer comprised of neurons with nonlinear activation function; and an output layer which aggregates the outputs from the hidden layer neurons. The output from SLFN is:

$$y = g(\sum_{j=1}^{h} w_{jo} v_j + b_{ho}), \quad j = 1, \ldots, m \tag{2.7}$$

$$v_j = f(\sum_{i=1}^{n} w_{ij} x_i + b_{ih}), \quad i = 1, \ldots, n \tag{2.8}$$

where x_i is the input to the neuron i_i; $f()$ and $g()$ are nonlinear activation functions; v_j is the output of hidden layer neuron h_j; y is the output of this SLFN; n and m are the number of input features and the number of the hidden layer neurons, respectively; w_{ij} is the weight of the connection between the input variable i_i and the neuron h_j of the hidden layer; w_{jo} is the weight of the connection between the hidden layer neuron h_j and the output; b_{ih} and b_{ho} are the biases.

To train an SLFN, random values are assigned to the weights, then the weights are tuned by a certain method such as back-propagation (BP) [91] or using a closed form solution [54, 92].

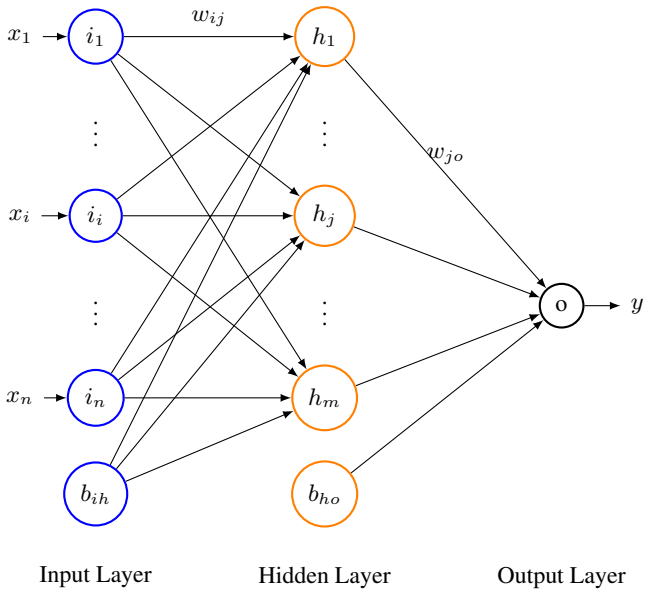


FIGURE 2.1: Schematic of a Neural Network Model

It has been prove that a SLFN with sufficient hidden nodes is a universal approximator with adaptive learning ability. However, back propagation algorithm is time consuming and often trapped in local minima.

2.2.3 Random Vector Functional Link Neural Network

RVFL network is a randomized version of the SLFN, which has direct connections between input and output neurons (functional link), and uses fixed random weights and closed form least square estimation instead of the BP to tune the weights [52, 53]. Figure 2.2 is the schematic diagram of an RVFL network.

It is worth noting that all hidden layer weights w_{ij} in a RVFL are generated with uniformly distributed random values within the interval $[-S,+S]$, where S is a scale factor to be determined during the parameter tuning stage [56]. Therefore, the output v_j from the hidden neuron h_j can be calculated based on the activation function. Here logistic sigmoid function is used as an example:

$$v_j = \text{logsig}(\sum_{i=1}^{n} w_{ij}x_i + b_{ih}), \quad i = 1, \ldots, n \tag{2.9}$$

where x_i is the training data.

The output layer weight vector $\boldsymbol{w}_o$, which includes both w_{io} and w_{jo}, needs to be determined by a certain optimization method. Due to the efficiency of closed-form solution, RVFL employs

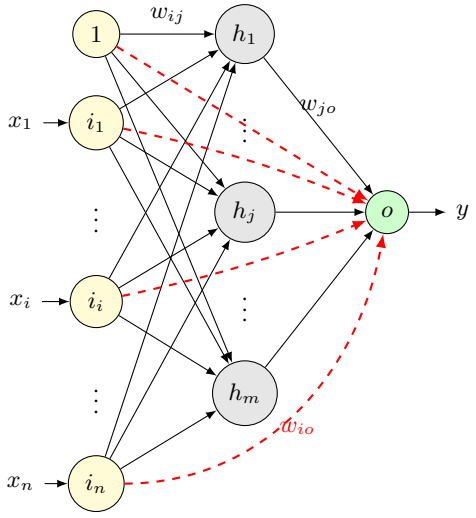


FIGURE 2.2: Schematic Diagram of an RVFL Network. The dashed arrows show the direct connections between the input neurons and the output neurons, whose weights are denoted as w_{io}.

least square estimation to calculate the output layer weights:

$$\boldsymbol{w}_o = (\boldsymbol{v}^T\boldsymbol{v})^{-1}\boldsymbol{v}^T\boldsymbol{y} \tag{2.10}$$

where $\boldsymbol{y}$ is the training target vector.

The predicted values can thus be calculated by applying the obtained $\boldsymbol{w}_o$ and $\boldsymbol{w}_{ij}$ to testing data:

$$\hat{\boldsymbol{y}}_{\boldsymbol{s}} = \boldsymbol{w}_o \cdot \text{logsig}(\boldsymbol{w}_{ij} \cdot \boldsymbol{x}_s) \tag{2.11}$$

where $\hat{\boldsymbol{y}}_{\boldsymbol{s}}$ is the predicted testing values and $\boldsymbol{x}_s$ is the testing data [93].

Generally speaking, RVFL network is a universal approximator with good efficiency because of the randomly generated weights between input and hidden layers and the close form solution for parameter computation. However, the memory requirement increases significantly as the number of hidden neurons increasing.

2.2.4 Support Vector Regression

The Support Vector Machine (SVM) is a machine learning algorithm proposed by Cortes and Vapnik [27] based on statistical learning theory. Structural risk minimization is the basic concept of this method. A version of SVM for regression was proposed in [94]. Support vector regression has been widely applied in time series forecasting problems [10]. There are two versions of SVR: ϵ-SVR and ν-SVR.

ϵ-Support Vector Regression

Suppose a time series data set is given as follows

$$D = \{(\boldsymbol{x}_i, y_i)\}, 1 \leq i \leq N \tag{2.12}$$

where $\mathbf{x}_i$ is the input vector at time i with m elements and y_i is the corresponding output data. The regression function can be defined as

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}) + b \tag{2.13}$$

where $\boldsymbol{w}$ is the weight vector, b is the bias, and $\phi(\boldsymbol{x})$ maps the input vector $\boldsymbol{x}$ to a higher dimensional feature space. $\boldsymbol{w}$ and b can be obtained by solving the following optimization problem:

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\xi}^*} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \tag{2.14}$$

Subject to:

$$\begin{aligned} & y_i - \boldsymbol{w}^T\phi(\boldsymbol{x}_i) - b \leq \epsilon + \xi_i \\ & \boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b - y_i \leq \epsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0 \end{aligned} \tag{2.15}$$

where C is a predefined positive trade-off parameter between model simplicity and generalization ability, ξ_i and ξ_i^* are the slack variables measuring the cost of the errors.

For nonlinear input data set, kernel functions can be used to map from original space onto a higher dimensional feature space in which a linear regression model can be built. The dual problem is

$$\min_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)^T Q(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \epsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{N} y_i(\alpha_i - \alpha_i^*) \tag{2.16}$$

Subject to:

$$\begin{aligned} & \boldsymbol{e}^T(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) = 0, \\ & 0 \leq \alpha_i, \alpha_i^* \leq C \end{aligned} \tag{2.17}$$

where $\boldsymbol{e} = [1, \ldots, 1]^T$ is the vector of all ones, Q is an l by l positive semidefinite matrix, K is the kernel function, $Q_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j) \equiv \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)$.

Thus, the final SVR function is obtained as

$$y_i = f(\boldsymbol{x}_i) = \sum_{i=1}^{N}(\alpha_i^* - \alpha_i)K(\boldsymbol{x}_i, \boldsymbol{x}) + b \tag{2.18}$$

where α_i and α_i^* are the Lagrange multipliers. The most frequently used kernel function is the Gaussian radial function (RBF) with a width of σ

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = exp(-\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2/(2\sigma^2)) \tag{2.19}$$

ν-Support Vector Regression

ν-SVR uses a parameter $\nu \in (0, 1]$ to control the number of support vectors [95]. It has been proven that ν is an upper bound on the fraction of training errors and a lower bound of the fraction of support vectors [96].

Similar with ϵ-SVR, with (C, ν) as parameters, ν-SVR solves

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\xi}^*,\epsilon} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C(\nu\epsilon + \frac{1}{N}\sum_{i=1}^{N}(\xi_i + \xi_i^*)) \tag{2.20}$$

Subject to:

$$\begin{aligned} &y_i - \boldsymbol{w}^T\phi(\boldsymbol{x}_i) - b \leq \epsilon + \xi_i \\ &\boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b - y_i \leq \epsilon + \xi_i^* \\ &\xi_i, \xi_i^* \geq 0, \epsilon \geq 0. \end{aligned} \tag{2.21}$$

The dual problem is

$$\min_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)^T Q(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \boldsymbol{y}^T(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) \tag{2.22}$$

Subject to:

$$\begin{aligned} &\boldsymbol{e}^T(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) = 0, \boldsymbol{e}^T(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) \leq C\nu, \\ &0 \leq \alpha_i, \alpha_i^* \leq C/N \end{aligned} \tag{2.23}$$

The final approximate function is

$$y_i = f(\boldsymbol{x}_i) = \sum_{i=1}^{N}(\alpha_i^* - \alpha_i)K(\boldsymbol{x}_i, \boldsymbol{x}) + b \tag{2.24}$$

2.2.5 Kernel Ridge Regression

Ridge Regression is a linear model which addresses ordinary least squares by imposing a penalty on the size of coefficients (l2-norm regularization) [97]. The ridge coefficients minimize a penalized residual sum of squares which is shown as follows:

$$\min_{w} ||Xw - y||_2^2 + \alpha||w||_2^2 \tag{2.25}$$

where α is a complexity parameter that controls the amount of shrinkage. The coefficients are more robust to collinearity as α becomes lager.

Kernel ridge regression (KRR) combines Ridge Regression with the kernel trick [30, 31]. Thus it constructs a linear model in the space induced by the kernel we used for the data. The form of the model learned by KRR is similar with SVR, except for the different loss functions. KRR uses squared error loss instead of ε-insensitive loss which is applied in SVR. Moreover, KRR can be trained in closed-form and is typically faster for medium-sized datasets.

2.2.6 Random Forest

Random forest, or random decision forest [47], proposed by Breiman [43], is an ensemble learning method for both classification and regression problems. Random forest combines bagging and random subspace method (RSM) by conducting random feature subspace at each node of the classification and regression tree (CART) [45]. Bagging (bootstrap aggregating), developed by Breiman, is a widely used ensemble method [98]. In bagging ensemble method, one trains each weak learning machine on bootstrap samples of the original training samples and aggregates the outputs. RSM is a combining method which trains the learning machines on randomly chosen subspaces of the original input space, and combines the outputs by a majority vote or median [47]. More specifically, at each node of the decision tree in random forest, m features from totally n input features are randomly selected. Then, according to an impurity criterion, one of these features is used to perform a partition along the feature axis [45]. The algorithm of RF is presented in Table 2.1 [49, 99].

In a decision tree, in order to make the further separations in the children nodes being easier, the data in each non-leaf node is separated by a hyperplane, which does not need to be a good classifier at this stage [100]. As we mentioned above, most decision tree induction algorithms employ certain impurity measures, such as the GINI index and the twoing rule, to find a split with the lowest impurity score . The impurity criterion gives different impurity scores according to whether the distributions being near uniform or not by measuring the skewness of the distribution of different classes in the set of samples reaching the node.

TABLE 2.1: Random Forest

Random Forest Algorithm:
Given: X is the training dataset with dimension $N \times n$, where N is the number of observations, n is the number of input features. Y is the target values of the training dataset with dimension $N \times 1$. L is the number of trees in RF. T_i refers to each decision tree in RF, where $i = 1, \ldots, L$. m is the number of features randomly selected in each node of decision tree.
1). In each decision tree T_i in RF, generate the training set by sampling N times from all observations with replacement. 2). In each node of one decision tree, m randomly selected features are used to calculate the best split criterion for T_i. 3). Repeat step 2 until the decision tree T_i is fully grown. 4). Aggregate the outputs given by all the decision trees to obtain final result. For classification, the output value is determined by majority vote. For regression, the mean or median of all the outputs is treated as the predicted value.

In fact, many impurity measures are not differentiable with respect to the hyperplane parameters, which causes the development of search techniques for finding the best hyperplane in each node of a decision tree. For example, in classification and regression trees (CART), Breiman uses a deterministic hill-climbing algorithm to search the best hyperplane parameters. Such search algorithms face two main problems: computationally cumbersome in high-dimensional feature spaces and local optimum problem. To avoid being trapped in a local optimal solution, multiple trials can be used to decrease the possibility, or evolutionary algorithms can be employed to achieve optimization in all dimensions [101, 102]. We refer the readers to [103] for more information about decision tree induction methods.

For regression and time series forecasting, RF based algorithms have been successfully applied to solve many real world problems with big data recently. In [104], RF based new methods were employed to downscale census data with a higher accuracy and increased processing efficiency. In [105], the authors investigated the feasibility of using RF for hotzone identification at macro-level with the help of massive amount of data. It is worth noting that there is no oblique RF based method for time series forecasting found in the literature.

2.2.7 Deep Belief Network

Deep learning is a branch of machine learning algorithms that attempts to model high-level abstractions in data by using model architectures with complex structures, with multiple non-linear transformations [106]. Deep learning algorithms are fundamentally based on distributed representations, which means that observed data can be represented by interactions of many different factors on different levels. The main promise of deep learning is replacing handcrafted features with efficient algorithms for unsupervised feature extraction [107]. In other words, deep learning attempts to abstract important features in input data set by deep architecture in an unsupervised way.

The DBN proposed by Hinton [32] provides a new way to train deep generative models, which is called layer-wise greedy pre-training algorithm. Figure 2.3 shows the schematic diagram of a DBN. There is no inter-connection between the units in each layer. An restricted Boltzmann machine (RBM) is a neural network which can learn the probability distribution over the input dataset. Figure 2.4 shows the network structure of an RBM.

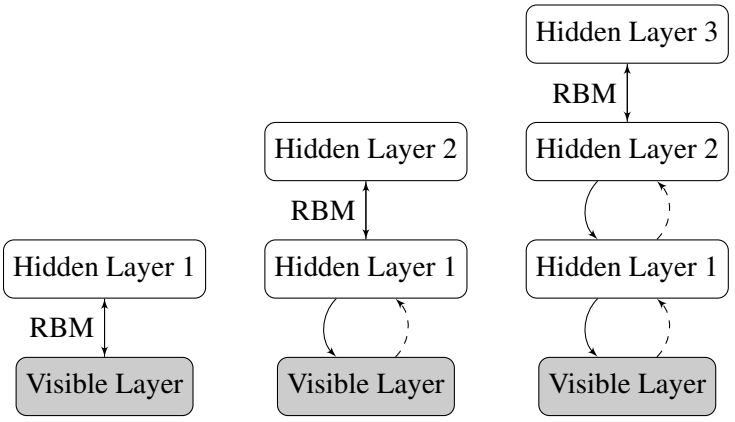


FIGURE 2.3: Flowchart of a Deep Belief Network (DBN)

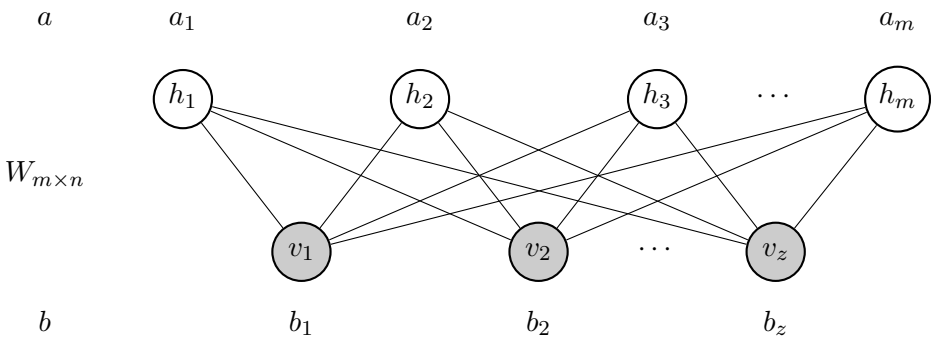


FIGURE 2.4: Schematic Diagram of a Restricted Boltzmann Machine (RBM)

The DBN pre-training procedure treats each consecutive pair of layers in the MLP as a restricted Boltzmann machine (RBM) [108] whose joint probability is defined as

$$P_{h,v}(h,v) = \frac{1}{Z_{h,v}} \cdot e^{(v^T W h + v^T b + a^T h)} \tag{2.26}$$

for the Bernoulli-Bernoulli RBM applied to binary v with a second bias vector b and normalization term $Z_{h,v}$, and

$$P_{h,v}(h,v) = \frac{1}{Z_{h,v}} \cdot e^{(v^T W h + (v-b)^T (v-b) + a^T h)} \tag{2.27}$$

for the Gaussion-Bernoulli RBM applied to continuous variable v [109]. In both cases the conditional probability $P_{h|v}(h|v)$ has the same form as that in an MLP layer.

The RBM parameters can be efficiently trained in an unsupervised fashion by maximizing the likelihood $\mathcal{L} = \prod_t \sum_h P_{h,v}(h, v(t))$ over training samples $v(t)$ with the approximate contrastive divergence algorithm [110].

The specific forms are given as:

$$\frac{\partial \mathcal{L}}{\partial W} = \sum_t v(t) E_{h|v}(h|v(t))^T - \sum_t \hat{v}(t) E_{h|v}(h|\hat{v}(t))^T \tag{2.28}$$

$$\frac{\partial \mathcal{L}}{\partial a} = \sum_t E_{h|v}(h|v(t)) - \sum_t E_{h|v}(h|\hat{v}(t)) \tag{2.29}$$

$$\frac{\partial \mathcal{L}}{\partial b} = \sum_t v(t) - \sum_t \hat{v}(t) \tag{2.30}$$

$$\hat{v}(t) = \sigma(W\hat{h}(t) + b) \tag{2.31}$$

where $\hat{h}(t)$ is a binary random sample from $P_{h|v}(\cdot|v(t))$.

To train multiple layers, one trains the first layer, freezes it, and uses the conditional expectation of the output as the input to the next layer and continues training next layers. Hinton and many others have found that initializing MLPs with pretrained parameters never hurts and often helps [32, 111].

2.3 Ensemble Learning

As discussed in Section 1.3, ensemble learning works by strategically combining multiple algorithms to achieve better performance. A typical ensemble learning framework is shown in Figure 2.5:

- Training data: A dataset (X, Y) is used for training, where X is the input data with dimension $N \times n$, Y is the target values of the input data with dimension $N \times 1$, N is the number of observations, n is the number of input features.
- Base learning model: The relatively weak learning algorithm which learns the generalized relationship between the input features and the target value.
- Diversity Generator: The algorithms used to increase the diversity of learning models. For example, in bagging, the training set is generated by sampling N times from all observations with replacement for each base learner. For boosting, the weights of each training sample is updated during the training process.

- Aggressor: The aggressor is responsible for combining the outputs of base learners the generate the final forecasting/classification result.

In Figure 2.5, X is the original training set. $X^{(i)}, i \in \{1, \dots M\}$ are the bootstrap version of datasets generated by the diversity generator. Moreover, $f^{(i)}$ are the base models, while f_{en} is the aggressor. The dashed red lines in the generation and base prediction parts denote boosting related ensemble framework, which indicates that the training is conducted in a sequential manner [112].

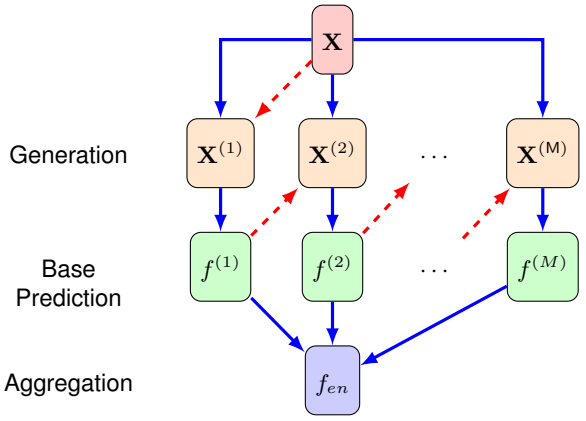


FIGURE 2.5: A typical framework of ensemble methods.

According to [113], there are two kinds of ensemble methods: competitive and cooperative ensemble classifiers/predictors. For competitive ensemble method, we train different predictors individually with different datasets or the same dataset but with different parameters and then aggregate the outputs of all the individual base learners. However, for cooperative ensemble method, the prediction task is divided 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 learners. In this thesis, both kinds of ensemble learning will be discussed along with their examples.

2.3.1 Competitive Ensemble Learning

Competitive ensemble learning, or multi-model ensemble learning, aims to construct better learning models by combining multiple base models with different initial conditions or different parameter settings. Therefore, the final forecasting results of the competitive ensemble learning model are generated by aggregating the outputs from all the base models or selected models. In [114], bias-variance-covariance decomposition is introduced, which shows that low-correlated base learners can benefit the overall performance of the ensemble method. That is

to say, diversity is a critical factor for competitive ensemble learning method, because low-correlated base models mean high degree of diversity. Some of the most frequently used ensemble methods based on data diversity are heterogeneous learning, bagging and boosting.

Heterogeneous Learning

Heterogeneous learning aims to leverage different types of heterogeneity such as task, view, data and model heterogeneity, to improve the learning performance. In [115], a heterogeneous ensemble method is proposed for electric load forecasting, which consists of two phases: overproduce and select. It first generates many different base models using different learning algorithms including EMD based models. Then, two base models with the best training performance are selected to generate the final forecasting results. The simulation results prove the advantages of heterogeneous learning methods. Another study on solar irradiance forecasting using heterogeneous learning model is reported in [116].

Bagging

Bagging, or bootstrap aggregation, is one of the most well-known independent ensemble learning methods. It samples with replacement from a given training set to generate diverse training sets for weak learners such as decision tree, single hidden layer neural networks and so on. In [57, 117], bagging is proved to significantly reduce the variance of base models.

There are many bagging related works in the literature. In [118], bagging is employed with ANN for short-term solar irradiance forecasting. The model consists of three kinds of neural networks: MLP, recurrent neural network (RNN) and RBFNN, which achieves the better forecasting result compared to single models. In [119], the performance of bagging, boosting and randomization are evaluated using 33 UCI datasets. The simulation results show that bagging is much better than boosting when substantial noise exists. In [120], bagging is used in Gaussian process regression models to obtain more robust and accurate predictions. In [121], the authors present an approach which uses a recurrent neural network to transform the spatio-temporal information of the input data in a new larger space, and then apply bootstrap techniques to improve the time series forecasting performance. Moreover, in [122], a combination of bagging (as a variance reduction technique) and boosting (as a bias reduction technique) is developed to create high precision and low variance ranking models.

Boosting

Adaptive Boosting (AdaBoost) is an extension of Bagging, which also belongs to sequential ensemble methods [123]. At the first stage, Bagging is employed to create a collection of subsets with high data diversity. Then it adjusts the weights of each data sample during each training iteration. Therefore, the overall performance of the group of base learners can be boosted.

There are many AdaBoost variants for both classification and regression problems, including modest AdaBoost [124], SpatialBoost [125], gradient boosting [126], AdaBoost.R [127], AdaBoost.RT [123], AdaBoost+ [128] and big error margin boosting [129]. In [130], Adaboost was improved in a setting in which hypotheses may assign confidences to each of their predictions, which was applied to both multi-class and multi-label problems. There are also several papers published to apply AdaBoost for TS forecasting problems. The examples include AdaBoost-ANN [131] and EMD-AdaBoost-ANN [62] for wind speed forecasting, and gradient boosting [126] for load forecasting.

Stacking

Stacking, also known as stacked generalization [132], is an ensemble method in which the aggregation stage is more generalized compared with bagging and boosting. In stacking based ensemble methods, the weights of the outputs from each base predictor are estimated by another machine learning method, which can also be a supervised learning algorithm. For example, in [133–135], SVM was used for aggregating the outputs from the base learners. Moreover, due to the supervised learning method in the aggregating stage, relatively weaker but faster base learners can be used , and it usually results in heterogeneous ensemble methods as mentioned above [136, 137].

2.3.2 Cooperative Ensemble Learning

Cooperative Ensemble Learning, which uses the concept of "Divide and conquer", works by decomposing the original task into a collection of sub-tasks until they are simple enough to be solved directly. Typical approaches include wavelet decomposition and empirical mode decomposition.

Wavelet Transform

Wavelet transform is a popular mathematical tool to decompose TS signal into its frequency components, which are beneficial for signal processing and analysis [138]. Wavelet transform

is similar to the Fourier transform with a main difference in the merit function. The wavelet transform uses a collection of wavelet functions to represent the original signal, while traditional Fourier transform decomposes the signal into sines and cosines [139].

In wavelet transform, all the generated wavelets are shifted and scaled copies of a basic wavelet, which is called mother wavelet. A mother wavelet is a square integrable function $\psi(t)$ and satisfies the admissibility condition, which ensures the reversibility of wavelet transform [139, 140]:

The discrete wavelet transform (DWT) decomposes the signal $x(t)$ into mutually orthogonal set of wavelets using a discrete set of the wavelet scales and translations, which is very effective for TS data sampled at a fixed interval of time [140]. With suitably chosen grid points on $s - \tau$ plane, the discrete wavelets can be defined as:

$$\psi_{j,k}(t) = s_0^{-j/2}\psi(s_0^{-j}t - k\tau_0), j, k \in \mathbb{Z} \tag{2.32}$$

where s_0 is the scaling factor which is usually chosen as 2, τ_0 is the fixed translation factor. The discrete wavelet decomposition of signal $x(t)$ can be represented by:

$$\begin{aligned} W_x(j,k) &= \int_{-\infty}^{\infty} x(t)\psi_{j,k}^*(t)dt \\ x(t) &= \frac{1}{c_\psi}\sum_{j,k\in\mathbb{Z}} W_x(j,k)\psi_{j,k}(t) \end{aligned} \tag{2.33}$$

Maximal Overlap Discrete Wavelet Transform (MODWT) [140], as a variant of DWT, applys low-pass and high-pass filters to the input signal at each level, which has some advantages over standard DWT. The main difference between MODWT and traditional DWT is that MODWT is highly redundant and non-orthogonal. The redundancy increases the effective degrees of freedom (EDOF) on each scale and thus decreases the variance of statistical estimates, which allows better comparison of TS signal with its decomposition [141]. Moreover, the MODWT does not decimate the coefficients in order to keep the number of wavelet and scaling coefficients the same as the number of input data samples at every level of the transform. Therefore, the MODWT is well-defined for all sample sizes N. However, for a complete decomposition of J levels, the DWT requires N to be a multiple of 2^J [139, 141].

Many DWT based decomposition methods are applied for TS signal in the literature [142–144]. For example, in [142], a number six Daubechies wavelet was used for DWT, results in three detailed decomposition and one approximated decomposition. The proposed wavelet-ARIMA model outperformed the conventional ARIMA method for short term wind speed forecasting. Moreover, a wavelet Recurrent Neural Network (RNN) for solar irradiance forecasting was reported in [145], in which the number seven Daubechies wavelets were used. The number of

hidden neurons was determined by cut-and trial method. The simulation results showed that the wavelet based RNN outperforms the RNN without wavelet decomposition.

Empirical Mode Decomposition

EMD [68], also known as HHT, is a method to decompose a signal into several intrinsic mode functions (IMF) along with a residue which stands for the trend. EMD is an empirical approach to obtain instantaneous frequency data from non-stationary and nonlinear data sets.

The system load is a random non-stationary process composed of thousands of individual components. The system load behavior is influenced by a number of factors, which can be classified as: economic factors, time, day, season, weather and random effects. Thus, EMD algorithm can be very effective for load demand forecasting.

An IMF is a function that has only one extreme between zero crossings, along with a mean value of zero. The shifting process which EMD uses to decompose the signal into IMFs is described as follows:

1. For a time series signal $x(t)$, let m_1 be the mean of its upper and lower envelopes as determined by a cubic-spline interpolation of local maxima and minima.
2. The first component h_1 is computed by subtracting the mean from the original time series: $h_1 = x(t) - m_1$.
3. In the second shifting process, h_1 is treated as the data, and m_{11} is the mean of h_1's upper and lower envelopes: $h_{11} = h_1 - m_{11}$.
4. This sifting procedure is repeated k times until one of the following stop criterion is satisfied: i) m_{1k} approaches zero, ii) the numbers of zero-crossings and extrema of h_{1k} differs at most by one, or iii) the predefined maximum iteration is reached. h_{1k} can be treated as an IMF in this case and computed by: $h_{1k} = h_{1(k-1)} - m_{1k}$.
5. Then it is designated as $c_1 = h_{1k}$, the first IMF component from the data, which contains the shortest period component of the signal. We separate it from the rest of the data: $x(t) - c_1 = r_1$. The procedure is repeated on r_j: $r_1 - c_2 = r_2, \ldots, r_{(n-1)} - c_n = r_n$.

As a result, the original time series signal is decomposed as a set of functions:

$$x(t) = \sum_{i=1}^{n}(c_i) + r_n \tag{2.34}$$

where the number of functions n in the set depends on the original signal.

Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) [146] is an improved version of EMD, which is designed to solve the mode mixing problem and reduce the high computational cost caused by EMD and its other variants. This technique adds a Gaussian white noise with unit variance and a noise coefficient to the signal instead of just adding white noise. The resulting decomposition is complete, with a numerically negligible error.

In the literature, there are many research works applying EMD and its variants for TS forecasting. [16, 62, 63, 147]. For example, in [148], an EMD-SVR model was developed for short-term load forecasting. In [149], after decomposition by EMD, the sub-signals were divided into high-frequency and low-frequency subsets, which were modeled by SVR and AR, respectively. The rational behind the ensemble model is that the low frequency components are more towards linear TS, which can be modelled by linear models.

Part I

Ensemble Learning with Applications in Power Systems

Chapter 3

Oblique Random Forest Ensemble via Least Square Estimation for Electric Load Forecasting

It is well known that electricity power supply planning plays an important role in the management of modern power system. Accurate forecasting is beneficial for unit commitment, power system security, as well as energy transfer scheduling [150]. As stated in [151], for short-term electric load forecasting, the ballpark saving from 1% reduction in forecast error for a utility with $1GW$ peak is roughly $\$300,000$ per year. Therefore, the goal of load forecasting can be concluded as providing reliable power supply while keeping the operation costs and energy wastage as low as possible.

In this thesis, various ensemble learning methods are proposed and investigated for short term electric load forecasting, including decision tree ensembles, ensemble deep learning methods and ensemble incremental learning models. In this chapter, a decision tree ensemble method, named oblique random forest with least square estimation, is introduced and evaluated using electric load data [77].

3.1 Characteristics of Electric Load Data

In this part, the electricity load demand data sets from Australian Energy Market Operator (AEMO) were used for the simulation [160]. As shown in Figure 3.1, which is the time plot for electricity load demand data with two weeks time window, the electricity load demand data

is sampled every half an hour and has 48 data points for one day. Moreover, the electricity load data shows three main nest cycles: daily, weekly and yearly, which are caused by various influence factors such as climate and social activities.

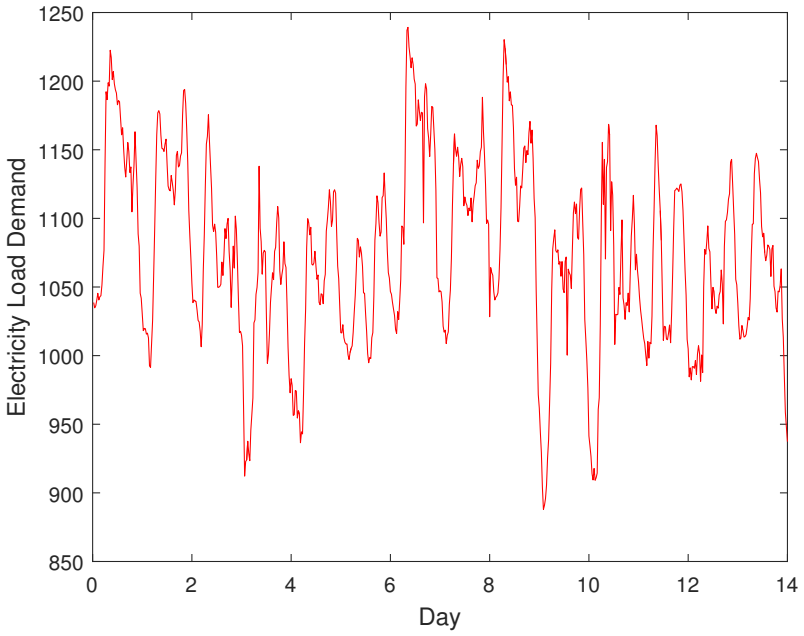


FIGURE 3.1: Time plot of load demand data with a time window of two weeks

To identify cycles and patterns in load demand TS, we employ autocorrelation function (ACF) as a guidance for informative feature subset selection. Suppose a time series data set is given as $X = \{X_t : t \in T\}$, where T is the index set. The lag k autocorrelation coefficient r_k can be computed by:

$$r_k = r(X_t, X_{t-k}) = \frac{\sum_{t=k+1}^{n}(X_t - \bar{X})(X_{t-k} - \bar{X})}{\sum_{t=1}^{n}(X_t - \bar{X})^2} \tag{3.1}$$

where $\bar{X}$ is the mean value of all X in the given time series, r_k measures the linear correlation of the time series at times t and $t - k$.

From Figure 3.2, which shows the ACF for electricity load demand data with a time window of one week, three strongest dependent lag variables can be identified: the value of previous half-an-hour (X_{t-1}), the value at the same time in the previous week (X_{t-336}), as well as the value at the same time in the previous day (X_{t-48}). Therefore, in order to take all the most informative lag variables into consideration, we select the data points of the whole previous day (X_{t-49} to X_{t-96}) and the same day in the previous week (X_{t-337} to X_{t-384}) to construct the input feature set for one day ahead load demand forecasting.

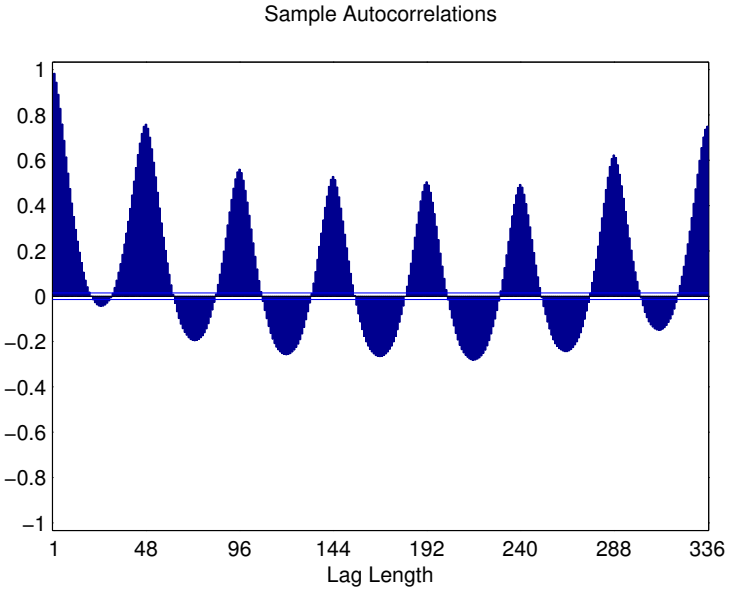


FIGURE 3.2: Autocorrelation function for electricity load demand data in TAS

3.2 Proposed Ensemble Method with Oblique Random Forest

In this work, motivated by the excellent performance of random forest [16, 49],we propose a novel oblique random forest for TS forecasting. The contributions of this work are in three aspects. First of all, although standard RF has been widely used for classification, regression and time series forecasting problems [49], oblique RF still needs further investigation in these research fields.Hence, we make the first attempt to study the oblique RF in the context of time series forecasting, especially for electricity load demand forecasting. Secondly,instead of impurity score based selection method, we alternatively propose to use least-square classifier in each node of the decision trees to perform data partitioning, which performs both better and fast. Finally, the advantage of the proposed method is demonstrated using eight generic TS datasets and five electricity load demand datasets compared with six benchmark algorithms: persistence, support vector regression (SVR), single-hidden layer feedforward neural network (SLFN), deep belief network (DBN), random forest (RF) and the multisurface proximal support vector machine based oblique random forest (MPSVM-RF).

In this section, the theory of the proposed least square classifier based oblique RF is explained. Moreover, we show the advantages of the proposed method in view of computational complexity.

3.2.1 Oblique Random Forest

Conventional random forest [98] employs orthogonal (or univariate/ axis-parallel) decision tree which explicitly searches for an "optimal" feature candidate from the feature subset to split the data based on impurity criteria such as Gini impurity, information gain and so on [45]. As an alternative, oblique (multivariate) random forest usually searches for an oblique hyperplane in each internal node based on a linear combination of random selected feature subset. In [152], Murthy et al. introduced OC1, which is an algorithm for generating multivariate decision trees based on randomized search procedures. In [101, 102], oblique decision trees are induced with evolutionary algorithms to perform optimization in multiple dimensions simultaneously. In [153], Zhang et al. proposed an oblique random forest with extremely randomized trees, which shared the same concept introduced in [154].

Please note that all the above method are based on selecting one hyperplane based on the impurity score. As reported in [100], all of the impurity measures only depend on the distribution of different classes on each side of the hyperplane, thereby they do not really capture the geometric structure of class regions. Instead of only taking the label information into consideration, the geometric structure also focuses on the internal data structure including the distance of the data point to the decision hyperplane. Therefore the hyperplane will change as any of the relevant feature changes. The problem is well addressed in the classification domain [49, 100]. In each node of the decision tree, a fast support vector machine variant is employed to better capture the geometrical structure of the data samples. However, this issue remains untouched in the context of time series forecasting. Motivated by this, in this work we fill in this research gap by making the first attempt to propose an oblique random forest for time series forecasting. In each node of a decision tree, we alternatively propose to use least-square classifier in each node to perform data partitioning.

Least square fitting is a mathematical process to find the best-fitting curve to a given dataset by minimizing the sum of the squares of the errors. The obtained best fitting curve can be applied as a hyperplane to perform classification at each node of decision trees, in which the whole randomly selected feature subset is utilized for training.

Generating the decision tree to random forest is straight-forward. The algorithm of the proposed method is shown as follows. Moreover, Figure 3.3 is the flow chart of the proposed method. The definition of the symbols is the same as in Table 2.1.

1. For a time series signal $x(t)$, construct the training dataset X by using n historical data points as input features, which means that the input feature set has the dimension of n. The corresponding target vector is y.

2. For each decision tree T_i in the proposed oblique RF, we generate the training set by sampling N times from all available observations with replacement.

3. In the j^{th} node n_{ij} of decision tree T_i, m features are randomly selected from totally n features to perform partition.

4. To perform partition for regression problem, we first find the median value m_{ij} of the target vector in the node n_{ij}. Then all the samples whose target value is smaller than or equal to m_{ij} will be labeled as -1, while the samples with target value bigger than m_{ij} will be labeled as 1. At this stage, least square method is applied to find the best hyperplane which divide the sample subset into two classes and transfer them to two children nodes.

5. Repeat step 3 & 4 until the decision tree T_i is fully grown. There are two stop criterion: i). the tree reaches a pure node; ii). the number of samples reaches its minimum value. All the leaf nodes are labeled by the mean value of the target values of all the samples in this leaf node.

6. Repeat step 2 to 5 to train totally L decision trees, thereby constructing an oblique RF model.

7. To perform forecasting, for example, the feature vector of one sample $X(a)$ is inputted to the obtained oblique RF without the target value $y(a)$. After going through all the decision trees, totally L outputs are generated. Then aggregate the outputs from all of the decision trees in the proposed oblique RF to calculate the forecasting value. In this work, the mean value of all the outputs is used as the final forecasting result.

In Section 3.4 and 3.5, we empirically show that the proposed method lead to better performance compared with the multisurface proximal SVM (MPSVM) based oblique RF [49].

3.3 Experiment Setup

Some details of the experimental setup are shown in this section, such as normalization, implementation, parameter selection and performance estimation.

3.3.1 Methodology

Normalization is a "scaling down" transformation of the features to avoid a large difference among input variables, which is very important for neural network based methods. In the paper,

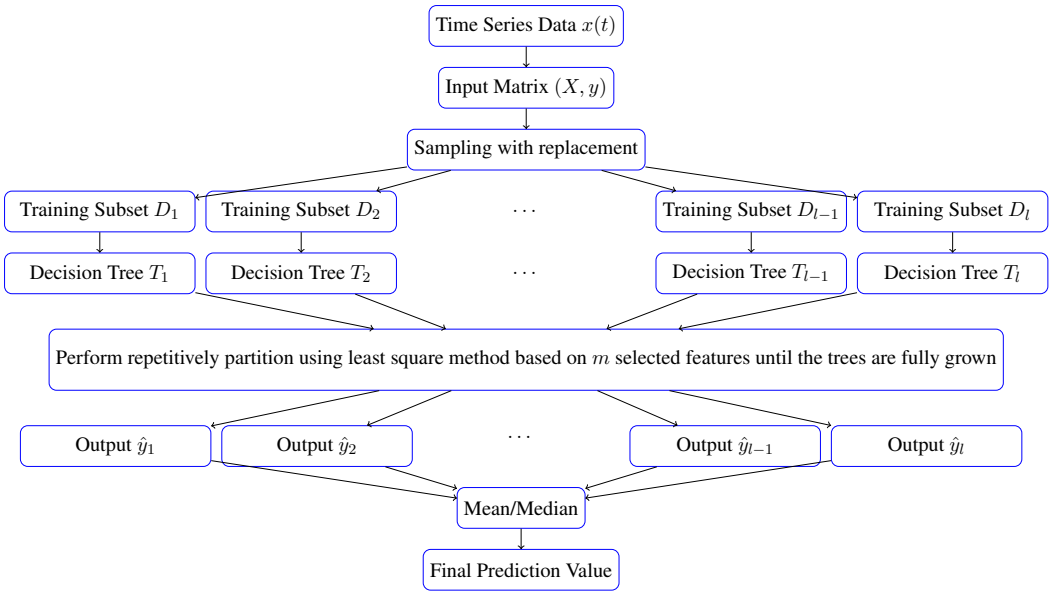


FIGURE 3.3: Schematic Diagram of the Proposed Oblique Random Forest

min-max normalization within [0,1] interval is applied to all the training and testing values.

$$y_i' = \frac{y_{max} - y_i}{y_{max} - y_{min}} \tag{3.2}$$

where y_i is the original value and y_i' is the scaled value.

To implement the simulation, LIBSVM toolbox is used for SVR model [155], while deep learning toolbox in Matlab is utilized to develop SLFN and DBN [156]. RF is developed from the function "Treebagger" in Matlab. We set the parameter "NumPredictorsToSample" as one third of the number of input features to invoke RF algorithm. The proposed method and MPSVM-RF are developed by the authors using Matlab [49].

For SVR, we choose RBF kernel function with its parameters chosen by a grid search. For SLFN, the number of BP iterations is also selected by grid search, whose searching range is $[200, 1000]$ with a step size of 100. Moreover, for comparison based on the same condition, the numbers of decision trees in RF, MPSVM-RF and the proposed method are both 500 [49].

3.3.2 Performance Estimation

Two error evaluation measures are used to examine the accuracy of the forecasting results in this paper: Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE), which are defined as:

$$
\begin{aligned}
RMSE &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i' - y_i)^2} \\
MAPE &= \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i' - y_i}{y_i}\right|
\end{aligned}
\tag{3.3}
$$

where y_i' is the predicted value of corresponding y_i, and n is the number of data points in the time series.

3.4 Assessment on generic time series datasets

To demonstrate the effectiveness of Oblique RF along with all the benchmark methods to obtain generalized conclusion, eight generic TS datasets are chosen to estimate the performances [54, 157]. The statistics of the eight datasets are summarized in 3.1.

TABLE 3.1: Summary of the eight generic TS datasets

Dataset	Length	Min	Median	Mean	Max	Std
Manchas	176	0.1	39.55	44.76	154.4	34.77
Poluicao-SO_2	365	3.346	17.65	19.6	56.9375	10.26
Poluicao-PM_{10}	365	25.518	52.972	60.34	152.976	25.1987
Ipi	187	65.81	103.34	103.627	148.31	18.28
Bebida	187	49.63	91.97	92.658	134.62	20.35
Fortaleza	149	468	1399	1445.3	2836	496.952
Atmosfera	365	53.34	81.97	81.1516	95.79	7.9579
Consumo	154	75.39	116.355	120.9447	232.01	27.2448

In this work, three assessments were implemented for each generic TS dataset, which were on 1, 5 and 10 steps ahead forecasting. Each dataset was partitioned into two parts: the first 70% was used for training and the remaining 30% was used for testing. Because there are small values close to zero in the datasets Manchas and Poluicao-SO_2, MAPE can not be used for error measurement. Therefore, the performances were only measured by RMSE. Min-max normalization within [0,1] interval was applied to TS data. The input feature set is composed of lag values with length of $1/20$ of the total length.

Totally seven methods including persistence method were simulated for the assessment. Persistence method is actually the simplest forecasting model, which assumes that the conditions at the future time of forecast are the same as past values. The persistence method normally performs well for very short term time series forecasting since there is little change for nonlinear influencing factors during a short period time. Therefore, the persistence method can be treated as a baseline for evaluating the effectiveness of learning models.

The forecasting results are shown in Table 3.2. Friedman test and Nemenyi post-hoc test were applied to test the statistical difference among all the learning models and the results are shown in Figure 3.4.

TABLE 3.2: Forecasting results for eight generic time series datasets

Dataset	Horizon	Prediction model						
		Persistence	SVR [27]	SLFN [158]	DBN [32]	RF [49]	MPSVM-RF [43]	Proposed
Manchas	1	19.91	15.07	18.60	**13.39**	17.13	17.47	13.94
	5	56.19	21.69	40.02	31.41	31.40	22.74	**21.22**
	10	32.59	24.65	25.54	26.68	27.78	**19.42**	23.14
Poluicao-SO_2	1	4.35	5.30	6.45	**4.09**	7.17	8.49	7.10
	5	**5.57**	8.08	6.07	6.79	10.70	10.94	10.60
	10	6.44	9.05	**6.40**	8.14	10.39	11.91	11.81
Poluicao-PM_{10}	1	14.56	21.16	24.48	17.57	17.40	**14.51**	15.18
	5	20.88	**16.49**	30.77	25.04	23.33	17.78	19.72
	10	25.49	**17.09**	30.55	26.92	27.60	19.42	21.77
Ipi	1	10.59	15.25	22.94	11.22	15.41	16.52	**6.31**
	5	30.81	13.33	14.65	7.61	12.99	15.34	**6.08**
	10	16.23	15.97	18.54	7.66	13.55	17.08	**7.41**
Bebida	1	13.17	12.33	12.46	12.61	12.21	**11.44**	12.22
	5	16.07	11.83	11.92	13.06	12.05	12.57	**11.79**
	10	14.82	23.05	24.67	**10.58**	16.50	20.37	16.03
Fortaleza	1	569.77	546.76	**485.81**	504.55	490.55	504.19	486.30
	5	769.58	564.90	595.68	554.05	579.62	**547.16**	552.75
	10	642.19	527.00	590.42	**502.73**	533.59	509.50	506.01
Atmosfera	1	7.90	7.55	6.76	6.80	7.54	7.31	**6.75**
	5	9.41	9.09	11.42	**7.61**	8.34	7.82	7.90
	10	9.84	9.96	11.06	**7.77**	9.28	8.01	8.10
Consumo	1	19.68	**13.03**	13.71	15.37	15.08	15.91	18.12
	5	19.41	15.64	15.06	15.22	**14.03**	14.61	14.87
	10	21.10	**10.05**	12.60	10.41	13.27	16.51	13.49

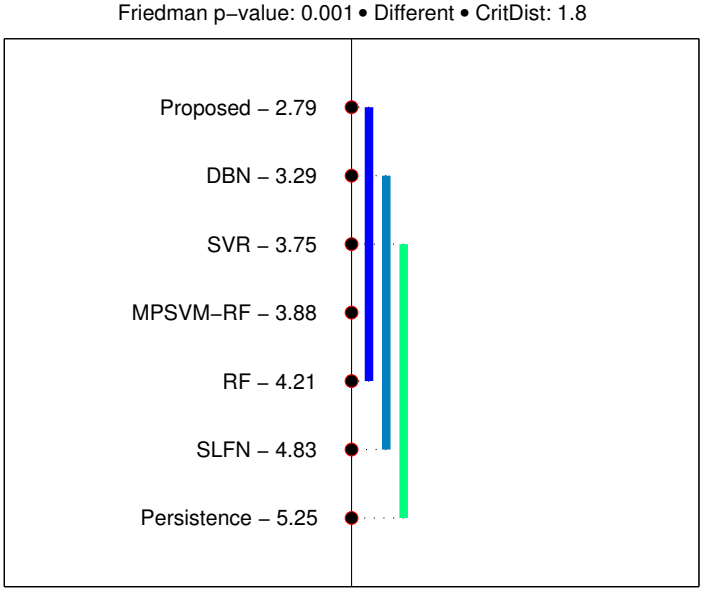


FIGURE 3.4: Nemenyi testing for generic TS forecasting. The critical distance is 1.8.

As shown in in Figure 3.4, the methods with better ranks are at the top whereas the methods with worse ranks are at the bottom. The methods have statistically the same performance if they

are within a vertical line whose length is less than or equal to a critical distance. The critical distance is calculated by:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{3.4}$$

where k is the number of algorithms, N is the number of data sets, and q_α is the critical value based on the Studentized range statistic divided by $\sqrt{2}$ [159].

Therefore, we can conclude from the statistical testing results that all the benchmark learning models have better rank compared with persistence method. The proposed oblique RF significantly outperforms SLFN with a 95% confidence, and has better rank compared with DBN, SVR, MPSVM-RF and standard RF.

3.5 Experiments of electricity load demand time series forecasting

In this work, five electricity load demand datasets are chosen to verify the performance of the proposed ensemble method.

The electricity load demand datasets published by Australian Energy Market Operator (AEMO) were chosen for the simulations [160]. Especially, the load data of year 2013 from five states in Australia were selected to train and test the proposed ensemble method: New South Wales (NSW), Tasmania (TAS), Queensland (QLD), South Australia (SA) and Victoria (VIC). Considering the fact that the performance is influenced by large sample size [5], the load data of four months for each state were chosen to reduce the factors of different seasons: January, April, July and October. Therefore, there are totally 20 datasets for the simulations, in which the data of first three weeks was used to train the model, while the the remaining data was used for testing.

In this study, for electricity load demand forecasting, the forecasting horizon of one-day ahead is adopted for the comparison, which belongs to short term load demand time series forecasting. In the field of machine learning, short term time series forecasting is most widely researched, because in short term the historical data is more meaningful for forecasting and thus has relatively high accuracy, which can be benefit industry plan designing."

For one-day ahead forecasting, due to the strong daily seasonality in the load demand data, the accuracy of persistence method still remains in a reasonable range. Therefore, we still use the persistence method as the baseline. Table 3.3 shows the forecasting results for one-day ahead electricity load demand forecasting. The numbers in bold mean that the corresponding method has the best performance for this dataset under this performance measure. Form this table, we can conclude that all the benchmark learning models can outperform the persistence method. Table 3.4 shows the computation time of all the benchmark models.

TABLE 3.3: Forecasting results for one day ahead electricity load demand forecasting

Dataset	Month	Metrics	Prediction model						
			Persistence	SVR [27]	SLFN [158]	DBN [32]	RF [49]	MPSVM-RF [43]	Proposed
NSW	Jan	RMSE	978.24	703.43	750.53	639.75	521.14	619.91	**509.69**
		MAPE	8.55%	6.23%	7.2%	5.95%	**4.26%**	5.94%	4.40%
	Apr	RMSE	729.50	474.38	578.05	**361.63**	500.70	477.50	495.08
		MAPE	6.71%	4.27%	5.41%	**3.36%**	4.25%	4.41%	4.19%
	Jul	RMSE	609.82	574.30	534.75	415.81	387.15	464.72	**338.51**
		MAPE	6.22%	5.86%	5.38%	4.11%	4.01%	4.72%	**3.15%**
	Oct	RMSE	587.14	393.32	345.07	350.82	**296.53**	347.20	313.38
		MAPE	5.36%	3.74%	3.48%	3.41%	**2.78%**	3.49%	3.09%
TAS	Jan	RMSE	89.82	60.97	69.92	63.96	60.68	64.88	**58.80**
		MAPE	7.24%	4.81%	5.42%	4.98%	4.77%	5.11%	**4.61%**
	Apr	RMSE	157.73	111.89	94.40	93.81	**92.64**	104.46	92.68
		MAPE	10.22%	7.48%	6.3%	6.12%	**6.10%**	6.59%	**6.10%**
	Jul	RMSE	120.47	90.99	89.17	87.30	90.48	91.51	**85.18**
		MAPE	8.11%	5.89%	6.28%	6.04%	6.17%	6.35%	**5.72%**
	Oct	RMSE	109.46	79.45	72.86	75.73	69.80	72.33	**67.79**
		MAPE	7.48%	5.55%	5.24%	5.15%	4.63%	4.92%	**4.62%**
QLD	Jan	RMSE	461.09	282.07	299.32	228.86	**195.85**	234.30	208.17
		MAPE	5.25%	3.65%	3.61%	2.78%	**2.41%**	3.01%	2.59%
	Apr	RMSE	489.63	266.39	339.93	247.56	231.01	276.96	**222.71**
		MAPE	6.25%	3.53%	3.77%	2.99%	2.78%	3.52%	**2.66%**
	Jul	RMSE	430.46	223.17	203.00	213.20	**156.08**	194.29	174.86
		MAPE	5.90%	3.10%	3.03%	2.95%	**2.32%**	2.92%	2.46%
	Oct	RMSE	417.33	298.76	263.12	251.34	236.50	259.43	**221.02**
		MAPE	5.54%	3.93%	3.46%	3.40%	2.88%	3.38%	**2.82%**
VIC	Jan	RMSE	990.74	587.98	811.43	915.21	739.65	**584.31**	646.61
		MAPE	9.48%	7.16%	9.32%	8.79%	8.77%	**7.11%**	7.46%
	Apr	RMSE	669.87	330.93	359.03	353.02	366.16	445.74	**331.71**
		MAPE	8.40%	4.43%	4.95%	4.55%	4.65%	5.74%	**4.31%**
	Jul	RMSE	721.85	297.07	305.88	**276.25**	302.15	283.61	295.12
		MAPE	9.76%	4.38%	4.29%	**3.72%**	4.29%	4.18%	4.25%
	Oct	RMSE	577.70	391.11	347.91	389.06	364.32	352.66	**314.91**
		MAPE	8.30%	4.50%	4.79%	4.85%	4.16%	4.29%	**3.96%**
SA	Jan	RMSE	433.57	337.10	411.66	401.25	349.87	347.85	**302.42**
		MAPE	14.32%	13.34%	13.72%	13.62%	13.41%	13.20%	**12.01%**
	Apr	RMSE	180.20	124.43	119.4	**117.61**	127.90	123.80	122.75
		MAPE	9.36%	6.71%	6.42%	6.67%	6.88%	6.51%	**6.36%**
	Jul	RMSE	289.94	150.84	151.06	**148.23**	154.67	152.50	157.57
		MAPE	16.84%	8.54%	8.66%	**8.50%**	8.95%	8.85%	8.92%
	Oct	RMSE	240.53	210.72	233.48	204.16	218.30	226.04	**201.26**
		MAPE	11.54%	8.94%	10.03%	9.33%	9.11%	10.04%	**8.33%**

TABLE 3.4: Average computation time of electric load forecasting models

SVR	SLFN	DBN	RF	MPSVM-RF	Proposed
21.34s	18.91s	34.89s	15.81s	16.59s	14.91s

There are five benchmark methods implemented to perform the comparison: SVR, SLFN, DBN, standard RF and MPSVM-RF. According to Table 3.3, the proposed method achieves the best performance for 23 measurements out of 40, RF outperforms the other methods in 9 measurements, DBN achieves the best performance 7 times, while MPSVM-RF performs the best in remaining 2 times. By comparing the results between DBN and SLFN, the learning ability of deep learning can be verified. Moreover, DBN, standard RF and MPSVM-RF are able to achieve the best performance for certain measures, which means that both of deep learning and ensemble methods can improve the performance of basic learning models. In addition, the proposed oblique RF outperforms the standard RF in most cases, which shows the advantages of multivariate models for electricity load demand TS forecasting.

As we have mentioned in previous section, the forecasting results need to be tested by statistical testing method to verify the conclusions made above. Therefore, the Friedman test and Nemenyi post-hoc test are applied to perform the comparison among all the learning models. Figure 3.5 and Figure 3.6 show the statistical testing results for load demand forecasting based on RMSE and MAPE, respectively. As shown in the figures, the Friedman $p-$values are very small which

can be concluded that there are significant performance differences among the six different methods. Moreover, the methods with better ranks are at the top, while the methods with worse ranks are at the bottom. If the distance between two methods is larger than the critical distance, which is 2.0 in this testing, then it can be said that there exists statistical significant difference between these two methods. Therefore, we can conclude that the proposed method has the best rank and significantly outperforms SVR and SLFN for electricity load demand forecasting with a 95% confidence. It is also worth noticing that MPSVM-RF has similar performance with RF and DBN, and has been significantly outperformed by the proposed method based on MAPE.

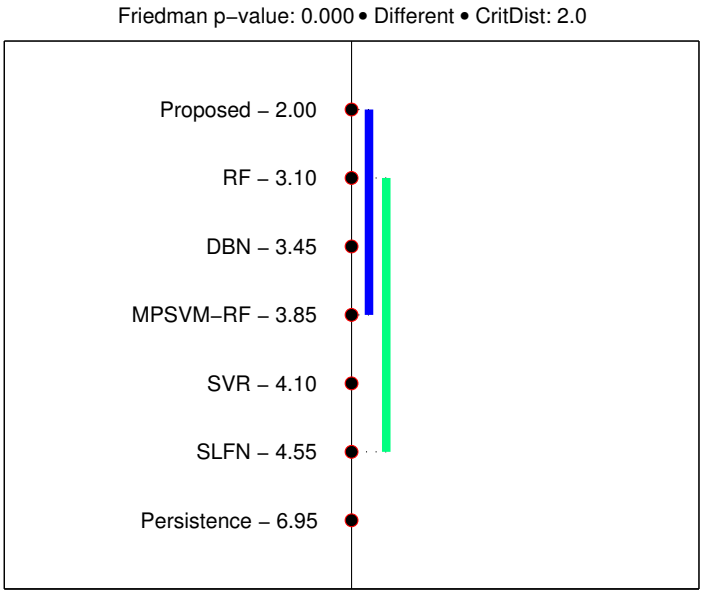


FIGURE 3.5: Nemenyi testing for electricity load demand forecasting based on RMSE. The critical distance is 2.0.

3.6 Summary

In this chapter, we make the first attempt to study the oblique RF in the context of time series forecasting, especially for electricity load demand forecasting. In each node of the decision trees, instead of the single "optimal" feature based orthogonal standard random forest, a least square classifier is employed to perform partition. To reveal the superior performance of the proposed least square based oblique RF, we have conducted a comparison with six benchmark algorithms using eight generic time series datasets and five load demand datasets. The following conclusions are made from the forecasting results:

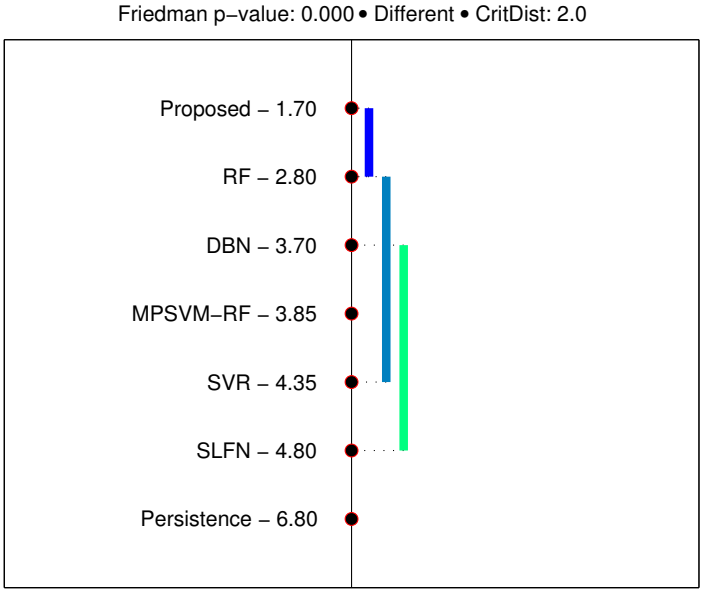


FIGURE 3.6: Nemenyi testing for electricity load demand forecasting based on MAPE. The critical distance is 2.0.

1. Proposed oblique RF has better performance compared with the original RF in both generic TS assessment and short term electricity load demand forecasting, which shows the advantages of multivariate methods.

2. Random Forest and the proposed oblique variant, as ensemble methods, they outperform SVR and SLFN for short term electricity load demand forecasting.

3. Proposed least square estimation based oblique RF outperforms MPSVM based oblique RF for time series forecasting, which indicates the importance of algorithm selection for hyperplane generation in oblique RF.

4. Deep learning has comparable performance with RF based methods in both experiments, which shows its learning ability for highly nonlinear features.

Chapter 4

Ensemble Deep Learning for Electric Load Time Series Forecasting

As we have mentioned in Chapter 1, deep learning is one of the most successfully machine learning family in the literature. In this chapter, two ensemble deep learning methods based on deep belief networks are presented: ensemble deep belief network (EDBN) [34] and EMD based ensemble deep belief network (EMD-DBN) [16].

4.1 Ensemble Deep Belief Network

For regression and time series forecasting, the prediction results can be different when the number of epochs of back propagation training is changed. Therefore, we can combine all the outputs generated by ANNs trained with different number of epochs. By analyzing the relationships between these outputs and target output values, it is possible to assign each output a corresponding weight value to compute the overall predicted output value. In this work, we choose an ensemble of deep learning algorithm composed of DBNs trained using different number of epochs and an SVR with inputs as the outputs of the DBNs and output as the final prediction. This work has been published in 2014 IEEE Symposium Series on Computational Intelligence [34]. The detailed procedure is shown as follows:

1. Train a DBN by using the input data matrix $\mathbf{X}$.
2. By setting the back propagation epochs from 100 to 2000 with step size equal to 100, we are able to get 20 prediction outputs y_1 to y_{20} [161]. The DBN is re-initialized 20 times.
3. Put all the outputs into a matrix $\mathbf{X}_{new}$, which is used to train an SVR with the expected prediction values $\mathbf{Y}$.

4. Finally, the output of the SVR is treated as the final prediction result.

Figure 4.1 shows the overall schematic of this ensemble method.

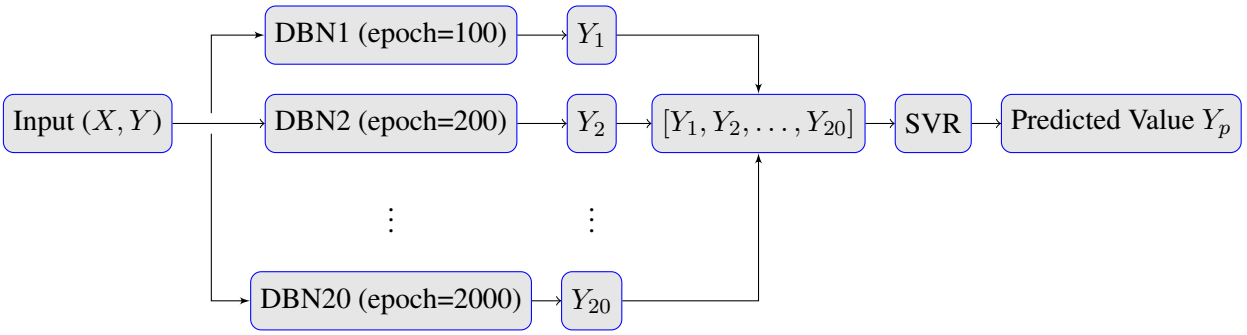


FIGURE 4.1: Schematic Diagram of the proposed Ensemble Deep Learning Network

4.1.1 Datasets

There are four time series datasets (Mackey-Glass dataset and three electricity load demand datasets) and three regression datasets (2D planes, Friedman Artificial Domain and California Housing datasets) used in the comparison.

The Mackey-Glass dataset is a time series generated by the Mackey-Glass equation to model the blood cell regulation. This dataset is widely used in the literature as a benchmark for prediction models. In the experiment, 9000 data points were used with first 6000 data points for training and the remaining 3000 data for testing.

The electricity load demand data sets from Australian Energy Market Operator (AEMO) were also used for the comparison [160]. Especially, the data sets of year 2013 from New South Wales (NSW), South Australia (SA) and Tasmania (TAS) were chosen to train and test the proposed method. The first nine months was used to train the model, and the last three months was used as the testing set. Thus, there are totally 13100 examples for training and 4370 examples for testing.

2D planes dataset is an artificial data set generated by equations introduced in [162][163]. Friedman Artificial Domain dataset was first generated in [164] and also described in [165][163]. For both of these datasets, 10000 data points were used with the first 7500 for training and the remaining 2500 for testing.

California Housing dataset is generated by collecting information on related variables in California from the 1990 Census [166][163]. The final data contains 20640 observations with 9 inputs, while the output is the median house value. In this work, 15480 data points were used for training and the remaining 5160 data points were used for testing.

4.1.2 Results

Prediction Results for Time Series Forecasting

From prediction results for Mackey-Glass data set in Table 4.1, the effectiveness of all of these prediction methods can be appreciated.

TABLE 4.1: Prediction results for Mackey-Glass Time Series

Mackey-Glass	SVR	ANN	DBN	ENN	Proposed
Training RMSE	0.0025	0.002	0.0018	0.0012	**0.0015**
Testing RMSE	0.0024	0.002	0.0018	0.0226	**0.0015**
Training MAPE	1.25%	2.33%	2.06%	0.87%	**0.43%**
Testing MAPE	1.03%	2.45%	2.17%	1.13%	**0.43%**

The comparison results for time series load demand forecasting are shown in Tables 4.2, 4.3 and 4.4. For SVR, we choose the RBF kernel function with parameters chosen by a grid search. The range of C is $[2^{-4},2^{4}]$, and the range of σ is $[10^{-3},10^{-1}]$. For ANN, the size of the neural network is [48 96 1], which is a one hidden layer model. The sizes of RBM in DBN is [20 20]. For the proposed ensemble learning method, 20 DBNs and the SVR have the same parameters as listed above. ENN is the ensemble version of the 20 ANNs (trained using epochs ranging from 100 to 2000) and combined using an SVR. The last column named AEMO shows the average forecasting error in year 2013 given on the AEMO website [160].

By analyzing the forecasting outputs in Tables 4.2 to 4.4, we can find that these methods also perform well for short-term time series load demand forecasting. Moreover, The ensemble learning methods have more accurate outputs than single structure algorithms. However, SVR has a slightly better forecasting performance than artificial neural networks for these data sets. This phenomenon is probably caused by the reason that there is only one hidden layer in the neural networks used here [167]. The prediction results of DBN are better than ANN, which shows the advantage of deep learning methods. Most outstandingly, the proposed ensemble deep learning method composed of DBN and SVR has yielded both the best training reconstruction results and the most accurate prediction outputs.

TABLE 4.2: Prediction results for load demand of New South Wales

NSW	SVR	ANN	DBN	ENN	Proposed	AEMO
Training RMSE	75.5476	99.4513	90.4974	79.1079	**59.8561**	/
Testing RMSE	74.3053	95.8105	90.2061	78.6394	**72.2545**	/
Training MAPE	2.25%	3.00%	2.73%	2.34%	**1.79%**	2.00%
Testing MAPE	2.83%	4.12%	3.50%	2.96%	**2.71%**	/

TABLE 4.3: Prediction results for load demand of South Australia

SA	SVR	ANN	DBN	ENN	Proposed	AEMO
Training RMSE	40.6467	36.8863	33.1023	32.3606	**26.6159**	/
Testing RMSE	44.6742	38.8585	35.9375	34.9473	**30.5989**	/
Training MAPE	3.64%	4.38%	3.74%	3.63%	**3.35%**	4.53%
Testing MAPE	5.30%	6.22%	5.70%	5.32%	**4.98%**	/

TABLE 4.4: Prediction results for load demand of Tasmania

TAS	SVR	ANN	DBN	ENN	Proposed	AEMO
Training RMSE	18.7509	18.9368	19.0076	19.9086	**18.3066**	/
Testing RMSE	20.1068	19.7952	19.9187	19.9034	**19.7580**	/
Training MAPE	3.00%	3.06%	3.05%	3.06%	**3.01%**	3.17%
Testing MAPE	3.43%	3.41%	3.41%	3.41%	**3.38%**	/

Prediction Results for Regression

Similar to time series part, the comparison results for regression are shown in Tables 4.5, 4.6 and 4.7. For 2D plane and Friedman Artificial Domain datasets, the size of the neural network is [10 20 1], while the size for California housing is [9 18 1]. The rest of parameters are the same as before. To make the numbers more comparable, the RMSE values for regression forecasting were calculated using scaled data.

From prediction results for regression, we can have similar conclusions as in time series forecasting. Especially, for California Housing result in Table 4.7, the advantage of ensemble deep learning method is outstanding. Therefore, compared with the performance on simple equation generated datasets, the ensemble deep learning method demonstrates much stronger ability on real complicated regression problems.

TABLE 4.5: Prediction results for 2D planes dataset

CART	SVR	ANN	DBN	ENN	Proposed
Training RMSE	0.0399	0.0425	0.0397	0.0402	**0.0321**
Testing RMSE	0.0406	0.0420	0.0412	0.0428	**0.0403**
Training MAPE	7.52%	8.61%	7.50%	7.56%	**6.11%**
Testing MAPE	7.49%	8.16%	7.59%	7.61%	**7.49%**

TABLE 4.6: Prediction results for Friedman Artificial Domain dataset

FAD	SVR	ANN	DBN	ENN	Proposed
Training RMSE	0.0304	0.0350	0.0310	0.0314	**0.0300**
Testing RMSE	0.0339	0.0349	0.0320	0.0315	**0.0313**
Training MAPE	5.59%	6.75%	5.87%	5.85%	**5.51%**
Testing MAPE	6.38%	6.82%	6.14%	5.96%	**5.93%**

TABLE 4.7: Prediction results for California Housing

CH	SVR	ANN	DBN	ENN	Proposed
Training RMSE	0.1389	0.1209	0.0926	0.1111	**0.0834**
Testing RMSE	0.1637	0.1803	0.1773	0.1615	**0.1508**
Training MAPE	29.33%	27.06%	21.00%	22.46%	**17.17%**
Testing MAPE	28.36%	33.31%	32.26%	29.44%	**27.33%**

4.1.3 Summary for Ensemble DBN

In this work, we proposed an ensemble deep learning method by combining DBN and SVR. The proposed method has been evaluated with Mackey-Glass time series dataset, three electricity load demand datasets and three regression datasets. The proposed method has been compared with four benchmark methods: SVR, ANN, DBN and ensemble feedforward NN. Based on RMSE and MASE, the proposed ensemble deep learning method has outperformed the four benchmark methods for both time series and regression datasets. In addition, the proposed method has the potential ability to deal with massive and more complicated datasets.

4.2 Empirical Mode Decomposition based Ensemble Deep Belief Network

In previous section, we have proposed an ensemble deep learning algorithm for regression and time series forecasting [34], which composed of DBNs trained using different number of BP epochs and an SVR applied to analyze the relationship between these outputs and target values. It is worth noting that the input to each DBN is the original time series data. To further improve the ensemble learning architecture, in this section, we adopt the concept of "divide and conquer", and construct a novel electricity load demand forecasting method based on EMD and deep learning algorithms. The advantages of the proposed method are demonstrated on real world datasets compared with nine benchmark learning algorithms: Persistence, SVR, ANN, DBN, RF, EMD based SVR, EMD based ANN, EMD based RF, as well as the ensemble DBN proposed in the previous work.

In the proposed method, the load demand data is decomposed into several IMFs and one residue by EMD method which has been introduced in Chapter 2. A DBN composed of two RBMs and one ANN is applied to each IMF including the residue. As the completeness of the forecasting for all sub series, the prediction results can be aggregated by single learning machine or simply summed to obtain the final prediction. In addition, to avoid overfitting phenomenon, 6-fold cross-validation is applied during the training process. Figure 4.2 shows the overall schematic of this ensemble method. The procedure of this proposed method is shown as follows:

1. The time series signal is decomposed by EMD into several IMFs and one residue.
2. For each IMF and residue, we construct one training matrix as the input for one DBN.
3. Train DBN to obtain the predicted results for each of the extracted IMF and residue.
4. Combine all the prediction results by summation or with a linear neural network to formulate an ensemble output for TS.

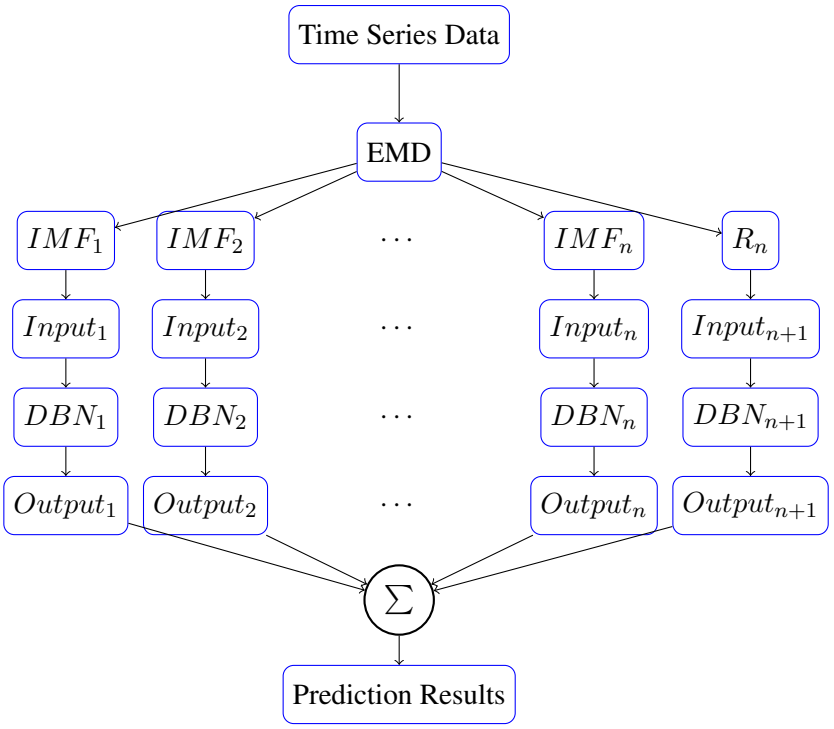


FIGURE 4.2: Schematic Diagram of the Proposed EMD based Deep Learning Approach

Figure 4.3 shows an example of the decomposed load demand TS signal with a time window of one month.

4.2.1 Experiment Setup

In this work, the performance of proposed ensemble method is evaluated by comparing with nine benchmark methods: Persistence, SVR, ANN, DBN, Ensemble DBN (EDBN), EMD based SVR model(EMD-SVR), EMD based ANN model (EMD-SLFN) and EMD based RF (EMD-RF).

The electricity load demand datasets from Australian Energy Market Operator (AEMO) were used for the comparison [160]. Especially, the datasets of year 2013 from New South Wales (NSW), Tasmania (TAS), Queensland (QLD), South Australia (SA) and Victoria (VIC) were chosen to train and test the proposed method. For each area, four months were chosen to reflect

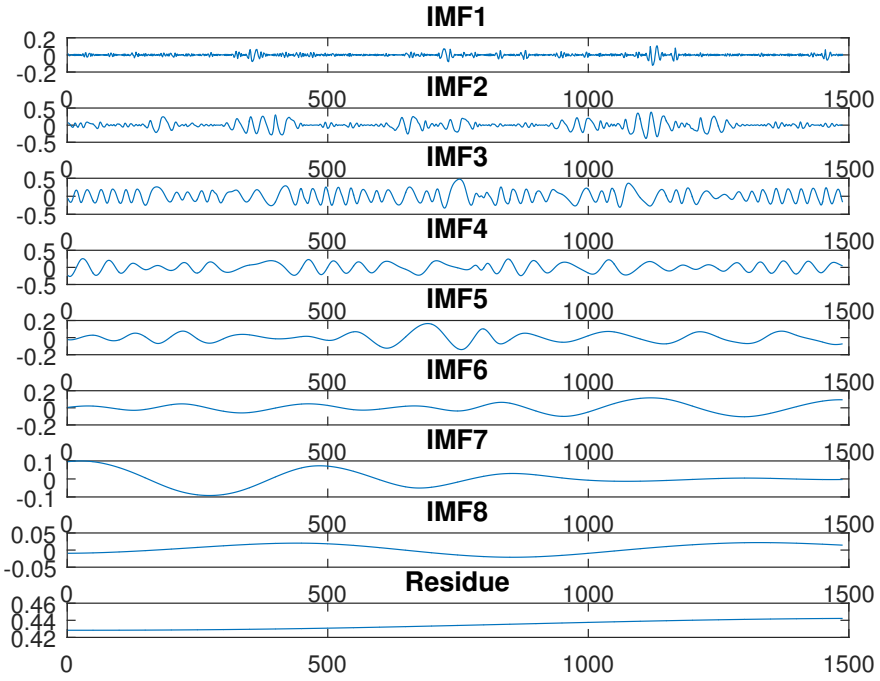


FIGURE 4.3: Example of the obtained IMF components after EMD with a time window of one month.

the factors of different seasons: January, April, July and October. During the simulation, the first three weeks was used to train the model, and the remaining one week was used for testing. Thus, there are totally 1008 examples for training and 336 examples for testing. Moreover, six fold cross-validation was employed during training to improve the generalization.

To implement the simulation, LIBSVM toolbox was used for the SVR model [155], while deep learning toolbox was used for neural networks, including ANN, DBN, EDBN [34], EMD-ANN and the proposed method [156]. RF and EMD-RF are developed from the function "TreeBagger" in Matlab. We set the parameter "NumPredictorsToSample" as one third of the number of input features to invoke RF algorithm.

For SVR and EMD based SVR, we choose the RBF kernel function with parameters chosen by a grid search. As suggested by the authors of LIBSVM toolbox, exponentially growing sequences of C and σ is used for parameter selection, where the range of C is $[2^{-4}, 2^{4}]$, and the range of σ is $[10^{-3}$,$10^{-1}]$. For ANN and EMD-ANN, the size of neural networks is determined by the size of input vector. For DBN and the proposed method, two RBMs are stacked for pre-training with the size of [100 100]. Based on grid search within the range of [200, 1000], The number of iterations for back propagation is set as 500. For RF and EMD based RF, the number of decision trees is set as 500 [49].

4.2.2 Results and Comparison

In this study, two forecasting horizons are adopted for comparison: half an hour (very short term) and one day ahead (short term).

4.2.2.1 Performance Comparison for Half-an-hour ahead Load Forecasting

The simplest forecasting method is persistence method, which assumes that the conditions at the future time of forecast are the same as the current values. The persistence method works well for very short term load demand forecasting since the temperature and human factors change little during a short time period. Therefore, persistence method can be treated as a baseline for evaluating the effectiveness of machine learning models. The one step ahead (half an hour) forecasting results of persistence method are shown in Table 4.8. We can see that all of the machine-learning algorithms outperform the persistence method for half an hour ahead forecasting.

The original load demand time series data was modeled by SVR, SLFN and RF without decomposition to reveal the advantages of EMD based hybrid approach. Comparing the forecasting results listed in Table 4.8, we can conclude that the EMD based hybrid approach generally outperforms the single structure machine learning algorithms most of the time. Moreover, EMD-SVR, EMD-SLFN and EMD-RF model has comparable performance with each other. In addition, the proposed EMD-DBN model has the best performance for half-an-hour ahead forecasting in most cases.

The comparison results of Nemenyi test among all the learning methods based on RMSE and MAPE are shown in Figure 4.4. The Nemenyi test is a post-hoc test which is used when all classifiers are compared to each other [159]. As shown, the methods with better ranks are at the top whereas the methods with worse ranks are at the bottom. The methods within a vertical line whose length is less than or equal to a critical distance have statistically the same performance. The title of the graphs shows Friedman p-value. If it is smaller than 0.05, there exists significant difference among these models. The critical difference is calculated by:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{4.1}$$

where k is the number of algorithms, N is the number of data sets, and q_α is the critical value based on the studentized range statistic divided by $\sqrt{2}$ [159]. Therefore, we can conclude from the results of statistical testing that our proposed method has the best rank and significantly outperforms the non-ensemble methods with a 95% confidence. It is also worth noting that the

proposed EMD-DBN method has better rank compared with EDBN, which shows the advantages of divide and conquer concept.

TABLE 4.8: Prediction results for half-an-hour ahead load forecasting

Dataset	Month	Metrics	Prediction model									
			Persistence	SVR [27]	ANN [158]	DBN [32]	RF [43]	EDBN [34]	EMD-SVR [168]	EMD-ANN [169]	EMD-RF	Proposed
NSW	Jan	RMSE	164.02	94.24	96.66	79.16	93.36	75.42	78.56	82.11	76.10	**49.86**
		MAPE	1.64%	0.93%	0.98%	0.78%	0.88%	0.70%	0.78%	0.88%	0.76%	**0.53%**
	Apr	RMSE	248.14	162.57	140.74	70.36	142.85	134.47	114.09	87.76	120.16	**69.55**
		MAPE	2.43%	1.88%	1.26%	**0.64%**	1.18%	1.14%	1.07%	0.82%	1.10%	0.65%
	Jul	RMSE	235.66	117.87	165.42	105.63	114.83	78.09	**74.29**	81.22	120.77	75.09
		MAPE	2.31%	1.20%	1.68%	1.09%	1.20%	**0.67%**	**0.67%**	0.81%	1.22%	0.70%
	Oct	RMSE	159.98	58.26	76.64	62.58	69.06	64.36	54.58	66.00	76.58	**51.68**
		MAPE	1.65%	0.64%	0.82%	0.70%	0.75%	0.66%	0.60%	0.74%	0.82%	**0.55%**
TAS	Jan	RMSE	17.80	13.87	13.84	12.90	11.80	13.24	12.54	**11.39**	13.52	11.59
		MAPE	1.27%	1.09%	1.09%	1.01%	1.03%	1.06%	0.97%	0.78%	1.09%	**0.74%**
	Apr	RMSE	37.42	25.26	27.03	19.53	24.82	21.35	21.76	18.03	25.19	**16.23**
		MAPE	2.32%	1.49%	1.63%	1.25%	1.26%	1.21%	1.36%	1.16%	1.45%	**1.07%**
	Jul	RMSE	43.84	34.03	33.97	30.43	33.59	**22.62**	30.90	29.14	32.34	24.44
		MAPE	2.73%	2.10%	2.03%	1.76%	2.07%	**1.36%**	1.91%	1.98%	2.17%	1.54%
	Oct	RMSE	22.80	15.89	18.49	16.80	16.94	20.41	14.94	9.37	13.69	**8.81**
		MAPE	1.63%	1.09%	1.36%	1.19%	1.19%	1.34%	1.06%	0.70%	0.97%	**0.66%**
QLD	Jan	RMSE	109.39	51.31	62.97	44.95	40.30	51.03	42.12	33.88	33.34	**25.39**
		MAPE	1.50%	0.70%	0.85%	0.62%	0.54%	0.63%	0.57%	0.48%	0.44%	**0.34%**
	Apr	RMSE	137.11	71.30	65.84	48.48	57.20	60.27	51.30	56.02	54.78	**48.34**
		MAPE	1.91%	0.94%	0.93%	0.66%	0.86%	0.75%	**0.61%**	0.73%	0.81%	0.67%
	Jul	RMSE	127.23	46.07	51.79	38.45	45.52	42.00	35.53	41.48	45.39	**30.61**
		MAPE	1.92%	0.72%	0.82%	0.54%	0.85%	0.60%	0.53%	0.62%	0.69%	**0.44%**
	Oct	RMSE	110.19	57.90	63.17	61.03	61.37	55.49	54.89	46.78	48.41	**40.46**
		MAPE	1.61%	0.80%	0.92%	0.86%	0.77%	0.71%	0.75%	0.69%	0.67%	**0.56%**
VIC	Jan	RMSE	174.95	117.79	114.73	106.25	120.34	**78.58**	82.96	117.45	115.66	98.75
		MAPE	2.52%	1.54%	1.55%	1.32%	1.58%	**1.05%**	1.09%	1.56%	1.59%	1.35%
	Apr	RMSE	162.18	148.89	149.20	104.48	102.34	75.59	96.24	77.02	68.30	**64.11**
		MAPE	2.15%	1.97%	1.99%	1.49%	1.53%	0.98%	1.33%	0.99%	0.92%	**0.87%**
	Jul	RMSE	171.99	69.41	119.43	86.63	62.57	66.36	119.27	114.34	61.84	**58.70**
		MAPE	2.44%	1.01%	1.74%	1.26%	1.00%	0.91%	1.73%	1.67%	**0.88%**	**0.88%**
	Oct	RMSE	139.47	62.88	96.63	91.20	87.11	68.50	90.49	91.38	**55.19**	57.95
		MAPE	1.97%	0.92%	1.42%	1.35%	1.20%	0.89%	1.23%	1.23%	0.85%	**0.84%**
SA	Jan	RMSE	72.11	50.37	55.65	59.17	53.92	**39.87**	58.29	46.34	42.52	51.91
		MAPE	3.04%	1.92%	2.33%	2.54%	2.23%	1.69%	2.29%	1.73%	**1.55%**	1.69%
	Apr	RMSE	58.53	45.40	39.55	44.85	46.42	35.65	**26.14**	27.98	31.27	33.44
		MAPE	3.03%	1.93%	2.33%	2.54%	2.50%	1.75%	**1.37%**	1.54%	1.83%	1.89%
	Jul	RMSE	75.05	47.68	56.12	48.55	59.99	38.03	37.38	39.16	31.93	**29.18**
		MAPE	4.25%	2.34%	3.53%	3.07%	3.49%	1.98%	2.17%	2.33%	1.86%	**1.70%**
	Oct	RMSE	48.49	42.74	37.94	40.56	42.48	43.59	30.16	**25.14**	26.59	34.17
		MAPE	2.62%	1.77%	2.23%	2.17%	2.30%	1.92%	1.67%	1.69%	1.71%	**1.62%**

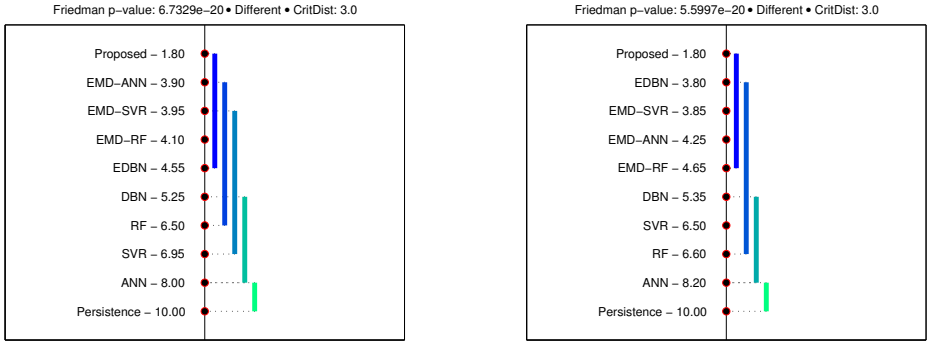


FIGURE 4.4: Nemenyi testing results for half-an-hour ahead load forecasting based on RMSE (left) and MAPE (right). The critical distance is 3.0.

4.2.2.2 Performance Comparison for One-day ahead Load Forecasting

The prediction results for one day ahead load forecasting are shown in Table 4.10. Similar to very short term load forecasting, the performance comparison includes the persistence method. In this case, the time horizon is 24 hour, which means that we assume the predicted value is

the same as the value 24 hours ago. Due to the daily seasonality of the load demand data, the accuracy of persistence method falls in an acceptable range. Therefore, the effectiveness of the involved machine learning algorithms can be verified by outperforming the persistence methods. Same as the previous experiment, the Nemenyi test is also used to compare the one-day ahead forecasting performances. The results based on RMSE and MAPE are shown in Figure 4.5. Moreover, the average computation time of all benchmark models are shown in Table 4.9.

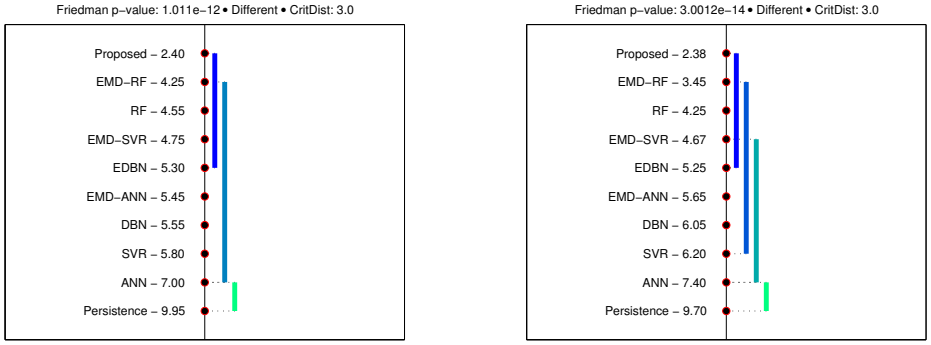


FIGURE 4.5: Nemenyi testing results for one-day ahead load forecasting based on RMSE (left) and MAPE (right). The critical distance is 3.0.

TABLE 4.9: Average computation time of electric load forecasting models

SVR	ANN	DBN	RF	EDBN	EMD-SVR	EMD-ANN	EMD-RF	Proposed
44.21s	32.91s	67.19s	23.23s	302.45s	245.98s	162.34s	112.39s	330.81s

By analyzing the forecasting outputs of SVR and ANN listed in Table 4.10, we can find that these two methods have comparable performances. This phenomenon may be due to the reason that both models have similar network structure with one hidden layer [167]. It is also worth noting that the DBN model outperforms both SVR and ANN for one day ahead forecasting and half an hour ahead load forecasting. Moreover, the EMD based hybird methods outperform the single structure models, which can confirm the advantages of EMD based ensemble algorithms. Most outstandingly, according to the Nemenyi testing rank, we can conclude that the proposed EMD-based DBN approach has successfully outperformed all the benchmark methods in both experiments on both forecasting horizons.

4.2.3 Comparative Experiments

In this section, three comparative experiments were implemented to evaluate the performance of our proposed method. The comparison conditions such as dataset partitioning and cross-validation were kept the same for the reported methods [3–5] and the proposed method.

TABLE 4.10: Prediction results for one day ahead load forecasting

Dataset	Month	Metrics	Prediction model									
			Persistence	SVR [27]	ANN [158]	DBN [32]	RF [43]	EDBN [34]	EMD-SVR [168]	EMD-ANN [169]	EMD-RF	Proposed
NSW	Jan	RMSE	978.24	703.43	750.53	639.75	**521.14**	636.03	611.20	748.30	544.17	541.53
		MAPE	8.55%	6.23%	7.2%	5.95%	**4.26%**	5.70%	5.19%	6.66%	4.54%	4.62%
	Apr	RMSE	729.50	474.38	578.05	**361.63**	500.70	551.74	569.28	512.59	495.28	377.63
		MAPE	6.71%	4.27%	5.41%	3.36%	4.25%	4.78%	5.27%	4.57%	4.21%	**3.22%**
	Jul	RMSE	609.82	574.30	534.75	415.81	387.15	414.90	402.69	345.90	353.90	**322.04**
		MAPE	6.22%	5.86%	5.38%	4.11%	4.01%	4.07%	3.95%	3.09%	3.67%	**3.08%**
	Oct	RMSE	587.14	393.32	345.07	350.82	296.53	334.12	**272.01**	299.34	333.82	282.34
		MAPE	5.36%	3.74%	3.48%	3.41%	2.78%	3.14%	2.76%	2.90%	3.17%	**2.71%**
TAS	Jan	RMSE	89.82	60.97	69.92	63.96	65.90	60.68	61.73	63.38	58.51	**56.10**
		MAPE	7.24%	4.81%	5.42%	4.98%	4.77%	4.82%	4.49%	4.87%	4.67%	**4.05%**
	Apr	RMSE	157.73	111.89	94.40	93.81	92.64	109.78	104.59	87.41	86.61	**85.13**
		MAPE	10.22%	7.48%	6.3%	6.12%	6.10%	7.28%	6.87%	5.92%	**5.80%**	**5.80%**
	Jul	RMSE	120.47	90.99	89.17	87.30	90.48	85.19	92.54	82.92	81.34	**73.91**
		MAPE	8.11%	5.89%	6.28%	6.04%	6.17%	6.04	6.09%	5.50%	5.54%	**4.93%**
	Oct	RMSE	109.46	79.45	72.86	75.73	69.80	80.81	82.85	80.85	73.86	**68.26**
		MAPE	7.48%	5.55%	5.24%	5.15%	**4.63%**	5.05	5.60%	5.63%	4.88%	4.75%
QLD	Jan	RMSE	461.09	282.07	299.32	228.86	195.85	218.55	196.20	273.70	**178.63**	191.22
		MAPE	5.25%	3.65%	3.61%	2.78%	2.41%	2.69%	2.56%	3.28%	**2.21%**	2.56%
	Apr	RMSE	489.63	266.39	339.93	247.56	231.01	259.34	264.00	237.58	**201.74**	243.68
		MAPE	6.25%	3.53%	3.77%	2.99%	2.78%	3.33%	3.47%	3.11%	**2.44%**	2.93%
	Jul	RMSE	430.46	223.17	203.00	213.20	156.08	159.45	164.68	174.64	150.01	**142.84**
		MAPE	5.90%	3.10%	3.03%	2.95%	2.32%	2.32%	2.46%	2.45%	2.29%	**2.08%**
	Oct	RMSE	417.33	298.76	263.12	251.34	236.50	292.93	**218.71**	248.55	260.94	219.19
		MAPE	5.54%	3.93%	3.46%	3.40%	2.88%	3.53%	**2.82%**	3.27%	3.15%	2.88%
VIC	Jan	RMSE	990.74	**587.98**	811.43	915.21	739.65	762.16	806.29	781.17	783.58	762.57
		MAPE	9.48%	**7.16%**	9.32%	8.79%	8.77%	9.14%	9.48%	9.07%	9.32%	8.86%
	Apr	RMSE	669.87	330.93	359.03	353.02	366.16	343.18	363.50	376.12	393.63	**321.59**
		MAPE	8.40%	4.43%	4.95%	4.55%	4.65%	4.49%	4.67%	4.79%	5.04%	**4.35%**
	Jul	RMSE	721.85	297.07	305.88	**276.25**	302.15	285.14	298.12	386.64	300.65	285.45
		MAPE	9.76%	4.38%	4.29%	**3.72%**	4.29%	3.65%	4.27%	5.29%	4.26%	3.83%
	Oct	RMSE	577.70	391.11	347.91	389.06	364.32	401.02	309.63	332.44	344.06	**322.91**
		MAPE	8.30%	4.50%	4.79%	4.85%	4.16%	4.72%	3.78%	4.15%	3.92%	**3.73%**
SA	Jan	RMSE	433.57	337.10	411.66	401.25	349.87	363.49	280.70	397.66	288.85	**238.09**
		MAPE	14.32%	13.34%	13.72%	13.62%	13.41%	14.43%	11.13%	13.80%	13.03%	**10.46%**
	Apr	RMSE	180.20	124.43	119.4	**117.61**	127.90	105.39	121.60	126.78	124.60	125.31
		MAPE	9.36%	6.71%	**6.42%**	6.67%	6.88%	6.56%	6.78%	6.87%	6.65%	6.76%
	Jul	RMSE	289.94	150.84	151.06	148.23	154.67	148.55	**141.78**	153.22	161.71	160.82
		MAPE	16.84%	8.54%	8.66%	8.50%	8.95%	8.59%	**8.48%**	9.53%	9.12%	9.60%
	Oct	RMSE	240.53	210.72	233.48	204.16	218.30	203.53	203.38	199.77	209.70	**192.74**
		MAPE	11.54%	8.94%	10.03%	9.33%	9.11%	9.32%	8.39%	8.54%	8.22%	**8.11%**

Comparative Experiment with SRSVRCABC Model

The first experiment uses historical monthly electric load demand data of Northeast China to compare with two benchmark methods: seasonal recurrent SVR with chaotic artificial bee colony (SRSVRCABC) model in [3] and TF-ε-SVR-SA model in [4]. According to Hong's paper, there are totally 64 monthly electric load data points from January 2004 to April 2009, which are divided into three parts: the training set (32 data points, December 2004 to July 2007), the validation data set (14 data points, August 2007 to September 2008), and the testing data sets (7 data points, from October 2008 to April 2009). Moreover, based on the same comparison conditions, 25 data points are fed in as input matrix to predict the following monthly load data.

Table 4.11 shows the actual values and the forecast values obtained using all benchmark methods. Obviously, the proposed method has the smallest MAPE values compared with ARIMA, TF-ε-SVR-SA and SRSVRCABC models.

TABLE 4.11: Forecasting results for monthly electric load demand in Northeastern China as used in [3, 4]

Time point	Actual	ARIMA	TF-ε-SVR-SA [4]	SRSVRCABC [3]	Proposed
October 2008	181.07	192.9316	184.5035	178.4199	181.1451
November 2008	180.56	191.127	190.3608	188.3091	182.4483
December 2008	189.03	189.9155	202.9795	195.3528	184.2608
January 2009	182.07	191.9947	195.7532	187.0825	184.1379
February 2009	167.35	189.9398	167.5795	166.1220	178.7636
March 2009	189.30	183.9876	185.9358	185.1950	184.8475
April 2009	175.84	189.3480	180.1648	179.5335	175.7244
MAPE(%)		6.044	3.799	2.387	**1.998**

Comparative Experiment with AFCM

In the second comparative experiment, the electricity load demand data of May 2007 from New South Wales, Australia is used for the simulations. According to [5], this experiment is divided into two parts: one part with small sample size, and another part with large sample size. In the first part, the data set contains the load demand data from 00:00 on May 2 to 23:30 on May 8 with the same interval of 30 min. The data set is divided into two parts: one is the training set which contains the historical data of first six days; another one is the testing set which contains the remaining one day's data. In the second part with large sample size, 1104 data points from May 2 to May 24 are used to train the model to predict the load demand in the following one week from May 18 to May 24. The adaptive fuzzy combination model (AFCM) from [5] along with SVR are implemented to compare with the proposed model.

TABLE 4.12: Forecasting results for electric load demand in New South Wales in 2007 [5]

Sample Size	Metric	SVR	AFCM [5]	Proposed
Small	MAPE	1.3678%	0.9905%	**0.6695%**
	RMSE	145.865	125.323	**83.570**
Large	MAPE	1.6580%	1.2325%	**0.9187%**
	RMSE	181.617	158.754	**118.492**

From the forecasting results listed in Table 4.12, some observations can be made from comparison. First of all, all the learning methods are effective for short time load demand forecasting since all of them can give reasonable results. Second, by comparing the differences between small and large sample size parts, the proposed EMD-DBN method can reduce the influence caused by redundant information in the large size data set to give better performance. Finally, it is clear that the EMD based deep learning method has outperformed the benchmark methods in both experiments.

Comparative Experiment with PSF-NN

For the third comparative experiment, according to [6], we use electricity load demand data for the state of NSW in Australia for three years: 2009, 2010 and 2011. The data from the first two years is used to train the prediction model, while the remaining data for 2011 is used for testing. The forecasting horizon is still one day. There are four benchmark methods, the pattern sequence-based forecasting (PSF) method and three combined PSF-NN models with three different feature sets. Table 4.13 shows the comparative results. The proposed EMD based deep learning approach outperforms all the benchmark methods on both error measures.

TABLE 4.13: Comparative Results with PSF-NNs [6]

Prediction method	MAE[MW]	MAPE[%]
Proposed	**266.58**	**3.00**
PSF	352.03	3.96
PSF-NN1	311.94	3.44
PSF-NN2	402.13	4.51
PSF-NN3	300.77	3.37

4.2.4 Summary for EMD-DBN

In this section, we present an ensemble deep learning method based on EMD and DBN. The proposed method has been evaluated with three electricity load demand datasets from AEMO. Nine benchmark methods have been compared to verify the effectiveness of the proposed method: Persistence, SVR, ANN, DBN, RF, EDBN, EMD-SVR, EMD-SLFN and EMD-RF. Two error measures (RMSE and MAPE) were used to evaluate the performance of these prediction models. Moreover, two comparative experiments are also implemented to verify the effectiveness of the proposed method. According to the prediction results, several observations can be concluded:

1. EMD based hybrid methods normally outperform the corresponding single structure models for load demand time series forecasting.

2. Deep learning algorithms show their advantages in dealing with nonlinear features when the forecasting horizon increases.

3. Random Forests, as a decision tree based method, is effective for load demand forecasting with the advantage of fast training.

4. The proposed EMD based ensemble deep learning approach has the best performance according to the statistical testing.

Chapter 5

Ensemble Incremental Learning Random Vector Functional Link Network for Short-term Electric Load Forecasting

As we have discussed in Chapter 1, incremental learning model can update itself based on the new training samples, therefore it requires high efficiency. In Chapter 4, the ensemble deep learning models achieve excellent performance based on the error measurements. However, they have complicated network structure which needs a relatively long period of training time. In this chapter, an ensemble incremental learning method based on DWT, EMD and RVFL is presented for electric load forecasting, which has advantages on both accuracy and efficiency [78].

5.1 Incremental Learning with RVFL

Incremental learning RVFL is suitable for real time applications since the learning model needs to be updated whenever the new input patterns are available. As introduced in [170], by taking the pseudoinverse of a partitioned matrix, the stepwise updating of the weight in RVFL can be achieved easily due to the advantages of flat structure of RVFL.

Denote $\mathbf{A}_n$ as the $n \times m$ pattern matrix representing the input matrix consisting of all input vectors combined with enhancement components. a' is the $m \times 1$ new sample pattern which

should be added to update the RVFL. Thus, the new pattern matrix A_{n+1} is shown as:

$$A_{n+1} \triangleq \begin{bmatrix} A_n \\ a' \end{bmatrix} \tag{5.1}$$

where n is the number of samples, as well as the discrete time instance, m is the dimension of the input pattern. Therefore, the pseudoinverse of A_{n+1} can be updated based only on the pseudoinverse of A_n and the imported new input vector a'. The precedure of calculation is shown as follows:

$$A_{n+1}^{+} = [A_n^{+} - db'|d] \tag{5.2}$$

where

$$\begin{aligned} b' &= a'A_n^{+} \\ d &= (1 + b'b)^{-1}A_n^{+}b \end{aligned} \tag{5.3}$$

Same with the pattern matrix A_n, we denote the output vector as Y_n, weight matrix as W_n, and the new output as y'. According to the least square solution, we can obtain:

$$\begin{aligned} W_{n+1} &= A_{n+1}^{+}Y_{n+1} \\ W_n &= A_n^{+}Y_n \end{aligned} \tag{5.4}$$

Therefore, the new weight matrix W_{n+1} can be calculated by:

$$W_{n+1} = W_n + (y' - a'W_n)d \tag{5.5}$$

where d is the obtained optimal learning rate of weight updating.

5.2 Proposed incremental RVFL based ensemble method

EMD, as an efficient ensemble method to perform TS signal decomposition, has a major drawback called mode mixing problem: "one IMF may consist of signal spanning a wide band of frequency, or more than one IMFs contain signals in a similar frequency band" [171]. In the literature, this problem is normally solved by ensemble of EMD (EEMD) [172], which works by applying EMD to uncorrelated Gaussian noise added TS signal repetitively and combining the results to remove the noise. In this chapter, the MODWT is employed to deal with the frequency issue, followed by EMD to perform better decomposition. Then an RVFL network is trained for each IMF and residue. The outputs of all sub-series are combined and analyzed by another RVFL to formulate the final prediction results. Because of high efficiency of RVFL model, the computation speed of the whole ensemble model can be ensured. Figure 5.1 is the schematic

diagram of the pre-training phase of this proposed ensemble method, whose procedures can be concluded as follows:

1. Apply MODWT to decompose the original TS into several frequency components.
2. Apply EMD to decompose each frequency component into several IMFs and one residue.
3. Construct the training matrix as the input of each RVFL network for each obtained sub-series. Then train an RVFL network for each of the extracted IMF and residue.
4. Construct a new input matrix by combining three aspects: the prediction results of all the sub-series, the original TS signal and the temperature data. Then an incremental RVFL is trained using this new training matrix to formulate the final prediction results.

Incremental learning allows the learning model updating itself whenever the new input patterns are available, which guarantee the effectiveness of the proposed method for long term application. Therefore, after the pre-training phase, the learned ensemble model including the pseudoinverse of A_n and the weight matrix W_n for the incremental RVFL can be updated by the incremental learning phase:

1. When a new input sample is given to the network, MODWT and EMD can be applied to decompose the signal and obtain new input pattern A_{n+1} by combining all the new outputs from all sub-series. Then the new weight matrix W_{n+1} can be updated using equation 5.5.
2. The validation data is applied to check the error level. If the error decreases, we keep the updates, otherwise the weight matrix is not changed.
3. Repeat steps 1 and 2 whenever new samples are presented to the network.

5.3 Experiment Setup

5.3.1 Datasets

In this work, the electric load datasets from Australian Energy Market Operator (AEMO) [173] were used for evaluating the performance of benchmark learning models. There are totally fifteen electric load datasets of year $2013 \sim 2015$ from five states of Australia: New South Wales (NSW), Tasmania (TAS), Queensland (QLD), South Australia (SA) and Victoria (VIC). For each dataset, the first nine months were used for training, while the remaining three months were used for testing. The statistics of the datasets are summarized in Table 5.1.

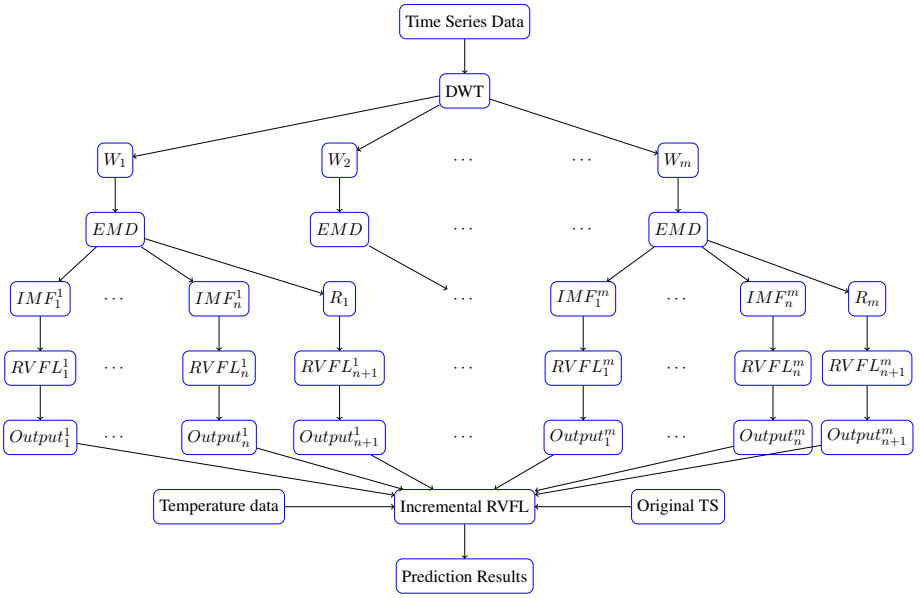


FIGURE 5.1: Schematic Diagram of the Proposed DWT-EMD based Incremental RVFL Network

The electric load data is sampled every half an hour, therefore 48 data points are recorded for one day. To identify cycles and patterns in load TS, we employ autocorrelation function (ACF) as a guidance for informative feature subset selection. Suppose a time series data set is given as $X = \{X_t : t \in T\}$, where T is the index set. The lag k autocorrelation coefficient r_k can be computed by:

$$r_k = r(X_t, X_{t-k}) = \frac{\sum_{t=k+1}^{n}(X_t - \bar{X})(X_{t-k} - \bar{X})}{\sum_{t=1}^{n}(X_t - \bar{X})^2} \tag{5.6}$$

where $\bar{X}$ is the mean value of all X in the given time series, r_k measures the linear correlation of TS signal at times t and $t-k$.

As shown in Chapter 3, three strongest dependent lag variables can be identified: the value of previous half-an-hour (X_{t-1}), the value at the same time in the previous week (X_{t-336}), as well as the value at the same time in the previous day (X_{t-48}). Therefore, in order to take all the most informative lag variables into consideration, we select the data points of the whole previous day (X_{t-48} to X_{t-96}) and the same day in the previous week (X_{t-336} to X_{t-384}) to construct the input feature set for one day ahead load forecasting in this work.

After DWT-EMD decomposition, we get $m \times n$ IMFs and m residues. Let's mark the IMFs and residues as $\mathbf{X}^{i,j}$, where $1 \leqslant i \leqslant m, 1 \leqslant j \leqslant n+1$. For each IMF and residue, when we want to predict the value at time t (marked as $X_t^{i,j}$), the corresponding input features include the data points of the whole previous day ($X_{t-48}^{i,j}$ to $X_{t-96}^{i,j}$) and the same day in the previous week ($X_{t-336}^{i,j}$ to $X_{t-384}^{i,j}$). This is the content of the input matrix for the RVFLs analyzing the sub-series. As a result, the predicted values of all the sub-series for time t are:

$$\{\hat{X}_t^{i,j} : 1 \leqslant i \leqslant m, 1 \leqslant j \leqslant n+1\} \tag{5.7}$$

As mentioned above, to construct the input matrix for the incremental RVFL, we combine the predicted values with original input signal (X_{t-48} to X_{t-96}, and X_{t-336} to X_{t-384}), and the recently observed temperature data (T_{t-1}).

TABLE 5.1: Summary of AEMO load datasets

Dataset	Year	Length	Min	Median	Mean	Max	Std
QLD	2013	17520	4148.7	5752.1	5703.7	8278.4	747.0
	2014	17520	4073.0	5726.0	5745.7	8445.3	794.0
	2015	17520	4281.4	6005.6	6035.4	8808.7	777.2
NSW	2013	17520	5113.0	8045.0	7981.6	13788	1190.9
	2014	17520	5138.1	7987.4	7917.8	11846	1170.1
	2015	17520	5337.4	7990.4	7979.8	12602	1232.7
TAS	2013	17520	659.5	1109.0	1129.3	1650.3	142.3
	2014	17520	569.1	1088.7	1109.7	1630.1	139.0
	2015	17520	479.4	1112.3	1138.2	1667.2	145.3
SA	2013	17520	728.6	1389.3	1426.6	2991.3	301.7
	2014	17520	682.5	1360.8	1403.3	3245.9	312.8
	2015	17520	696.3	1352.7	1398.5	2870.4	306.0
VIC	2013	17520	3551.6	5458.1	5511.8	9587.5	895.9
	2014	17520	3272.9	5307.8	5324.4	10240	921.4
	2015	17520	3369.1	5186.5	5194.6	8579.9	864.7

Temperature data of above five states in Australia is also considered for electric load forecasting in this study, which is provided by Australian Bureau of Meteorology [174]. Specifically, the temperature data from high demand areas in each state is considered: Sydney and Canberra for NSW, Horbat and Launceston airport for TAS, Brisbane for QLD, Adelaide for SA, and Melbourne for VIC. The daily maximum temperature and daily minimum temperature datasets are used as additional features to help improve the performance of the proposed electric load forecasting method. The influence of temperature data will be discussed in Section 5.4.3.

5.3.2 Variations of RVFL network

There are eight different RVFL network configurations by varying the network components: input layer and hidden layer biases, along with input-output connections (functional links). In [54, 56], the authors compared the performance of all the variations of RVFL network for both regression and classification problems. According to the conclusions made, the direct input-output connections can improve the performance of RVFL network significantly. Moreover, although the input and hidden layer biases have little influence on the performance, it is still necessary to retain biases to ensure the neural networks function properly as a universal approximator. Hence, in this work, the RVFL network with direct input-output connections and input/hidden layer biases was selected for our proposed EMD-RVFL network. Moreover,

RWSLFN (the RVFL variant without functional links) is also included to perform comparison and verify the results reported in [54, 56].

5.3.3 Methodology

To implement the simulation, the deep learning toolbox was used for neural networks, including SLFN and EMD based SLFN (EMD-SLFN) [156]. RF and EMD-RF were developed from the function "TreeBagger" in Matlab. We set the parameter "NumPredictorsToSample" as one third of the number of input features to invoke RF algorithm. RVFL, EMD-RVFL and the proposed incremental DWT-EMD-RVFL were developed by the authors in Matlab based on the work in [56].

For SLFN and EMD-SLFN, the size of neural networks is determined by the size of input vector. Based on the experience in our previous work [175], the number of iterations for back propagation is set as 500 to avoid overfitting. For RF and EMD based RF, because of the same reasons as above, the number of decision trees is set as 500. For RVFL, EMD-RVFL and the proposed method, according to suggestion in [54, 56], the randomization used a uniform distribution in $[-1, 1]$, the number of hidden neurons is selected over $1000 : 10000$ with a step-size of 1000.

5.3.4 Error Measurement

There are two error measures being used to evaluate the performance of learning models in this work: RMSE and MAPE. Except for above traditional error measures, readers are also suggested to try some new error measures. For example, in [176], a new error measure is proposed. It finds a restricted permutation of the predicted value to minimize the point-wise error, which may be able to reduce the so-called "double penalty" effect.

5.4 Results and Discussion

5.4.1 Effect of Functional Links and Number of Hidden Neurons

We focus on the effect of the direct connections in RVFL network in this section. In order to make a reasonable comparison, the other issues (e.g. parameters, activation functions, range of randomization, etc.) in RVFL with and without functional links (RWSLFN) were set the same, except for the number of hidden neurons. Table 5.2 shows the prediction results using RVFL and RWSLFN with different number of hidden neurons. The row "win-tie-lose" in the bottom means the number of times that RVFL wins, ties, and loses to RWSLFN, respectively. Several observations can be concluded by this comparison:

1. The number of hidden neurons have important influence on the performance of RVFL variants. Sufficient number of hidden neurons can ensure the completeness of information obtained.

2. The superiority of functional links can also be observed, which may due to the reason that the direct links can serve as a regularization for the randomization [54].

3. However, the advantage of functional links decreases as the number of hidden neurons increasing, which means that the functional links and number of hidden neurons complement each other for information gaining.

Therefore, taking model complexity into consideration, RVFL with direct links and reasonable number of hidden neurons is recommended.

In this study, several hundreds or thousands hidden neurons are sufficient for load forecasting. However, for practical application, the number of hidden neurons should be chosen based on the characteristics of different datasets, as well as the requirement of accuracy and efficiency. We recommend readers try different RVFL models with various numbers of hidden neurons, and choose the most suitable one according to the actual requirements.

TABLE 5.2: Performance comparison between RVFL variants with and without direct links

Dataset	Year	Metrics	Number of hidden neurons					
			100		1000		10000	
			+link	-link	+link	-link	+link	-link
QLD	2013	RMSE	**325.761**	330.520	**318.559**	318.747	**314.144**	314.150
		MAPE	**4.069%**	4.150%	3.968%	**3.967%**	3.926%	3.926%
	2014	RMSE	**414.871**	424.211	403.132	**402.781**	396.435	**396.394**
		MAPE	**4.989%**	5.125%	4.810%	**4.804%**	4.735%	**4.734%**
	2015	RMSE	**378.435**	384.532	368.445	**368.360**	**362.576**	362.613
		MAPE	**4.461%**	4.549%	4.309%	**4.307%**	**4.249%**	4.250%
NSW	2013	RMSE	**641.722**	648.763	**627.046**	627.223	**617.595**	617.609
		MAPE	**6.605%**	6.707%	**6.318%**	6.323%	**6.198%**	6.199%
	2014	RMSE	**624.748**	627.883	621.408	**620.368**	624.098	**624.058**
		MAPE	**6.159%**	6.237%	6.067%	**6.054%**	6.062%	6.062%
	2015	RMSE	728.598	**725.640**	715.637	**712.831**	711.597	**711.240**
		MAPE	**6.861%**	6.927%	6.666%	**6.639%**	6.629%	**6.626%**
TAS	2013	RMSE	**75.911**	77.032	**75.589**	75.744	**74.974**	74.992
		MAPE	**5.509%**	5.618%	**5.469%**	5.484%	**5.395%**	5.397%
	2014	RMSE	**72.788**	72.957	**72.219**	73.725	73.115	**72.232**
		MAPE	**5.507%**	5.588%	**5.505%**	5.518%	**5.450%**	5.451%
	2015	RMSE	**69.146**	70.172	**69.717**	69.917	**68.882**	68.908
		MAPE	**4.934%**	5.029%	**4.969%**	4.985%	**4.894%**	4.896%
SA	2013	RMSE	**198.108**	200.202	**194.991**	195.031	**193.207**	193.230
		MAPE	**11.961%**	12.099%	**11.759%**	11.763%	**11.652%**	11.653%
	2014	RMSE	**177.654**	178.835	**177.390**	177.447	**177.201**	177.206
		MAPE	**11.215%**	11.316%	**11.095%**	11.098%	**11.030%**	11.031%
	2015	RMSE	**245.944**	249.006	**243.512**	243.669	243.061	**243.054**
		MAPE	**12.838%**	12.987%	**12.685%**	12.693%	12.486%	12.486%
VIC	2013	RMSE	**564.909**	568.801	**555.698**	555.947	**548.845**	548.892
		MAPE	**8.734%**	8.853%	**8.473%**	8.479%	8.314%	8.314%
	2014	RMSE	**563.379**	566.304	560.036	**559.695**	**558.722**	558.729
		MAPE	**9.735%**	9.780%	9.603%	**9.593%**	9.535%	9.535%
	2015	RMSE	**575.090**	581.875	**562.418**	562.751	**556.468**	556.501
		MAPE	**9.245%**	9.370%	**8.940%**	8.946%	**8.816%**	8.817%
Win-tie-lose			**29-0-1**		**19-0-11**		**18-5-7**	

5.4.2 Effect of Incremental Learning

In this part, we concentrate on the effectiveness of incremental learning. As we have mentioned above, for each dataset, the first nine months are used for training, and the last three months are used for testing. For incremental learning, the data points in testing subset are imported to the network one by one, thereby stepwise updating the model obtained in training phase. Meanwhile, for non-incremental learning, the model is fixed during testing phase. The performance comparison of incremental and non-incremental learning is shown in Table 5.3. All the models have 100 hidden neurons with functional links. Same with section 5.4.1, the row "win-tie-lose" in the bottom means the number of times that incremental learning wins, ties, and loses to non-incremental learning, respectively. From the comparison results "$30-0-0$", we can conclude that incremental learning is definitely beneficial for short term electric load TS forecasting with RVFL and its ensemble models. Moreover, it also worth noting that the proposed DWT-EMD-RVFL outperforms RVFL and EMD-RVFL in every case. The statistical testing is conducted and discussed in section 5.4.3.

TABLE 5.3: Performance comparison between incremental and non-incremental learning. I stands for incremental, and N stands for non-incremental.

Dataset	Year	Metrics	Learning Models					
			RVFL		EMD-RVFL		DWT-EMD-RVFL	
			I	N	I	N	I	N
QLD	2013	RMSE	**318.217**	325.761	**278.983**	290.260	**267.111**	285.491
		MAPE	**3.968%**	4.069%	**3.580%**	3.733%	**3.326%**	3.545%
	2014	RMSE	**394.353**	414.871	**322.909**	343.685	**328.702**	356.075
		MAPE	**4.778%**	4.989%	**3.988%**	4.193%	**3.932%**	4.238%
	2015	RMSE	**370.099**	378.435	**353.267**	366.844	**308.760**	325.967
		MAPE	**4.331%**	4.461%	**4.150%**	4.325%	**3.656%**	3.861%
NSW	2013	RMSE	**614.786**	641.722	**513.276**	538.169	**516.216**	530.103
		MAPE	**6.228%**	6.605%	**5.177%**	5.497%	**5.075%**	5.539%
	2014	RMSE	**604.592**	624.748	**520.422**	552.439	**496.081**	528.685
		MAPE	**5.843%**	6.159%	**5.136%**	5.484%	**4.674%**	4.967%
	2015	RMSE	**674.971**	728.598	**648.562**	697.413	**553.518**	626.850
		MAPE	**6.311%**	6.861%	**6.031%**	6.482%	**5.060%**	5.681%
TAS	2013	RMSE	**74.955**	75.911	**65.979**	67.173	**62.569**	65.430
		MAPE	**5.440%**	5.509%	**4.744%**	4.841%	**4.535%**	4.739%
	2014	RMSE	**70.873**	72.788	**67.803**	69.545	**61.880**	63.719
		MAPE	**5.346%**	5.507%	**5.068%**	5.217%	**4.608%**	4.755%
	2015	RMSE	**67.156**	69.146	**66.033**	68.196	**61.610**	63.646
		MAPE	**4.767%**	4.934%	**4.736%**	4.911%	**4.371%**	4.531%
SA	2013	RMSE	**191.545**	198.108	**186.902**	195.089	**163.689**	171.889
		MAPE	**11.405%**	11.961%	**11.068%**	11.715%	**9.686%**	10.179%
	2014	RMSE	**173.237**	177.654	**169.683**	174.251	**151.902**	156.496
		MAPE	**10.815%**	11.215%	**10.547%**	10.945%	**9.373%**	9.712%
	2015	RMSE	**241.125**	245.944	**235.117**	240.803	**221.009**	241.887
		MAPE	**12.347%**	12.838%	**12.024%**	12.536%	**11.663%**	12.817%
VIC	2013	RMSE	**535.626**	564.909	**511.024**	538.539	**437.622**	469.076
		MAPE	**8.104%**	8.734%	**7.550%**	7.989%	**6.321%**	6.888%
	2014	RMSE	**522.461**	563.379	**469.280**	506.752	**400.502**	428.211
		MAPE	**8.773%**	9.735%	**7.752%**	8.635%	**6.426%**	6.982%
	2015	RMSE	**555.641**	575.090	**541.444**	562.414	**486.335**	516.758
		MAPE	**8.779%**	9.245%	**8.549%**	9.022%	**7.587%**	8.022%
Win-tie-lose			**30-0-0**		**30-0-0**		**30-0-0**	

5.4.3 Performance Comparison with Benchmarks

In this section, the performance of the proposed incremental DWT-EMD-RVFL approach is evaluated by comparing with several benchmark methods. First of all, the persistence method, which is the simplest forecasting method, is employed as the baseline for comparing the performance of learning models in this work. For persistence method, the load value at the same hour of last day is used as the forecast for each of the 24 hours of next day, which works well because of the highly periodic characteristic of electric load TS. Moreover, a modified version of another benchmark method GLMLF-B (General Linear Model based Load Forecaster - Benchmark) [177] is also employed, which can offer a higher baseline for the other machine learning algorithms. In this study, since the forecasting horizon is one day (or 48 steps), to predict X_t, we use the corresponding hour, day, month, as well as the load value X_{t-48} at the same time in the previous day, as input features to construct a multiple linear regression (MLR) model, which shares similar ideas with GLMLF-B. Except the persistence method, all the other benchmark models have made use of the temperature data, which can improve the performance of forecasting.

The prediction results for one-day-ahead electric load forecasting are shown in Table 5.4. The numbers in bold mean that the corresponding method achieves the best performance for this dataset under this performance measurement. The prediction results generated by the proposed method without temperature data are also recorded, which can be compared with the results from the proposed method with temperature data. From Table 5.4, we can clearly see that the proposed method achieves the best performance for every case except QLD. The MAPE values of the load prediction for QLD are significantly lower than the ones for all the other regions. This phenomenon proves the fact that the pattern of the load data of QLD is much more stable and simpler than the patterns of the load data of other regions, which is easier for the benchmarks to analyze. Therefore, our proposed method cannot show much advantage on the QLD datasets.

Moreover, statistical tests are employed to give a detail analysis about the performance differences among all the learning models. Same as previous sections, Friedman test plus Nemenyi post-hoc test are applied. The comparison results of statistical test based on RMSE and MAPE are shown in Figure 5.2 and Figure 5.3, respectively.

From the results of simulations and statistical tests, several conclusions can be made:

1. The original load TS data was modeled by SLFN, RF and RVFL without decomposition. Therefore, the advantages of EMD based ensemble methods can be revealed by performing comparisons.

2. By employing incremental learning, RVFL based models have comparable (or even better) performance with traditional NN and RF based models with less computation time.

TABLE 5.4: Prediction results for one-day-ahead electric load forecasting

Dataset	Year	Metrics	Prediction model									
			Persistence	GLMLF-B [177]	SLFN [158]	RF [43]	RVFL [52, 53]	EMD-SLFN [169]	EMD-RF [178]	EMD-RVFL [179]	Proposed -T	Proposed +T
QLD	2013	RMSE	492.589	355.503	307.892	278.511	281.583	230.600	266.686	244.820	233.221	**218.329**
		MAPE	6.348%	4.323%	3.912%	3.449%	3.518%	2.953%	3.273%	3.160%	2.967%	**2.797%**
	2014	RMSE	588.706	399.927	373.318	334.405	342.874	**256.671**	315.772	270.137	284.766	263.873
		MAPE	7.144%	5.073%	4.655%	4.100%	4.139%	3.243%	3.716%	3.442%	3.440%	**3.215%**
	2015	RMSE	553.077	369.409	368.554	328.787	329.461	318.157	341.356	304.317	277.861	**261.799**
		MAPE	6.633%	4.749%	4.546%	3.933%	3.835%	3.782%	3.806%	3.673%	3.341%	**3.170%**
NSW	2013	RMSE	901.511	643.155	614.006	556.797	608.919	519.295	497.665	438.721	436.890	**403.403**
		MAPE	8.657%	6.426%	6.067%	5.315%	6.139%	5.234%	4.773%	4.386%	4.240%	**4.040%**
	2014	RMSE	878.077	632.275	620.177	585.587	531.664	492.079	527.581	435.748	419.416	**380.410**
		MAPE	8.595%	6.069%	5.834%	5.486%	5.033%	4.758%	4.862%	4.357%	3.906%	**3.629%**
	2015	RMSE	1055.406	713.827	630.819	575.037	549.645	550.728	563.351	520.891	466.842	**400.443**
		MAPE	9.689%	6.669%	5.953%	5.222%	5.303%	5.289%	5.185%	5.033%	4.368%	**3.881%**
TAS	2013	RMSE	97.855	84.095	67.148	68.648	68.650	64.216	65.411	61.119	58.175	**56.725**
		MAPE	6.870%	6.042%	4.771%	5.016%	5.080%	4.823%	4.746%	4.468%	4.228%	**4.157%**
	2014	RMSE	86.863	82.193	69.768	67.517	67.683	62.987	67.395	64.239	58.721	**57.025**
		MAPE	6.512%	6.324%	5.305%	5.160%	5.219%	4.796%	5.142%	4.886%	4.361%	**4.298%**
	2015	RMSE	83.993	75.808	59.962	61.599	64.554	60.767	64.089	60.305	58.687	**55.837**
		MAPE	6.035%	5.452%	4.299%	4.414%	4.561%	4.352%	4.487%	4.353%	4.169%	**3.999%**
SA	2013	RMSE	288.779	206.009	167.367	166.099	166.461	168.514	150.116	163.474	146.936	**139.500**
		MAPE	16.022%	12.617%	10.118%	10.123%	10.377%	10.322%	9.734%	10.118%	8.740%	**8.572%**
	2014	RMSE	242.356	175.991	147.681	154.063	159.824	154.693	152.912	155.543	142.976	**133.605**
		MAPE	14.419%	11.659%	9.540%	9.798%	10.219%	9.993%	9.323%	9.956%	8.769%	**8.351%**
	2015	RMSE	365.141	244.246	196.482	173.032	183.903	176.610	225.926	177.217	192.189	**163.204**
		MAPE	18.395%	12.677%	11.168%	9.374%	10.073%	9.961%	11.938%	9.748%	10.122%	**9.086%**
VIC	2013	RMSE	781.683	584.131	560.457	487.053	537.208	439.463	477.357	451.230	376.096	**360.696**
		MAPE	10.711%	9.159%	8.531%	7.073%	8.269%	6.241%	6.895%	6.697%	5.368%	**5.241%**
	2014	RMSE	719.402	621.930	634.557	530.069	487.838	463.882	498.345	422.682	350.499	**335.468**
		MAPE	11.246%	10.079%	10.172%	8.564%	8.019%	7.807%	8.373%	6.872%	5.528%	**5.328%**
	2015	RMSE	874.691	684.131	485.400	479.175	486.965	518.785	486.611	471.469	430.092	**385.754**
		MAPE	12.886%	10.159%	7.669%	7.475%	7.260%	8.130%	7.230%	7.404%	6.661%	**6.106%**

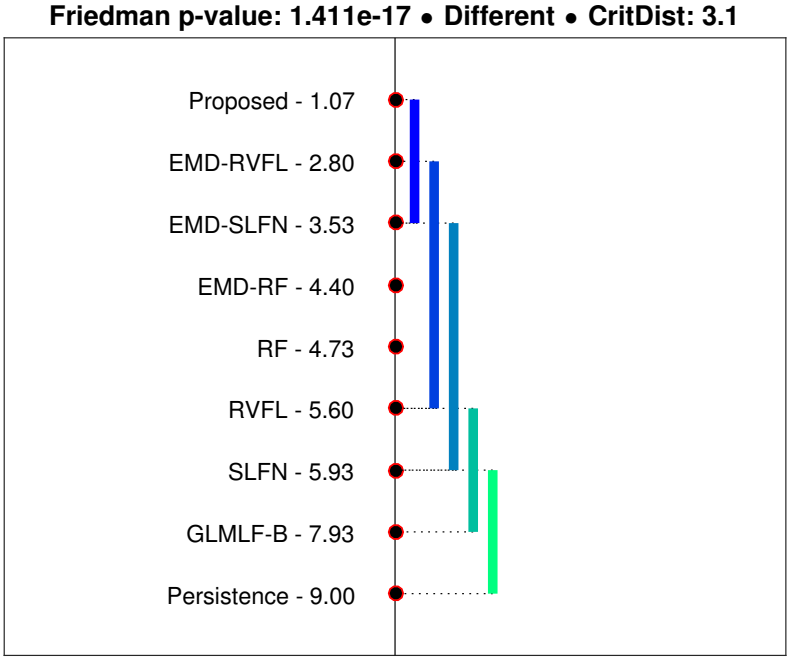


FIGURE 5.2: Nemenyi test for electric load forecasting based on RMSE. The critical distance is 3.1.

3. The proposed incremental DWT-EMD based RVFL approach achieves the best rank and significantly outperform the non-EMD based benchmarks and EMD-RF with a 95% confidence.

In order to find where in the forecast the proposed method offers a key advantage in performance,

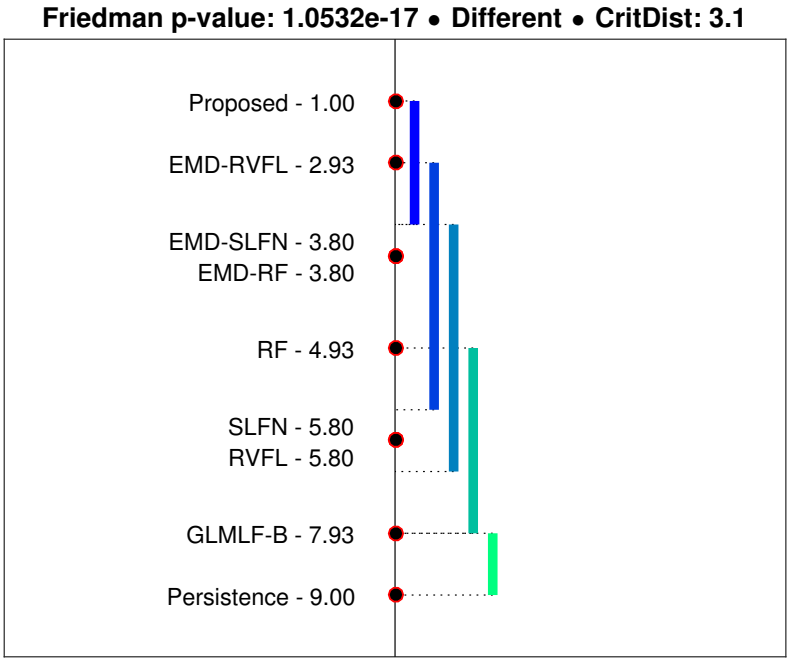


FIGURE 5.3: Nemenyi test for electric load forecasting based on MAPE. The critical distance is 3.1.

a comparison between the forecasting results for original RVFL and the proposed method was conducted. Comparisons of predicted values with actual values for the proposed method and RVFL network are shown in Figure 5.4 and Figure 5.5, respectively. This part of load data is selected from the testing dataset of NSW of the year 2013, with a time window of one week (from Sunday to Saturday).

From the comparison results, we can conclude that the key improvements caused by the proposed method are located on the data points during the weekends. In fact, the difference between the electric load in weekdays and weekends is one of the dominant challenges for prediction methods. Some published works, such as [180], deal with this problem by introducing additional input features of calendar information (e.g. holidays, weekends, etc.). However, in this work, under the help of decomposition algorithms DWT and EMD, as well as the incremental learning, the proposed method can detect the pattern changes caused by weekends, and modify the model by itself.

In our previously published paper [16], the benchmarks are evaluated using the same load datasets form AEMO of the year 2013. The prediction results for one day ahead load forecasting are shown in Table 3 in that paper. For each area, four months were chosen to reflect the factors of different seasons: January, April, July and October. To show the performance of the proposed method and the benchmark models in different seasons, the same comparison simulations are implemented in this study using load dataset from NSW of the year 2015. The

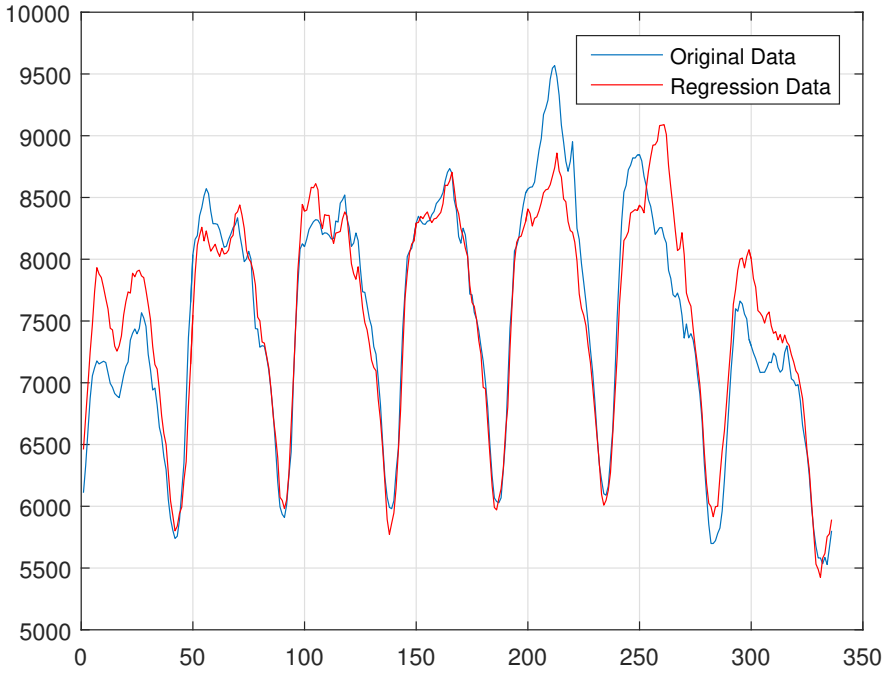


FIGURE 5.4: Comparison of predicted values with actual values for the proposed method. The y-axis represents for electric load power (MW), and each point on x-axis represents for half hour.

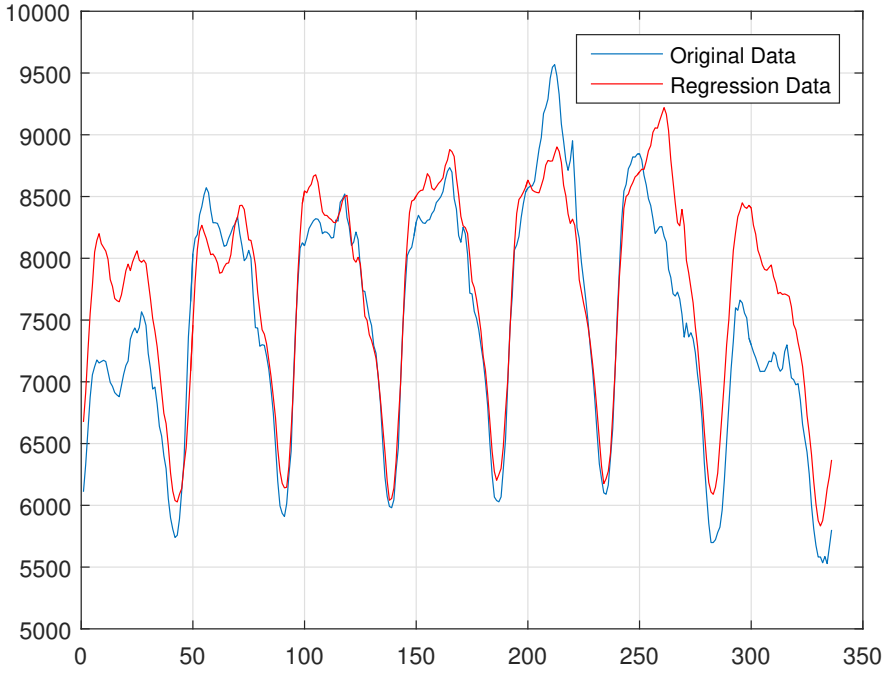


FIGURE 5.5: Comparison of predicted values with actual values for RVFL. The y-axis represents for electric load power (MW), and each point on x-axis represents for half hour.

results are shown in Table 5.5. From the results, by taking the factors of different seasons into consideration, we can tell that the benchmark methods performs relatively stable for the same dataset in different seasons. Moreover, in this case, our proposed method still outperforms all benchmarks models significantly.

TABLE 5.5: Prediction results for different seasons using the load data from NSW of the year 2015

Month	Metrics	Prediction model								
		Persistence	GLMLF-B [177]	SLFN [158]	RF [43]	RVFL [52, 53]	EMD-SLFN [169]	EMD-RF [178]	EMD-RVFL [179]	Proposed
Jan	RMSE	842.732	612.303	464.061	440.572	428.908	379.871	428.388	403.271	**193.800**
	MAPE	7.393%	5.599%	4.094%	3.527%	3.869%	3.319%	3.328%	3.423%	**1.857%**
Apr	RMSE	769.606	525.145	448.827	437.975	425.228	400.455	441.052	411.820	**212.703**
	MAPE	6.801%	5.259%	4.294%	3.992%	3.936%	4.031%	3.971%	3.861%	**2.030%**
Jul	RMSE	989.372	614.706	501.107	411.863	493.064	440.431	402.973	423.287	**296.743**
	MAPE	9.831%	6.135%	5.145%	4.290%	5.093%	4.719%	4.144%	4.423%	**2.961%**
Oct	RMSE	1620.508	1091.054	1021.127	953.213	1004.394	913.704	911.767	987.330	**659.407**
	MAPE	14.887%	9.404%	8.981%	7.139%	8.863%	7.217%	6.817%	7.035%	**5.934%**

5.4.4 Computation time comparison

The computation time of benchmark methods for electric load forecasting using the datasets of year 2015 is shown in Figure 5.6. It is easy to conclude that RVFL is much faster than ANN and RF. ANN needs to be iteratively tuned by BP algorithm to convergence to the optimal weights. RF needs to train a group of decision trees. Different from ANN and RF, RVFL has a closed form solution. Benefit from the good efficiency of RVFL, the proposed DWT-EMD-RVFL model also has a reasonable fast computation speed.

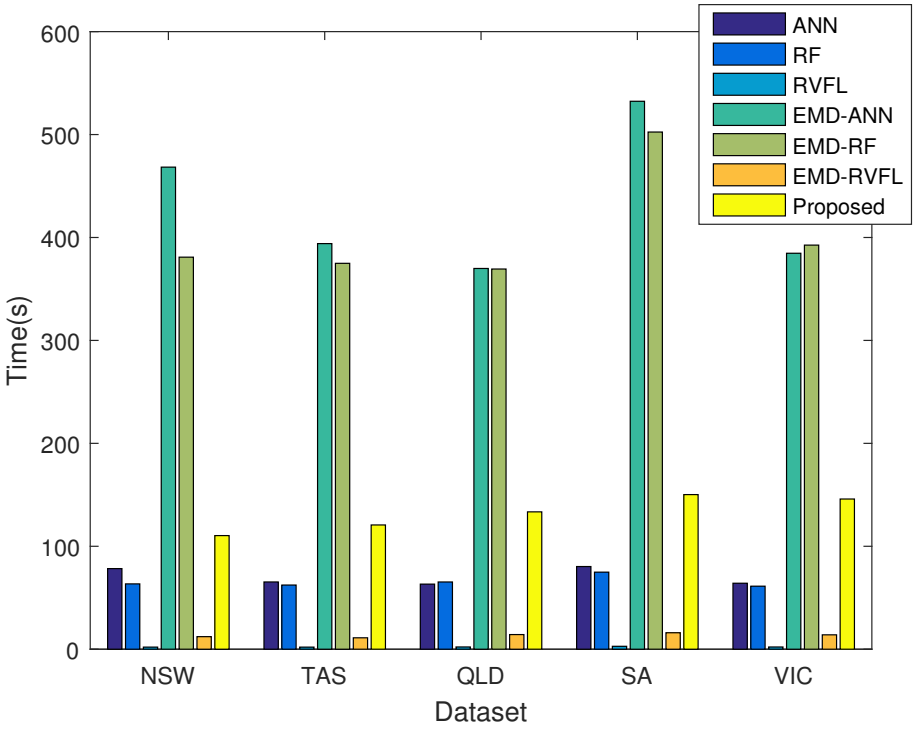


FIGURE 5.6: Computation time of learning models for electric load forecasting

5.5 Comparative experiment

In [7], Fan and Hyndman proposed a semi-parametric additive model for short-term load forecasting, which has been successfully used by AEMO to forecast the short-term loads of two regions with different characteristics in the Australian National Electricity Market. Specifically, the half-hourly demand datasets from Victoria, Australia of the years from January 2004 to September 2008 were used to train the models. The load datasets from October 2008 to March 2009 were used as out-of-sample test data. Except for the historical demand data, the additional input variables include lagged and future temperatures, and calendar variables. In this work, the simulations were implemented using the same training and testing datasets, thereby offering uniform comparisons. The comparison results are shown in Table 5.6, which lead to the conclusion that our proposed method has better performance compared with the benchmark models.

TABLE 5.6: Forecasting results for comparative experiment one. The results of additive model, ANN and Hybrid model are obtained from [7].

	Proposed		Additive model [7]		ANN		Hybrid	
Month	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE
Oct	**77.16**	**1.39%**	88.55	1.66%	134.87	2.57%	121.83	2.15%
Nov	**65.34**	**1.19%**	94.33	1.74%	140.52	2.63%	123.50	2.12%
Dec	**61.76**	**1.18%**	79.89	1.55%	126.39	2.49%	116.34	2.17%
Jan	**90.34**	**1.45%**	110.21	1.88%	168.04	2.81%	126.73	2.14%
Feb	**62.21**	**1.11%**	96.84	1.64%	139.68	2.37%	119.07	1.95%
Mar	**62.58**	**1.14%**	87.45	1.59%	123.21	2.29%	116.49	1.94%
Average	**69.90**	**1.24%**	92.82	1.68%	138.79	2.53%	120.66	2.08%

5.6 Summary

In this chapter, a hybrid incremental learning approach is presented for short-term electric load forecasting, which composed of Discrete Wavelet Transform (DWT), Empirical Mode Decomposition (EMD) and Random Vector Functional Link (RVFL) network. Fifteen electric load datasets from AEMO were used for evaluating the performance of the proposed method by comparing with several benchmarks. Moreover, two comparative experiments were also implemented to verify the effectiveness of the proposed method. Based on the experiment results, the following conclusions are made:

1. Both sufficient number of hidden neurons and functional links can benefit the overall performance of RVFL networks. Taking model complexity into consideration, RVFL with direct links and reasonable number of hidden neurons is recommended.

2. Incremental learning is beneficial for short term electricity load TS forecasting with RVFL and its ensemble models.

3. The proposed DWT-EMD based ensemble approach outperforms EMD based and single structure models.

4. The proposed incremental DWT-EMD based RVFL approach achieves the best rank and significantly outperforms the non-EMD based benchmarks and EMD-RF with a 95% confidence.

Chapter 6

Short-term Wind Power Ramp Forecasting with Empirical Mode Decomposition based Ensemble Learning Techniques

Wind, as a renewable energy source, has attracted public interests for decades. However, wind power TS signal often fluctuates and highly nonlinear because of the intermittent nature of the wind [11, 12]. The fluctuations caused by wind ramp is normally solved by the battery storage systems or conventional fossil power generator. But the effectiveness of the compensation will be low if there are large fluctuations. Wind power ramps, or significant fluctuations, can be classified into two types: up ramp and down ramp [181]. To the author's best knowledge, there is no well-known standard about the definition of wind ramp [182]. Therefore some wind ramp definitions appeared in the literature are adopted in this thesis. if we want to integrate the wind power into the power grid, accurate wind power ramp forecasting is important to optimize the planning and scheduling of power systems [13, 14]. It is also helpful to protect the power transmission and generation system from a sudden rise and drop in power supply [15].

In the literature, numerous research works have been published to forecast the wind power ramps. Some of the works focused on the definition to identify wind ramps [183–185]. In these papers, various power ramp definitions were introduced and compared. Another part of works investigated and presented various methods to forecast the wind power ramps. For example, in [13], artificial neural networks (ANNs) were employed for wind ramp forecasting. Moreover, probabilistic forecasting method is used instead of traditional point forecasting models. In [182], the authors compared SVM, RF and ANN for wind power ramp forecasting. On the other hand, in [181], the authors attempted to forecast the wind power ramp in the Electric Reliability

Council of Texas (ERCOT) wind farms. In that work, the forecasting models were developed from physical models. Another physical model could be found in [186]

6.1 Wind Power Ramp

Wind power ramp is normally caused by changes of wind speed in a specific time, which is quite common. There are two types of wind power ramps: the increasing power ramp (up ramp), and the decreasing power ramp (down ramp). In fact, wind power doesn't have a linear relationship with wind speed due to the different regions of wind power generation, including cut in region, cut out region, cubic region and maximum-output region.

In cut in region, the wind is not powerful to overcome the internal friction in the wind turbine, which means that the power output is zero. However, in the cut out region, the power output is also zero. The reason is that the turbine is halt during strong wind so that the wind turbine can avoid damages caused by over spinning or over heating. Moreover, the wind power remains constant during the maximum-output region, because the power generation is limited by the maximum output capability of the generator in the turbine. In the cubic region, there is a cubic relationship between wind speed and wind power which is supported by the theory in [187].

Significant power ramps can affect the power grid. A significant power ramp occurs when [184]:

$$max(P(t, \ldots, t + \Delta t)) - min(P(t, \ldots, t + \Delta t)) > P_{val} \tag{6.1}$$

where $P(t)$ is the wind power generated at time t, Δt is the time interval, and P_{val} is the threshold.

Another way to look at the power ramp is to examine the power ramp rate [182]. When the power ramp rate exceeds a threshold, we can identify a power ramp in that time frame.

$$\frac{|P(t + \Delta t) - P(t)|}{\Delta t} > PR_{val} \tag{6.2}$$

where PR_{val} is the threshold of the power ramp rate to determine whether there is a ramp or not.

Based on these two definitions, we can define our wind ramp forecasting tasks as two approaches: one is to treat the significant power ramp as binary classification (ramp or no ramp) and the other is to treat the significant power ramp as regression (power ramp rate) and then check whether the rate exceeds the threshold. In the paper, we set up two experiments corresponding to the two approaches.

6.2 Proposed Ensemble Method

In this work, CEEMDAN is used to decompose the wind power signal into several IMFs and one residue . Then a KRR model is trained for each IMF including the residue, which is much more efficient than using SVR, ANN or RF. Finally, a RVFL network is employed to combine the outputs from all sub-series, which generates the finally output [79]. Figure 6.1 is the schematic diagram of this proposed ensemble method, and the procedures can be concluded as:

1. Apply CEEMDAN to decompose the original TS into several IMFs and one residue.
2. Construct the training matrix as the input of each KRR for each IMF and residue.
3. Train KRRs to obtain the prediction results for each of the extracted IMF and residue.
4. Combine all the prediction results by an RVFL model to formulate an ensemble output. For wind power forecasting and wind power ramp forecasting, which belong to time series forecasting, the output is the predicted future value. For wind power ramp classification, the output is the class label.

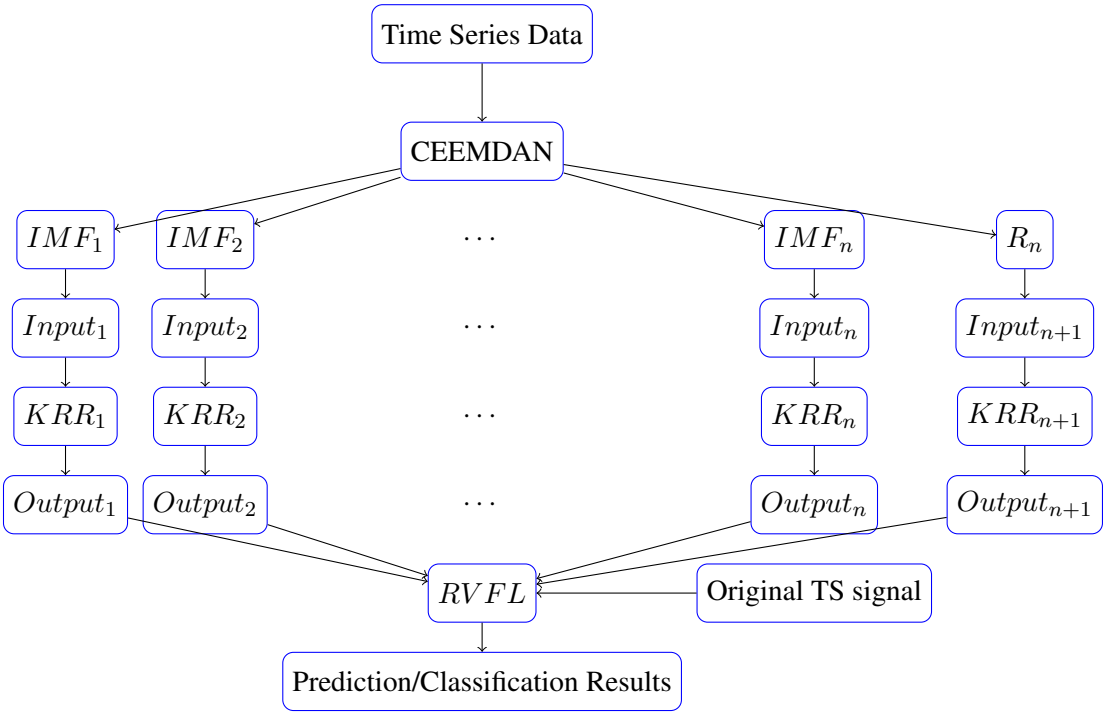


FIGURE 6.1: Schematic Diagram of the Proposed CEEMDAN-KRR-RVFL model

6.3 Experimental Setup

As mentioned in the previous section, the two approaches to forecast the wind power ramp correspond to classification and regression. In this study, the wind power ramp forecasting belongs to binary classification problem because there are only two classes: ramp and no ramp. Regression is to predict a continuous data based on a set of features. In this chapter, wind power ramp rate is a continuous variable and thus regression is applied in this approach.

6.3.1 Datasets

The wind power time series spans from November 2014 to March 2015, which is retrieved from ELIA wind power website [188]. The rated wind power is 712.9 MWh. Figure 6.2 shows a fraction of wind power time series. In this study, $D1$ to $D5$ represent for these five monthly wind power time series datasets. Sub-sampling from 15 min average to hourly average and scaling to $[0, 1]$ interval is applied to each monthly TS dataset. The first 50% of each monthly TS data is used for training and remaining 50% is used for test. Similar procedures are applied to wind power ramp rate time series.

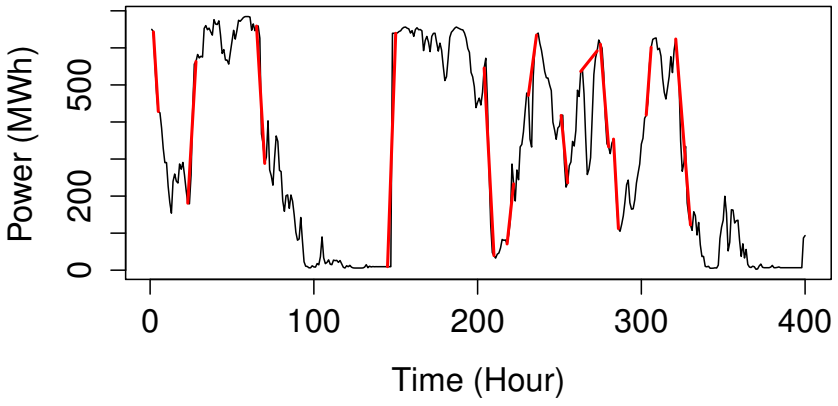


FIGURE 6.2: A Fraction of Wind Power Generated in an ELIA wind farm, red segments denote power ramps.

Based on the wind ramp definition, we set $P_{val} = 25\%$ of the rated power and $\Delta t = 4h$. From Figure 6.2, we can easily find the wind ramps, which are plotted in red segment lines. For the wind ramp rate calculation, we use hourly data difference to represent the ramp rate, i.e. $\Delta t = 1h$.

6.3.2 Data Preprocessing

Imbalanced Data

For wind power ramp data under classification approach, the classes are imbalanced. There are much more 'no ramp' cases (majority class) than 'ramp' cases (minority class), which leads the phenomenon that the accuracy is usually quite high when all predicted labels belong to majority class, therefore misleading the learning models. There are normally two ways to solve the imbalanced data problem: under-sampling the majority class or over-sampling the minority class [189]. In this chapter, down-sampling is used.

Outlier Removal

In order to forecast the wind power and the power ramp as accurately as possible, the outliers in the wind power time series should be removed. In this work, we treat both abnormal data and missing data as outliers. We use two phase outlier removal method: outlier detection phase followed by outlier smoothing phase.

A rolling window with a fixed width rolls along the time series during the outlier detection phase. As a result, there is a segment of time series $\mathbf{X}_w = \{x_i, x_{i+1}, \ldots, x_{i+w}\}$ for each window, where w is the window width. For each $\mathbf{X}_w$, the median absolute deviation (MAD) is calculated (as in equation (6.3)) and those data points that is $x_{MAD} \leq T_{th} \times \mathrm{MAD}$ is considered as an outlier.

$$\mathrm{MAD} = \mathrm{median}_i \left(|X_i - \mathrm{median}_j(X_j)| \right) \tag{6.3}$$

The outlier x_{MAD} identified based on MAD together with the missing data x_{NA} undergoes the outlier smoothing phase. These outlier data points x_t^*, $t^* \in outlier\ time\ stamp$ are estimated based on a causal moving average filter as shown:

$$\hat{x}_{t^*} = \frac{1}{2}(x_{t^*-2} + x_{t^*-1}) \tag{6.4}$$

where $\hat{x}_{t^*}$ is the smoothed value to the outlier.

6.3.3 Cross Validation

Cross validation is a method to avoid over-fitting by providing a validation data set that will evaluate the performance of the trained model before applying to the testing data. Moreover, k-fold cross validation represents that the cross validation process is executed k times to average out the uncertainties.

For time series regression, the partitioning is slightly different from normal classification and regression, because there are cross-correlations among the time series data. The training dataset is firstly randomly partitioned into $2k-1$ subsets and secondly the first $k-1$ subsets are used for training and the following subset is used for testing. Then this process is rolled forward for k times.

6.3.4 Performance Measures

Contingency table (or confusion matrix) is normally used to evaluate the binary class classification model. As shown in Table 6.1, the contingency table is a 2×2 matrix that reflects the relationship between the target and the predicted values.

TABLE 6.1: Contingency Table to Evaluate the Performance of the Binary Class Classification

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

Four performance metrics can be derived from the contingency table:

$$\text{Accuracy} = \frac{TP+TN}{TP+FN+FP+TN} \tag{6.5}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{6.6}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{6.7}$$

$$\text{F Score} = \frac{2TP}{2TP+FP+FN} \tag{6.8}$$

where C is the class of the target data, $\hat{C}$ is the class of the predicted data, TP, TN, FP and FN stand for true positive, true negative, false positive and false negative, respectively.

For imbalanced data, Precision, Recall and F score are preferred for performance evaluation instead of Accuracy [181, 185].

Numerous error metrics for regression models are reported in the literature [11, 12]. In this work, three error metrics are used: Normalized Root Mean Square Error (NRMSE), Normalized Mean

Absolute Error (NMAE) and Mean Absolute Scaled Error (MASE):

$$\text{NRMSE} = \frac{1}{max(x) - min(x)} \sqrt{\text{E}[(\hat{x} - x)^2]} \tag{6.9}$$

$$\text{NMAE} = \frac{1}{max(x) - min(x)} \text{E}\,|\hat{x} - x| \tag{6.10}$$

$$\text{MASE} = \frac{\sum_{j=1}^{n} |\hat{x}_j - x_j|}{\frac{n}{n-1} \sum_{i=2}^{n} |x_i - x_{i-1}|} \tag{6.11}$$

where $\hat{x}$ is the predicted data and x is the target data.

6.4 Results and Discussions

The proposed CEEMDAN-KRR-RVFL is compared with five commonly used machine learning models: ANN, SVM, RF, KRR and RVFL. 5-fold cross validation is employed to optimized the parameters and avoid over-fitting.

6.4.1 Wind Power Forecasting

The first experiment is to predict the wind power generated in the next 12 hours. The historical data for 48 hours is analyzed for prediction. The NMRSE measure for the benchmark methods is tabulated in Table 6.2. It can be observed that the forecasting error increases as forecasting horizon increasing. Overall, the performances of the five single structure methods over five datasets are comparable to each other. Outstanding, the proposed CEEMDAN-KRR-RVFL model achieves the best performance in every case, which reveals the advantage of the EMD based ensemble methods for wind power forecasting. Similar conclusions can be drawn from the examination of NMAE and MASE of the wind power forecasting.

Figure 6.3 shows the statistical testing results based on NRMSE. In the figure, the methods with better ranks are at the top whereas the methods with worse ranks are at the bottom. Moreover, the models within a vertical line whose length is less than or equal to a critical distance have statistically the same performance.

From the statistical testing results, it is proved that the five single structure benchmark models have no significant differences for wind power forecasting. The proposed CEEMDAN-KRR-RVFL achieves the best rank and significantly outperforms the other benchmark methods with a 95% confidence.

TABLE 6.2: Selected NRMSE of the Wind Power Forecasting over 12 hour Forecasting Horizon.

Dataset	Method	Horizon (hour) 1	4	8	12
D1	ANN	0.114	0.215	0.282	0.285
	RF	0.109	0.244	0.366	0.402
	SVR	0.118	0.209	0.244	0.272
	RVFL	0.088	0.192	0.238	0.272
	KRR	0.120	0.223	0.278	0.293
	Proposed	**0.055**	**0.121**	**0.156**	**0.194**
D2	ANN	0.115	0.22	0.342	0.393
	RF	0.095	0.231	0.355	0.393
	SVR	0.106	0.233	0.339	0.362
	RVFL	0.093	0.24	0.347	0.388
	KRR	0.120	0.239	0.360	0.389
	Proposed	**0.068**	**0.133**	**0.215**	**0.281**
D3	ANN	0.134	0.257	0.358	0.419
	RF	0.131	0.27	0.382	0.464
	SVR	0.157	0.281	0.387	0.453
	RVFL	0.107	0.272	0.334	0.400
	KRR	0.153	0.291	0.390	0.445
	Proposed	**0.053**	**0.090**	**0.153**	**0.266**
D4	ANN	0.122	0.257	0.38	0.399
	RF	0.116	0.251	0.36	0.415
	SVR	0.119	0.233	0.302	0.354
	RVFL	0.104	0.274	0.381	0.450
	KRR	0.121	0.267	0.379	0.443
	Proposed	**0.056**	**0.085**	**0.108**	**0.137**
D5	ANN	0.118	0.26	0.353	0.403
	RF	0.126	0.239	0.328	0.367
	SVR	0.146	0.276	0.376	0.411
	RVFL	0.105	0.252	0.352	0.443
	KRR	0.112	0.267	0.380	0.451
	Proposed	**0.046**	**0.102**	**0.157**	**0.151**

6.4.2 Wind Power Ramp Rate Forecasting

The second experiment is to predict the wind power ramp rate for next 12 hours. The NMRSE measure for the six methods is tabulated in Table 6.3. It is obviously that, for datasets D1 and D2, the forecasting error decreases when forecasting time horizon increases; for datasets D4, the forecasting error increases with increasing forecasting time horizon; for dataset D3, there is a local peak at 4 hour ahead forecasting horizon during the 1–12 hour horizons; and for dataset D5, there is a local valley at 8 hour ahead forecasting horizon during the examined forecasting horizons. Overall, the performances of the five single structure methods over five

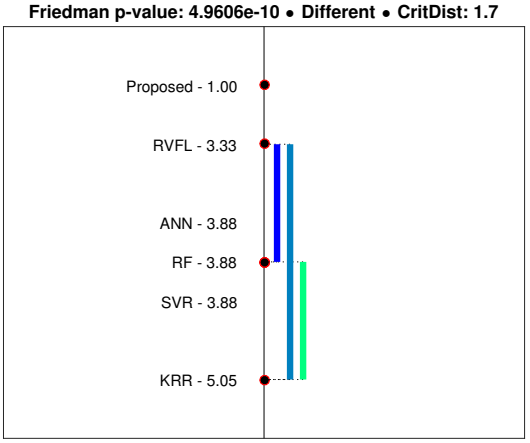


FIGURE 6.3: Nemenyi test for wind power forecasting based on NRMSE. The critical distance is 1.7.

datasets are comparable to each other. Meanwhile, the ensemble method outperforms all the other benchmark models in every case.

Same as the previous experiment, the Nemenyi test is also used to compare the wind power ramp rate forecasting performances. The results based on NRMSE are shown in Figure 6.4. It can be concluded that the proposed ensemble method has achieved the best rank and outperforms the benchmark models, except RF, with a 95% confidence.

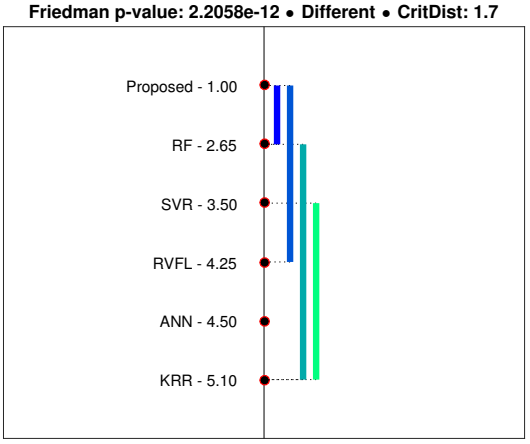


FIGURE 6.4: Nemenyi test for wind power ramp rate forecasting based on NRMSE. The critical distance is 1.7.

TABLE 6.3: Selected NRMSE of the Wind Power Ramp Rate Forecasting over 12 hour Forecasting Horizon.

		Horizon (hour)			
Dataset	Method	1	4	8	12
D1	ANN	0.467	0.427	0.377	0.364
	RF	0.429	0.441	0.439	0.436
	SVR	0.45	0.435	0.437	0.473
	RVFL	0.433	0.379	0.342	0.325
	KRR	0.467	0.440	0.438	0.451
	Proposed	**0.303**	**0.289**	**0.276**	**0.258**
D2	ANN	0.491	0.452	0.380	0.345
	RF	0.440	0.374	0.333	0.324
	SVR	0.456	0.415	0.368	0.338
	RVFL	0.459	0.422	0.382	0.370
	KRR	0.465	0.425	0.367	0.364
	Proposed	**0.290**	**0.278**	**0.254**	**0.253**
D3	ANN	0.377	0.376	0.365	0.358
	RF	0.299	0.295	0.332	0.361
	SVR	0.316	0.341	0.341	0.365
	RVFL	0.353	0.373	0.333	0.327
	KRR	0.350	0.376	0.356	0.372
	Proposed	**0.265**	**0.266**	**0.253**	**0.271**
D4	ANN	0.416	0.392	0.377	0.336
	RF	0.380	0.356	0.316	0.307
	SVR	0.399	0.401	0.367	0.330
	RVFL	0.405	0.413	0.436	0.456
	KRR	0.421	0.423	0.410	0.443
	Proposed	**0.227**	**0.231**	**0.242**	**0.254**
D5	ANN	0.374	0.346	0.329	0.320
	RF	0.336	0.314	0.286	0.273
	SVR	0.335	0.320	0.320	0.301
	RVFL	0.366	0.352	0.331	0.352
	KRR	0.340	0.353	0.334	0.349
	Proposed	**0.253**	**0.250**	**0.261**	**0.252**

6.4.3 Wind Power Ramp Classification

The third experiment is to classify the wind power ramp in the next 12 hours based on the previous 48 hours historical data. If there is one or more wind power ramps occurred in the next 12 hours, the class is '1' (has ramp), otherwise the class is '-1' (no ramp). The 5-fold cross validation is based on F Score to optimize the parameters of the methods. For classification problem, SVM and Kernel Ridge Regression Classification (KRRC) are used instead of SVR and KRR.

The error measures are tabulated in Table 6.4. We can see that in terms of F score, among the five single structure models, RVFL has the best overall performance with the exception on D4.

As F score is closely related to Precision and Recall, we also reported these two measures for the four datasets. Moreover, the proposed CEEMDAN-KRR-RVFL wins in all the cases, which proves its advantage in wind power ramp classification.

6.4.4 Computation Time

The average computation time of training and testing over five datasets are shown in Table 6.5. We can see that for the regression approaches: wind power forecasting and power ramp rate forecasting, RVFL and KRR have significantly shorter training and testing time than the other benchmark methods. For classification, only RF had shorter training time than RVFL but not for the testing time. Overall, due to the fast computation speed of RVFL and KRR, the proposed ensemble method has reasonable overall computation speed, while has the highest accuracy.

TABLE 6.4: Performance Measures of the Power Ramp Classification in the next 12 hours.

	Dataset	ANN	RF	SVM	RVFL	KRRC	Proposed
F Score	D2	0.280	0.228	0.281	0.281	0.215	**0.373**
	D3	0.255	0.333	0.240	0.263	0.228	**0.342**
	D4	0.413	0.393	0.447	0.395	0.390	**0.449**
	D5	0.273	0.252	0.258	0.321	0.255	**0.352**
Precision	D2	0.259	0.164	0.204	0.194	0.153	**0.287**
	D3	0.187	0.255	0.214	0.195	0.201	**0.261**
	D4	0.354	0.322	0.393	0.324	0.320	**0.395**
	D5	0.239	0.232	0.223	0.218	0.220	**0.246**
Recall	D2	0.304	0.377	0.449	0.507	0.364	**0.531**
	D3	0.403	0.481	0.273	0.403	0.262	**0.494**
	D4	0.495	0.505	0.516	0.505	0.498	**0.520**
	D5	0.319	0.275	0.304	0.609	0.302	**0.616**

TABLE 6.5: Average Computation Time (sec) over 5 Datasets.

Method	Power		Ramp Rate		Classification	
	Train	Test	Train	Test	Train	Test
ANN	89.79	7.73	102.07	19.53	307.63	10.68
RF	7.35	7.35	11.56	11.55	0.42	0.41
SVM	12.84	1.01	11.58	1.18	5.67	0.13
RVFL	0.80	0.01	0.80	0.01	2.95	0.08
KRR	1.26	0.10	1.18	0.12	2.25	0.48
Proposed	15.56	1.12	13.32	1.34	25.30	2.12

6.5 Summary

In this chapter, an ensemble learning method called CEEMDAN-KRR-RVFL is proposed for wind power ramp forecasting. Five wind power datasets and six benchmark models are used to demonstrate the attractiveness of the proposed method. According to the experiment results,

the proposed ensemble method achieves the best performance based on accuracy, while has reasonable fast computation speed.

For the future works, the wind power ramp classification can be extended from binary class to multiple class. Some examples of additional classes are up-ramp and down-ramp events, and strong ramp and weak ramp events. Probabilistic forecasting is another research direction to replace point forecasting in this work. For probabilistic forecasting, each forecast value is a set of probabilities and a corresponding interval instead of a single value. Moreover, more ensemble methods, deep learning methods and multi-variant models can be used to improve the performance of wind power ramp forecasting/classification.

Chapter 7

Summary of Part I

Ensemble methods generally improve the performance of a single forecasting model, which follows the concept of "perturb and combine", and have been applied in many fields such as time series forecasting, speech recognition, image classification and so on [57]. The rationale of ensemble methodology can be summarized as integrating multiple models to build a more predictive model (either classifier or regressor) [59].

In the first part of this thesis, we work on ensemble methods for power system related time series forecasting. In Chapter 3, we present a decision tree ensemble method named oblique random forest with least square estimation. The proposed oblique RF has better performance compared with the original RF in both generic TS assessment and short term electricity load demand forecasting, which shows the advantages of multivariate methods. In Chapter 4, two ensemble deep learning methods are proposed for short term electric load forecasting: ensemble deep belief network (EDBN) and EMD based ensemble deep belief network (EMD-DBN). According to the simulation results, the ensemble method employed improves the performance of DBN successfully. Moreover, in Chapter 5, we investigate the ensemble method for incremental learning, which makes use of a non-iterative model with high efficiency: Random Vector Functional Link (RVFL) network. The proposed DWT-EMD based ensemble approach outperforms EMD based and single structure models. On the other hand, in Chapter 6, ensemble learning methods are employed to deal with an important challenge on wind power utilization: wind power ramp forecasting. Three kinds of experiments are conducted: wind power forecasting, wind power ramp forecasting and wind power ramp classification. The computation speed is also compared among the benchmark models. The experimental results demonstrate the effectiveness of the proposed methods.

In this chapter, we conduct a overall comparison among all the ensemble learning methods mentioned in the above chapters for short term electric load forecasting. The ensemble learning methods include oblique RF, ensemble neural network (ENN), ensemble deep belief network

(EDBN), EMD based ensemble deep belief network (EMD-DBN), EMD based SVR (EMD-SVR), EMD based ANN (EMD-ANN), EMD based RF (EMD-RF), EMD based RVFL (EMD-RVFL), EMD-KRR-RVFL and DWT-EMD-RVFL. To make the experiment results consistence and comparable, we apply all above ensemble learning models for the same datasets as in Chapter 5, which are the electric load datasets from AEMO of year $2013 \sim 2015$ from five states of Australia. To help the readers recall the details of the datasets, Table 7.1 is repeated below, which has been demonstrated in Chapter 5.

TABLE 7.1: Summary of AEMO load datasets

Dataset	Year	Length	Min	Median	Mean	Max	Std
QLD	2013	17520	4148.7	5752.1	5703.7	8278.4	747.0
	2014	17520	4073.0	5726.0	5745.7	8445.3	794.0
	2015	17520	4281.4	6005.6	6035.4	8808.7	777.2
NSW	2013	17520	5113.0	8045.0	7981.6	13788	1190.9
	2014	17520	5138.1	7987.4	7917.8	11846	1170.1
	2015	17520	5337.4	7990.4	7979.8	12602	1232.7
TAS	2013	17520	659.5	1109.0	1129.3	1650.3	142.3
	2014	17520	569.1	1088.7	1109.7	1630.1	139.0
	2015	17520	479.4	1112.3	1138.2	1667.2	145.3
SA	2013	17520	728.6	1389.3	1426.6	2991.3	301.7
	2014	17520	682.5	1360.8	1403.3	3245.9	312.8
	2015	17520	696.3	1352.7	1398.5	2870.4	306.0
VIC	2013	17520	3551.6	5458.1	5511.8	9587.5	895.9
	2014	17520	3272.9	5307.8	5324.4	10240	921.4
	2015	17520	3369.1	5186.5	5194.6	8579.9	864.7

The prediction results for one-day-ahead electric load forecasting are shown in Table 7.2. The numbers in bold mean that the corresponding method achieves the best performance for this dataset under this performance measurement. Same as previous chapters, the Friedman test and Nemenyi post-hoc test are also applied to rank the ensemble learning models. The comparison results of statistical test based on RMSE and MAPE are shown in Figure 7.1 and Figure 7.2, respectively.

From the simulation and statistical testing results, several conclusions can be made about ensemble learning for short term electric load forecasting.

1. According to the sufficient small Friedman p-value, there exists significant differences among these ensemble learning methods.

2. Two ensemble deep learning methods, EMD-DBN and EDBN achieve the second and the third place among all the ensemble learning models, which demonstrates the attractiveness and potential of ensemble deep learning models for real world complicated time series forecasting.

3. Incremental learning is beneficial for short term electricity load TS forecasting with RVFL and its ensemble models.

TABLE 7.2: Prediction results for one-day-ahead electric load forecasting

Dataset	Year	Metrics	Prediction model								
			ORF [77]	ENN [34]	EDBN [34]	EMD-DBN [16]	EMD-SVR [168]	EMD-ANN [169]	EMD-RF [178]	EMD-RVFL [179]	DWT-EMD-RVFL
QLD	2013	RMSE	282.465	250.562	238.102	249.964	243.367	**232.264**	272.903	255.507	233.221
		MAPE	3.514%	3.328%	3.098%	3.592%	3.123%	**2.888%**	3.360%	3.276%	2.967%
	2014	RMSE	352.297	280.368	278.582	271.875	291.320	**262.607**	327.659	289.070	284.766
		MAPE	4.030%	3.920%	3.609%	3.602%	3.879%	**3.340%**	3.800%	3.623%	3.440%
	2015	RMSE	348.892	350.932	330.921	308.993	328.091	333.460	336.680	351.685	**277.861**
		MAPE	3.979%	3.897%	3.810%	4.024%	3.788%	3.857%	3.830%	3.995%	**3.341%**
NSW	2013	RMSE	601.403	511.870	449.051	466.845	451.129	452.886	515.574	495.384	**436.890**
		MAPE	5.912%	4.892%	4.402%	4.765%	4.358%	4.425%	4.942%	4.938%	**4.240%**
	2014	RMSE	586.905	529.572	492.12	**409.846**	489.231	500.450	532.745	503.918	419.416
		MAPE	5.376%	5.002%	4.689%	4.493%	4.783%	4.818%	4.897%	5.027%	**3.906%**
	2015	RMSE	618.089	634.54	623.029	497.732	607.563	645.314	660.351	679.760	**466.842**
		MAPE	5.433%	5.798%	5.590%	5.291%	5.309%	5.825%	6.078%	6.302%	**4.368%**
TAS	2013	RMSE	69.912	66.549	61.074	60.843	63.980	61.135	67.012	61.779	**58.175**
		MAPE	5.240%	4.785%	4.437%	4.536%	4.598%	4.257%	4.835%	4.376%	**4.228%**
	2014	RMSE	67.541	68.907	64.984	**50.519**	65.201	63.639	69.174	64.944	58.721
		MAPE	5.050%	5.168%	4.972%	**3.955%**	5.081%	4.779%	5.213%	4.847%	4.361%
	2015	RMSE	69.510	68.095	65.329	**45.023**	68.802	63.514	68.839	65.591	58.687
		MAPE	4.831%	4.894%	4.784%	**3.260%**	4.912%	4.524%	4.969%	4.690%	4.169%
SA	2013	RMSE	201.843	192.647	180.340	155.420	175.240	189.791	189.728	188.725	**146.936**
		MAPE	12.017%	11.908%	10.638%	9.004%	10.982%	11.159%	11.293%	11.161%	**8.740%**
	2014	RMSE	184.331	175.450	160.920	**120.308**	159.087	166.654	174.346	169.382	142.976
		MAPE	11.490%	10.392%	10.082%	**7.742%**	10.210%	10.116%	10.599%	10.556%	8.769%
	2015	RMSE	241.361	236.398	225.342	234.440	230.892	226.507	242.910	235.096	**192.189**
		MAPE	11.983%	11.872%	11.346%	11.892%	11.583%	11.661%	11.773%	11.880%	**10.122%**
VIC	2013	RMSE	499.683	496.783	481.902	387.037	489.409	461.255	502.737	499.537	**376.096**
		MAPE	7.034%	7.102%	7.120%	5.953%	6.893%	6.606%	7.141%	7.254%	**5.368%**
	2014	RMSE	523.690	501.529	510.234	394.842	501.231	472.920	497.792	457.876	**350.499**
		MAPE	8.783%	7.983%	7.456%	6.651%	8.546%	7.842%	8.355%	7.700%	**5.528%**
	2015	RMSE	554.269	548.320	525.341	481.864	525.903	538.957	540.052	542.721	**430.092**
		MAPE	8.231%	8.045%	8.021%	7.320%	8.045%	8.113%	8.098%	8.624%	**6.661%**

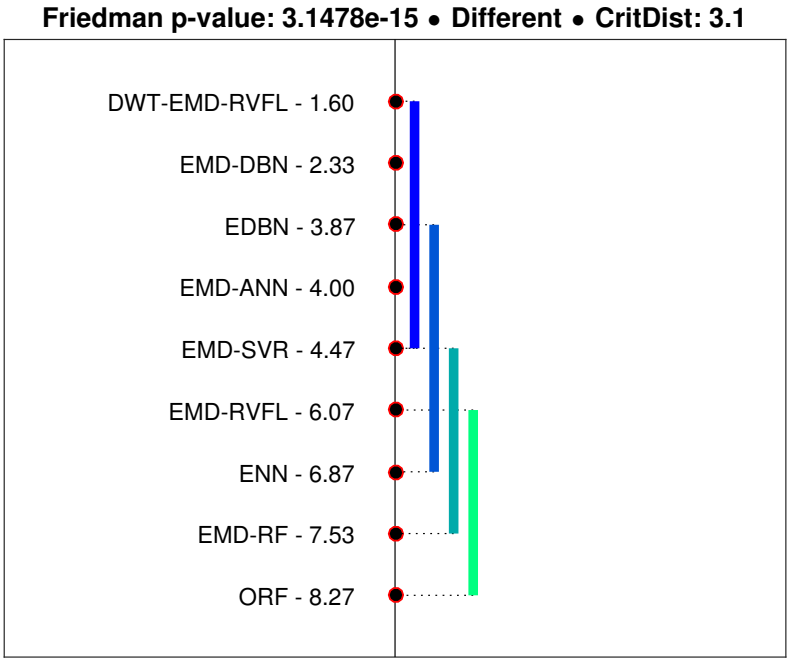


FIGURE 7.1: Nemenyi test for electric load forecasting based on RMSE. The critical distance is 3.1.

4. incremental DWT-EMD based RVFL approach achieves the best rank and significantly outperforms EMD-RVFL, ENN, ORF and EMD-RF with a 95% confidence.

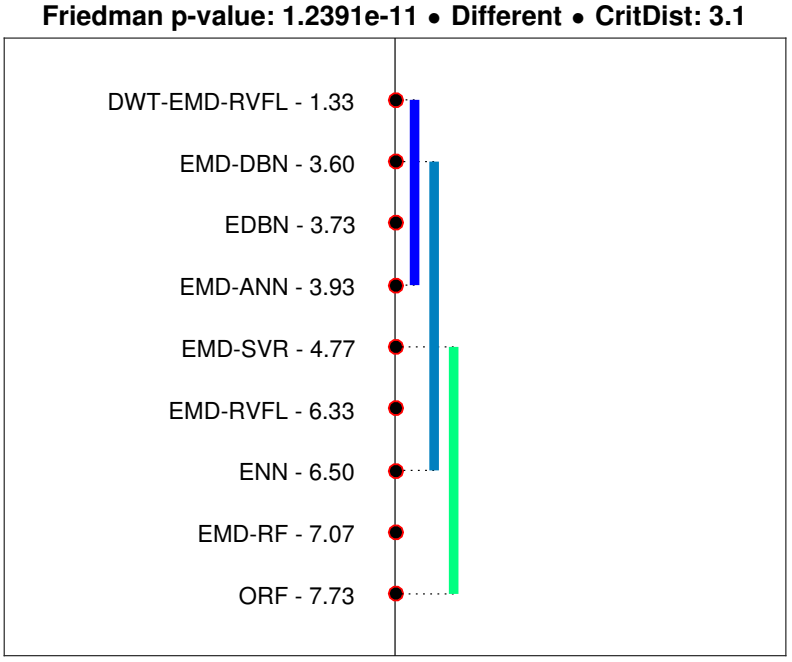


FIGURE 7.2: Nemenyi test for electric load forecasting based on MAPE. The critical distance is 3.1.

Part II

Ensemble Learning with Applications in Financial Markets

Chapter 8

Short-term Electricity Price Forecasting with Empirical Mode Decomposition based Ensemble Kernel Machines

In this part, we focus on ensemble learning for financial markets related time series forecasting. Two kinds of financial time series forecasting problems are considered: electricity price and stock price forecasting. In this chapter, short term electricity price forecasting by ensemble methods is investigated [80]. In next chapter, the topic would be ensemble learning fusion with multiple indicators for stock price forecasting.

8.1 Introduction of Electricity Price Forecasting

Electricity price forecasting plays an important role in the power market operation nowadays. Under the help of accurate short term electricity price forecasting methods, not only the power suppliers are able to adjust their biding strategies to achieve the maximum benefit, but also consumers can decide whether to buy electricity from the pool or use self-production capability to avoid unacceptable high prices [190]. Short term electricity price forecasting belongs to time series (TS) forecasting paradigm, which aims to predict the future electricity price ranging from hours to on day ahead by analyzing TS data itself and extracting meaningful characteristics. However, electricity is economically non-storable, while a constant balance between production and consumption is needed for stable power supply. In practice, electricity load demand TS often performs highly nonlinear patterns due to various exogenous factors such as climate change,

economic fluctuation, special occasions, and so on [34, 191]. These unique and specific reasons lead to price dynamics not observed in any other market and thus make accurate electricity price forecasting a challenging task [192].

Over the past seventeen years since the year 2000, a wide variety of methods and ideas have been published for electricity price forecasting (EPF) with varying degrees of success, which can be categorised into linear statistical methods and nonlinear machine learning models [54]. For linear models, normally statistical theories and mathematical equations are used for extrapolating the future values of TS. The most successful linear models include linear regression [26], Holt-Winters exponential smoothing [24], Autoregressive Integrated Moving Average (ARIMA) [25], and so on. Machine learning methods can learn features from and also make forecasts on TS data, which build a model from example inputs in order to make data-driven predictions, instead of following strictly static program instructions [193]. With the rapid development of computational intelligence, machine learning methods have been widely applied for various research fields including short-term electricity price forecasting. The most widely used machine learning algorithms include artificial neural network (ANN) [194], support vector regression (SVR) [27], fuzzy comprehensive evaluation [195], etc.

Kernel machines has become very popular since Support Vector Machine (SVM) being introduced in 1995 [27]. To define complex functions of the input space, SVM performs a non-linear mapping of the data into a high dimensional space, which is known as "kernel tricks". SVM has the advantage of giving a single solution that is characterized by the global minimum of the optimized functional, compared to ANN which is frequently trapped in a local minimum. Many SVM based electricity price forecasting algorithms exist in the literature. For example, in [29], a hybrid model called SVR-ARIMA that combines both SVR and ARIMA models was proposed for short term EPF problems. Besides for SVM, possibly the most elementary algorithm that can be kernelized is ridge regression. In other words, Kernel Ridge Regression (KRR) combines Ridge Regression (linear least squares with $l2-$norm regularization) with the kernel trick. In contrast to SVR, fitting KRR can be done in closed-form and is typically faster for medium-sized datasets [30, 31].

8.2 Proposed Ensemble Method

In this work, an ensemble method named EMD-KRR-SVR approach is proposed for electricity price forecasting. The electricity price data is decomposed into several IMFs and one residue by EMD method first. Then a KRR network is trained for each IMF including the residue, which is much more efficient than using SVR or SLFN. The final prediction results are given by combining the outputs from all sub-series using an SVR model, which ensures the overall accuracy.

Figure 8.1 is the schematic diagram of this proposed ensemble method, and the procedures can be concluded as:

1. Apply EMD to decompose the original TS into several IMFs and one residue.
2. Construct the training matrix as the input of each KRR for each IMF and residue.
3. Train KRRs to obtain the prediction results for each of the extracted IMF and residue.
4. Combine all the prediction results by an SVR model to formulate an ensemble output for TS forecasting.

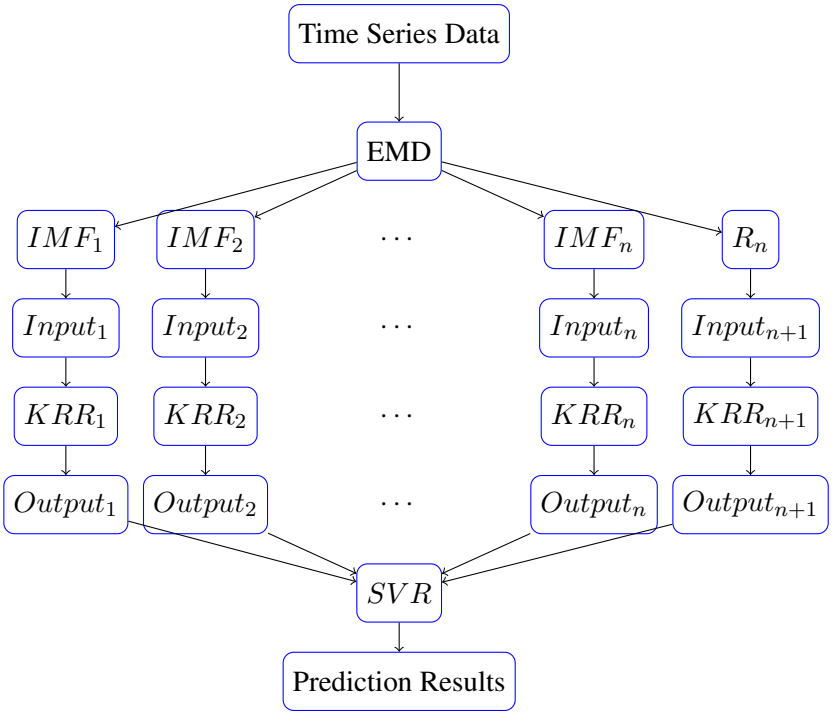


FIGURE 8.1: Schematic Diagram of the Proposed EMD-KRR-SVR approach

8.3 Experiment setup

In this work, the electricity price datasets from Australian Energy Market Operator (AEMO) [173] were used for evaluating the performance of benchmark learning models. There are totally three electricity price datasets of year 2016 from three states of Australia: New South Wales (NSW), Tasmania (TAS) and Queensland (QLD). For each dataset, to reduce the influence of climate change due to different season, four months were selected to perform comparison: January, April, July and October. For each month, the first three weeks were used for training, while the remaining one week was used for testing.

For the time series electricity price datasets, all the training and testing values are linearly scaled to $[0, 1]$. To implement the simulation, LIBSVM toolbox was used for the SVR model [155], while neural network toolbox in Matlab was used for constructing neural networks, including SLFN and EMD based SLFN (EMD-SLFN). Moreover, the Kernel Methods Toolbox for Matlab was used for KRR and the proposed EMD-KRR-SVR approach [196].

For SVR and EMD based SVR, we use the RBF kernel function with parameters chosen by a grid search. The range of C is $[2^{-4}, 2^4]$, and the range of σ is $[10^{-3}$,$10^{-1}]$. For SLFN and EMD-SLFN, the size of neural networks is determined by the size of input vector. The number of iterations for back propagation is set as 1000. We choose Gaussian kernel as the kernel in KRR. The regularization constant is searched within the range $[10^{-8}, 10^8]$ with the stepsize of $10^{0.2}$; while the range of Gaussian kernel width is $[10^{-4}, 10^4]$ with the same stepsize.

Root Mean Square Error (RMSE) is used to evaluate the performance of learning models. It is defined as

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i' - y_i)^2} \tag{8.1}$$

where y_i' is the predicted value of corresponding y_i, and n is the number of data points in the testing time series.

8.4 Results and Discussion

In this section, six benchmark methods were implemented for electricity price forecasting to perform a comparison with the proposed EMD-KRR-SVR model.

8.4.1 Performance comparison for short-term electricity price forecasting

In this work, the persistence method was employed as the baseline for comparing the performance of learning models. This method assumes the conditions at the future time the same as the current values, which has good accuracy due to the highly periodic characteristic of electricity price TS. The prediction results for short-term electricity price forecasting are shown in Table 8.1, where the forecasting horizon is half an hour. The numbers in bold mean that the corresponding method has the best performance for this dataset under this performance measure. According to the prediction results, we can conclude that all the machine learning models outperform the persistence method for short-term electricity price forecasting.

To reveal the advantages of EMD based ensemble methods, we implemented the single structure models SVR, SLFN and KRR for EPF, and conducted an comparison with their EMD hybrid models. Moreover, all of the EMD based ensemble methods have the best performance cases, which shows that they have comparable performance with each other. However, the proposed EMD-KRR-SVR achieves the best performance in most cases, which means that the proposed method has more advantages compared with the benchmark models.

TABLE 8.1: Prediction results for half-an-hour ahead electricity price forecasting ($/MWh)

Dataset	Month	Prediction model						
		Persistence	SVR [27]	SLFN [158]	KRR [30]	EMD-SVR [168]	EMD-SLFN [169]	Proposed
NSW	Jan	20.585	18.410	20.203	19.991	12.681	12.409	**12.356**
	Apr	34.512	30.004	32.566	32.994	25.120	22.131	**20.383**
	Jul	27.387	24.342	24.792	25.741	**19.101**	19.723	20.472
	Oct	19.336	16.729	18.058	18.963	12.345	12.767	**11.729**
TAS	Jan	22.403	21.184	21.757	21.831	18.303	**16.497**	18.045
	Apr	23.395	20.856	21.163	22.544	20.394	**19.196**	19.734
	Jul	25.636	23.278	24.797	23.104	**14.839**	15.891	15.615
	Oct	16.185	15.718	15.967	15.794	12.289	12.980	**12.131**
QLD	Jan	335.873	240.409	268.652	241.917	229.549	232.249	**227.077**
	Apr	31.803	30.582	30.426	31.469	20.748	23.367	**20.719**
	Jul	28.837	26.235	15.258	26.788	**19.160**	19.636	20.453
	Oct	21.221	18.693	23.697	20.098	13.246	14.985	**12.813**

In order to give a detailed analysis of these results, we employ Friedman test [197] and Nemenyi post-hoc test [198] to test the significance of the differences among these learning models. The Friedman test ranks the algorithms for each dataset separately, and then assign average ranks in case of ties. The null-hypothesis states that all the algorithms have the same performance. If the null-hypothesis is rejected, in order to tell whether the performances of two among totally k learning models are significantly different, the Nemenyi post-hoc test is applied to compare all the learning models with each other. The comparison results of statistical test based on RMSE is shown in Figure 8.2. From the statistical test results, the proposed EMD-KRR-SVR achieves the best rank and significantly outperforms the non-EMD based methods with a 95% confidence.

8.4.2 Computation time comparison

Figure 8.3 shows the computation time of benchmark methods for electricity price forecasting in Tasmania (TAS). Obviously, the computational speed of KRR is superior than SLFN and SVR. SVR requires a grid search on C and σ, and SLFN is iteratively tuned by BP algorithm to convergence to the optimal weights. These repetitive parameter tuning processes cause SLFN and SVR less efficient than KRR, which has closed form solutions.

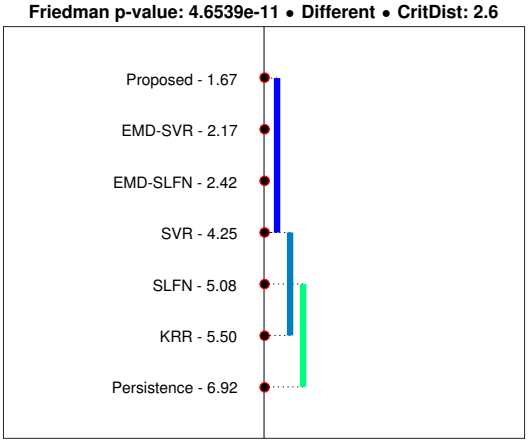


FIGURE 8.2: Nemenyi test for electricity price forecasting based on RMSE. The critical distance is 2.6.

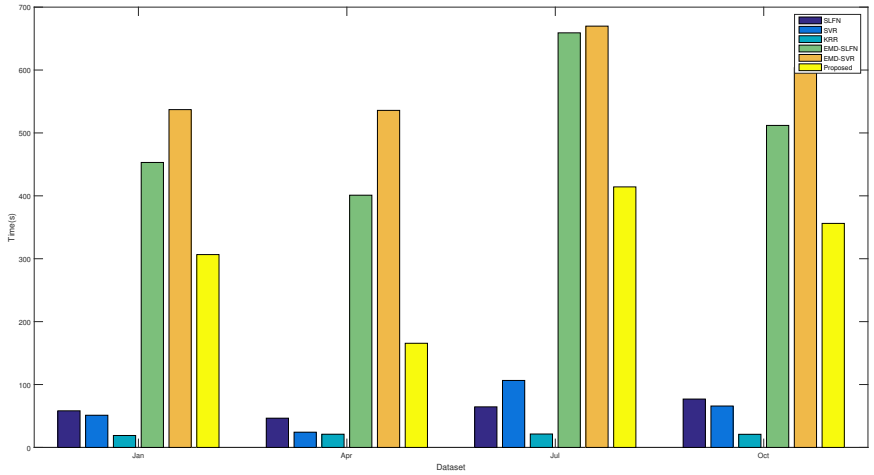


FIGURE 8.3: Computation time of learning models for electricity price forecasting in Tasmania (TAS)

8.5 Summary

In this chapter, we present an ensemble kernel machines for short-term electricity price forecasting composed of EMD, KRR and SVR. The electricity price signal was first decomposed into several intrinsic mode functions (IMFs) by EMD, followed by a KRR which was used to model each extracted IMF and predict the tendencies. Finally, the prediction results of all IMFs were combined by an SVR to obtain an aggregated output for electricity price. Three electricity price datasets from AEMO were used for evaluating the performance of the proposed method. Moreover, six benchmarks methods were implemented to perform a comparison with the proposed method. From the forecasting results, the following conclusions are made:

1. EMD based hybrid methods, including EMD-SVR, EMD-SLFN and the proposed EMD-KRR-SVR, significantly outperform the corresponding single structure models for short-term electricity price time series forecasting.

2. The computation time of KRR is the shortest among all of the benchmark models.

3. The proposed EMD-KRR-SVR approach achieves the best performance for short-term electricity price forecasting, and also has the advantages of efficiency.

Chapter 9

Fusion of Multiple Indicators with Ensemble Incremental Learning Techniques for Stock Price Forecasting

In this chapter, we investigate the performance of ensemble learning methods for short term stock price forecasting. The incremental ensemble learning model employed in this chapter shares the similar idea with Chapter 5, which is composed of Discrete Wavelet Transform (DWT), Empirical Mode Decomposition (EMD), Random Vector Functional Link network (RVFL) and Support Vector Regression (SVR).

9.1 Literature Review of Stock Price Forecasting

In modern financial markets and industrial fields, big data mining, analysis and forecasting play a important role for companies to optimize their plans and strategies in order to keep themselves competitive [17]. Stock price forecasting belongs to time series (TS) forecasting paradigm, which aims to predict the future stock market price ranging from hours to several days ahead by analyzing TS data itself and extracting meaningful characteristics [16]. Among all the challenging tasks in the field of financial time series forecasting, stock price forecasting is regarded as one of the most difficult one due to the highly non-linear and non-stationary patterns of stock price TS caused by numerous influence factors, such as economy, government, enterprise and investors [18].

Machine learning methods have been widely applied in various research fields including stock price forecasting. Some examples of widely used machine learning algorithms are generalized autoregressive conditional heteroscedasticity (GARCH) [199, 200], artificial neural network

(ANN) [194], and support vector regression (SVR) [27]. For example, in [201], an evolutionary Levenberg-Marquardt neural networks based hybrid model was proposed for stock price forecasting, along with some data pre-processing techniques. In [202], "a forecasting model based on chaotic mapping, firefly algorithm and SVR was proposed to predict stock market price with higher accuracy than ANNs".

In recent days, a number of works were published to investigate if deep learning methods can be adapted the field of financial forecasting, which include convolutional neural network (CNN) [203], deep belief network (DBN) [108], long short-term memory network (LSTM) [204]. For example, in [205], "a combination of neural tensor network and a deep convolutional neural network was presented to model the influences of events on stock price movements". Moreover, a deep learning model was proposed for stock prediction using numerical and textual information in [206], which converts newspaper articles into their distributed representations via Paragraph Vector and models the temporal effects of past events on opening prices about multiple companies with LSTM.

9.1.1 Indicators of Stock Market

To construct the model which can predict n days ahead stock price, ten technical indicators describing t^{th} day are used as inputs. The output is set as $(t+n)^{th}$ day's closing price. The definition and significance of some important technical indicators are explained as follows:

1. **MACD**, moving average convergence divergence, is a trading indicator used to reveal changes in the strength, direction, momentum, and duration of a trend in a stock's price. The MACD indicator is a collection of three time series calculated from historical price data, including the MACD series proper, the "average" series, and the "divergence" series. The MACD series is the difference between a short period exponential moving average (EMA), and a longer period EMA of the price series, which are normally set as 12-day EMA and 26-day EMA, respectively. The average series is an EMA of the MACD series itself. Furthermore, the divergence series is the difference of MACD series and average series.

2. **RSI**, short for relative strength index, belongs to momentum oscillators, which indicate to the trader whether or not a stock's price action is created by those over-buying or over-selling it. There has always been a little confusion over the difference between RSI and relative strength (RS), which is the ratio of two EMAs of upward changes and downward changes. Stocks which have had more or stronger positive changes have a higher RSI than stocks which have had more or stronger negative changes.

3. **Stochastic Oscillator** is a momentum indicator comparing the closing price of a security to the range of its prices over a certain period of time. The general theory serving as the foundation for this indicator is that in a market trending upward, prices will close near the high, and in a market trending downward, prices close near the low. Another indicator called Larry Willian's R% is similar to K% stochastic oscillator, mirrored at the 0%-line when using the same time interval.

4. **Commodity Channel Index** is a frequently used tool for traders to identify cyclical trends of stock market, including price reversals, price extremes and trend strength. CCI is calculated as the difference between the typical price of a commodity and its simple moving average, divided by the mean absolute deviation of the typical price.

Table 9.1 summarizes the details of ten technical indicators used in this work.

TABLE 9.1: Technical details of the selected stock market indicators [8]

Name of indicators	Formulas
Simple n-day moving average	$\frac{C_t+C_{t-1}+\cdots+C_{t-n-1}}{n}$
Weighted n-day moving average	$\frac{nC_t+(n-1)C_{t-1}+\cdots+C_{t-n-1}}{n+(n-1)+\cdots+1}$
Momentum	$C_t - C_{t-(n-1)}$
Stochastic K%	$\frac{C_t-LL_{t-(n-1)}}{HH_{t-(n-1)}-LL_{t-(n-1)}} \times 100$
Stochastic D%	$\frac{\sum_{i=0}^{n-1} K_{t-i}}{10}\%$
Relative strength index (RSI)	$100 - \frac{100}{1+(\sum_{i=0}^{n-1} UP_{t-i}/n)/(\sum_{i=0}^{n-1} DW_{t-i}/n)}$
Moving average convergence divergence (MACD)	$MACD(n)_{t-1} + \frac{2}{n+1} \times (DIFF_t - MACD(n)_{t-1})$
Larry William's R%	$\frac{H_n-C_t}{H_n-L_n} \times 100$
A/D (Accumulation/Distribution) oscillator	$\frac{H_t-C_{t-1}}{H_t-L_t}$
CCI (Commodity channel index)	$\frac{M_t-SM_t}{0.015D_t}$

C_t is the closing price, L_t is the low price and H_t is the high price at time t, $DIFF_t = EMA(12)_t - EMA(26)_t$, LL_t and HH_t are lowest low and highest high price in the last t days, respectively. $M_t = \frac{H_t+L_t+C_t}{3}$, $SM_t = \frac{\sum_{i=1}^{n} M_{t-i+1}}{n}$, $D_t = \frac{\sum_{i=1}^{n} |M_{t-i+1}-SM_t|}{n}$, UP_t means upward price change while DW_t is the downward price change at time t.

9.2 Proposed Fusion Incremental Learning Method

As an efficient ensemble method to perform TS signal decomposition, EMD has a major drawback called mode mixing problem: one IMF may consist of signal spanning a wide band of frequency, or more than one IMFs contain signals in a similar frequency band [171]. In the literature, this problem is normally solved by ensemble of EMD (EEMD) [172], which works by applying EMD to uncorrelated Gaussian noise added TS signal repetitively and combining the results to remove the noise. In this chapter, the MODWT is employed to deal with the frequency issue, followed by EMD to perform better decomposition. Then an RVFL network is trained for each IMF and residue. The outputs of all sub-series are combined and analyzed by another RVFL to formulate the final prediction results. Figure 9.1 is the schematic diagram of the pre-training phase of this proposed ensemble method, whose procedures are as follows:

1. Apply MODWT to decompose the original TS into several frequency components.
2. Apply EMD to decompose each frequency component into several IMFs and one residue.
3. Construct the training matrix as the input of each RVFL network for each obtained sub-series. Then train an RVFL network for each of the extracted IMF and residue.
4. Aggregate the prediction results of all the sub-series, along with the lagged closing price values and the corresponding technical indicators, to construct a new input matrix, which is used to train an SVR to formulate the final prediction results.

After the pre-training phase, the learned ensemble model including the pseudoinverse of A_n and the weight matrix W_n for the incremental RVFLs can be updated by the incremental learning phase:

1. When a new input sample is given to the network, MODWT and EMD can be applied to decompose the signal and obtain new input pattern A_{n+1} by combining all the new outputs from all sub-series. Then the new weight matrix W_{n+1} can be updated using equation 5.5.
2. The validation data is applied to check the error level. If the error decreases, we keep the updates, otherwise the weight matrix is not changed.
3. Repeat steps 1 and 2 whenever new samples are presented to the network.

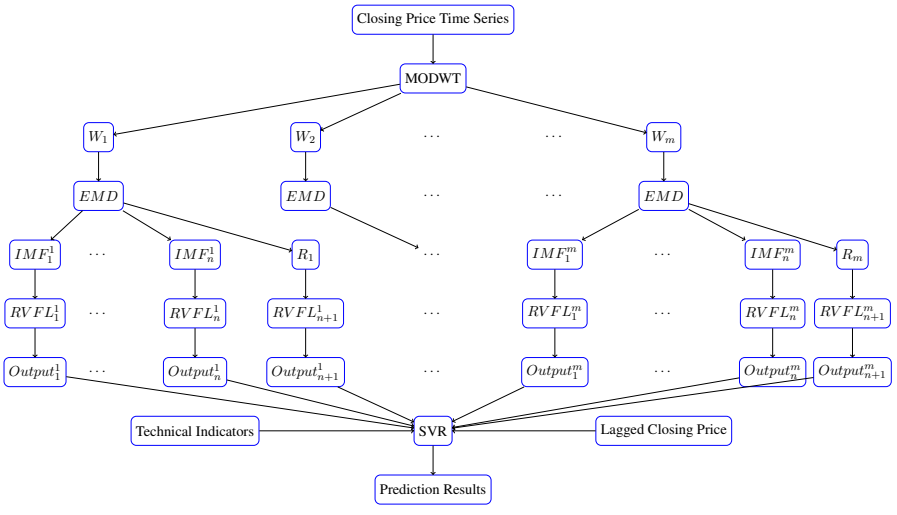


FIGURE 9.1: Schematic Diagram of the Proposed DWT-EMD based Incremental RVFL-SVR Model

9.3 Experiment setup

In this work, the stock price datasets of power related companies were used for evaluating the performance of benchmark learning models. Daily stock market prices for Chevron (from $03/01/2007$ to $30/12/2016$), Surgutneftegas (from $31/12/2007$ to $2/9/2016$), Lukoil (from $18/11/1996$ to $13/1/2017$), Exxonmobil (from $3/1/1993$ to $3/31/2017$) and PetroChina (from $03/01/2007$ to $28/06/2016$) were extracted from Yahoo Finance [207]. For each dataset, to compare the performance of learning models with different forecasting horizons, three kinds of simulations were conducted: one day ahead, two days ahead and one week ahead forecasting. Moreover, 80% of the data points in each dataset were used for training, while the remaining 20% was used for testing.

For the stock price datasets, all training values are linearly scaled to $[0, 1]$. The scaling formula is:

$$\bar{y}_i = \frac{y_{max} - y_i}{y_{max} - y_{min}} \tag{9.1}$$

To implement the simulation, LIBSVM toolbox was used for SVR based models, including SVR and EMD-SVR. Neural network toolbox in Matlab was used for constructing neural networks, including ANN and EMD based ANN (EMD-ANN). Moreover, RVFL, EMD-RVFL and the proposed incremental DWT-EMD-RVFL-SVR were developed by the authors in Matlab based on the work in [56].

For SVR based models, we use the RBF kernel function with parameters chosen by a grid search. The range of C is $[2^{-4}, 2^4]$, and the range of σ is $[10^{-3}$,$10^{-1}]$. For ANN and EMD-ANN, the size of neural networks is determined by the size of input vector. The number of iterations for back propagation is searched within [200, 2000], and optimized around 1000. For RVFL, EMD-RVFL and the proposed method, according to suggestion in [54, 56], the randomization used a uniform distribution in $[-1, 1]$, the number of hidden neurons is selected over $1000 : 10000$ with a step-size of 1000.

9.4 Results and Discussion

9.4.1 Performance Comparison with Benchmark Models

To evaluate the performance of the proposed DWT-EMD based incremental RVFL-SVR model, seven benchmark methods were implemented for stock market price forecasting to perform a comparison. The persistence method was employed as the baseline, which assumes the conditions at the future time similar to the current values and thus has relatively reasonable accuracy for TS forecasting. Table 9.2 shows the prediction results for short-term stock price forecasting.

In this work, three forecasting horizons are investigated: one day, two days and one week. The numbers in bold mean that the corresponding method has the best performance for this dataset under this performance measure. From Table 9.2, all the benchmark learning models can be proved to outperform the persistence method for short-term stock price forecasting.

To examine the effectiveness of EMD, we compare the single structure models with their EMD fusion models. From the results shown in Table 9.2, we can conclude that EMD can improve the performance of short-term stock price forecasting significantly. Besides that, SVR generally performs better than ANN and RVFL in all cases. Moreover, the proposed DWT-EMD-RVFL-SVR achieves the best performance in most of cases, thereby outperforms the benchmark models.

TABLE 9.2: Prediction results for stock market price forecasting

Dataset	Horizon	Metrics	Prediction model							
			Persistence	ANN [158]	RVFL [52, 53]	SVR [27]	EMD-ANN [169]	EMD-RVFL [179]	EMD-SVR [168]	DWT-EMD-RVFL-SVR (Proposed)
Chevron	1 day	RMSE	2.005	2.040	2.001	1.542	1.335	1.234	0.996	**0.952**
		MAPE	1.728%	1.757%	1.727%	1.327%	1.198%	1.082%	0.864%	**0.849%**
	2 days	RMSE	2.592	2.571	2.562	1.987	1.602	1.589	1.389	**1.378**
		MAPE	2.277%	2.265%	2.259%	1.808%	1.453%	1.356%	**1.195%**	1.197%
	1 week	RMSE	3.658	3.632	3.623	3.489	3.098	2.945	2.563	**2.362**
		MAPE	3.260%	3.224%	3.246%	3.027%	2.587%	2.451%	2.201%	**2.019%**
Lukoil	1 day	RMSE	1.305	0.987	0.956	0.948	0.535	0.550	0.451	**0.432**
		MAPE	2.106%	1.643%	1.601%	1.523%	0.861%	0.932%	0.733%	**0.701%**
	2 days	RMSE	1.588	1.335	1.325	1.305	0.701	0.706	0.704	**0.685**
		MAPE	2.608%	2.218%	2.198%	2.117%	1.123%	1.115%	1.114%	**1.089%**
	1 week	RMSE	2.179	2.105	2.098	2.063	1.167	1.153	0.905	**0.878**
		MAPE	3.623%	3.502%	3.489%	3.463%	1.956%	1.966%	1.482%	**1.354%**
Exxonmobil	1 day	RMSE	1.337	1.002	1.011	0.960	0.707	0.859	0.493	**0.485**
		MAPE	1.130%	0.852%	0.861%	0.813%	0.617%	0.749%	0.424%	**0.420%**
	2 days	RMSE	1.635	1.377	1.356	1.338	0.816	0.912	**0.665**	**0.665**
		MAPE	1.389%	1.120 %	1.134%	1.114%	0.716%	0.769%	**0.570%**	0.571%
	1 week	RMSE	2.161	2.151	2.041	1.988	1.277	1.131	0.905	0.893
		MAPE	1.882%	1.755%	1.743%	1.731%	1.037%	1.024%	0.774%	**0.763%**
Surgutneftegas	1 day	RMSE	18.762	18.467	18.479	13.260	12.134	13.191	7.686	**7.456**
		MAPE	2.846%	2.585%	2.584%	1.933%	1.843%	1.922%	1.241%	**1.125%**
	2 days	RMSE	22.628	22.357	20.876	18.325	14.547	14.407	9.162	**8.788**
		MAPE	3.389%	3.243%	3.123%	2.576%	2.223%	2.250%	1.288%	**1.264%**
	1 week	RMSE	36.038	32.329	31.267	30.246	16.934	16.888	16.833	**15.596**
		MAPE	5.764%	4.557%	4.578%	4.469%	2.697%	2.691%	2.374%	**2.288%**
PetroChina	1 day	RMSE	5.364	4.985	4.554	4.437	4.393	4.387	4.343	**4.249**
		MAPE	2.604%	2.512%	2.446%	2.447%	2.238%	2.239%	2.012%	**1.959%**
	2 days	RMSE	7.013	6.492	6.507	5.830	5.212	5.020	**4.420**	4.432
		MAPE	3.577%	3.163%	3.147%	2.781 %	2.576%	2.587%	**2.302%**	2.401%
	1 week	RMSE	11.490	9.873	9.652	9.365	8.765	6.845	6.895	**6.798**
		MAPE	5.271%	4.995%	4.813%	4.324%	4.032%	3.223%	**3.015%**	3.022%

Friedman test [197] and Nemenyi post-hoc test [198] are employed to test the significance of the differences among all the benchmark learning models. For Friedman test, the algorithms are ranked for each dataset separately. The null-hypothesis states that all the algorithms have the same performance. If p value is small enough, it means that the null-hypothesis can be rejected. In this case, we apply the Nemenyi post-hoc test to compare all the benchmark learning models with each other. As a result, we can tell whether the performances of two among totally k learning models are significantly different.

Figure 9.2 and Figure 9.3 show the comparison results of statistical test based on RMSE and MAPE, respectively. The methods with better ranks are at the top whereas the methods with worse ranks are at the bottom. The models have statistically the same performance if they are connected by a vertical line whose length is less than or equal to a critical distance . The critical distance can be calculated as:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{9.2}$$

where N is the number of data sets, k is the number of algorithms, and q_α is the critical value based on the Studentized range statistic divided by $\sqrt{2}$ [159]. The statistical test results show that the proposed DWT-EMD-RVFL-SVR achieves the best rank and significantly outperforms the non-EMD based methods with a 95% confidence.

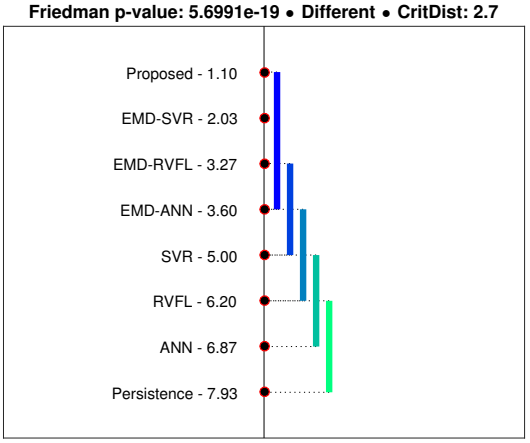


FIGURE 9.2: Nemenyi test for stock price forecasting based on RMSE. The critical distance is 2.7.

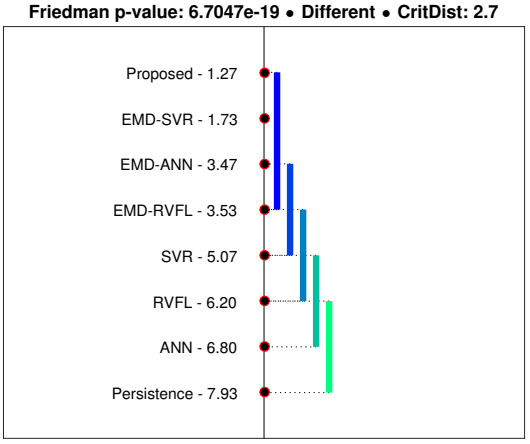


FIGURE 9.3: Nemenyi test for stock price forecasting based on MAPE. The critical distance is 2.7.

9.4.2 Computation time comparison

The computation time of benchmark methods for stock price forecasting using these five stock datasets is shown in Figure 9.4. It is easy to conclude that RVFL is much faster than ANN and SVR. The reason is that ANN needs to be iteratively tuned by BP algorithm to convergence to

the optimal weights, while RVFL has a closed form solution. Benefit from the good efficiency of RVFL, the proposed fusion model also has a reasonable fast computation speed.

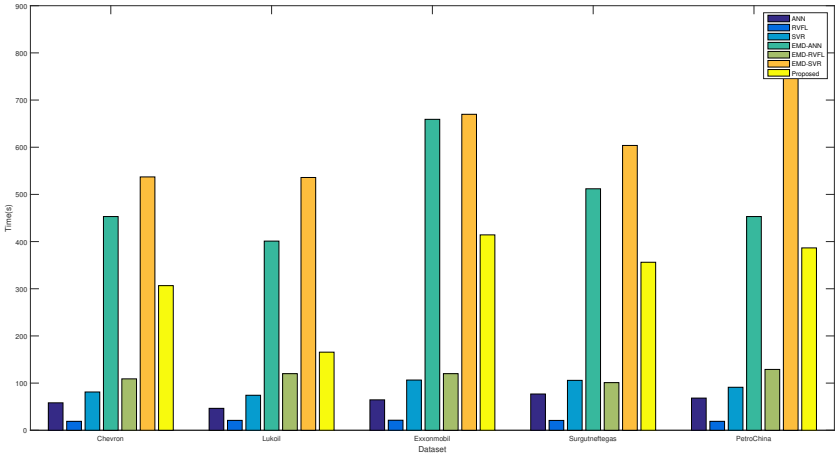


FIGURE 9.4: Computation time of learning models for stock price forecasting

9.5 Summary

In this chapter, a fusion incremental learning approach is presented for short-term stock price forecasting, which composed of Discrete Wavelet Transform (DWT), Empirical Mode Decomposition (EMD), Random Vector Functional Link (RVFL) network and Support Vector Regression (SVR). Besides the historical data of stock market closing price, ten indicators are also included to improve the performance of the fusion model. Five stock price datasets of power related companies were used for evaluating the performance of the proposed method by comparing with several benchmarks. Moreover, two comparative experiments were also implemented to verify the effectiveness of the proposed method. Based on the experimental results, the following conclusions are made:

1. Incremental learning RVFL is beneficial for short term stock price forecasting based on both accuracy and efficiency.

2. The proposed DWT-EMD decomposition method based fusion learning models outperform EMD based and single structure models.

3. The proposed incremental DWT-EMD based RVFL-SVR approach achieves the best rank and significantly outperforms the non-EMD based benchmarks with a 95% confidence.

Chapter 10

Summary of Part II

In the second part of this thesis, ensemble learning for financial markets related time series forecasting are investigated and discussed. Especially, two kinds of financial time series forecasting problems are considered: electricity price and stock price forecasting. In Chapter 8, an ensemble kernel machine is proposed for short-term electricity price forecasting, which is composed of EMD, KRR and SVR. In Chapter 9, a fusion incremental learning approach is presented for short-term stock price forecasting, composed of Discrete Wavelet Transform (DWT), Empirical Mode Decomposition (EMD), Random Vector Functional Link (RVFL) network and Support Vector Regression (SVR). Besides the historical data of stock market closing price, ten indicators are also included to improve the performance of the fusion model. The simulation results demonstrate the advantages of ensemble learning methods for financial market related time series forecasting.

Similar with Chapter 7, in this chapter, we also conduct an overall comparison among all the ensemble learning methods mentioned in this thesis for short term stock price forecasting. The ensemble learning methods include oblique RF, EMD based ensemble deep belief network (EMD-DBN), EMD based SVR (EMD-SVR), EMD based ANN (EMD-ANN), EMD based RF (EMD-RF), EMD based RVFL (EMD-RVFL), EMD-KRR-SVR, DWT-EMD-RVFL, and DWT-EMD-RVFL-SVR. We use the same datasets in Chapter 9, which are daily stock market price datasets for Chevron (from $03/01/2007$ to $30/12/2016$), Surgutneftegas (from $31/12/2007$ to $2/9/2016$), Lukoil (from $18/11/1996$ to $13/1/2017$), Exxonmobil (from $3/1/1993$ to $3/31/2017$) and PetroChina (from $03/01/2007$ to $28/06/2016$) from Yahoo Finance [207]. For each dataset, to compare the performance of learning models with different forecasting horizons, three kinds of simulations are conducted: one day ahead, two days ahead and one week ahead forecasting. Moreover, 80% of the data points in each dataset are used for training, while the remaining 20% is used for testing. The prediction results for short term stock price forecasting

are shown in Table 10.1. Moreover, the Friedman test and Nemenyi post-hoc test are also applied to rank the ensemble learning models. The comparison results of statistical test based on RMSE and MAPE are shown in Figure 10.1 and Figure 10.2, respectively.

TABLE 10.1: Prediction results for stock market price forecasting

Dataset	Horizon	Metrics	Prediction model								
			ORF [77]	EMD-DBN [16]	EMD-SVR [168]	EMD-ANN [169]	EMD-RF [178]	EMD-RVFL [179]	EMD-KRR-SVR [80]	DWT-EMD-RVFL	DWT-EMD-RVFL-SVR
Lukoil	1 day	RMSE	0.583	0.447	0.451	0.535	0.534	0.550	0.435	0.441	0.432
		MAPE	1.1828%	0.6448%	0.7338%	0.8618%	1.0658%	0.9328%	0.7048%	0.6708%	0.7018%
	2 days	RMSE	0.768	0.699	0.704	0.701	0.671	0.706	0.744	0.689	0.685
		MAPE	1.350%	1.110%	1.114%	1.123%	1.232%	1.115%	1.232%	1.045%	1.089%
	1 week	RMSE	1.565	0.892	0.905	1.167	1.284	1.153	1.029	0.851	0.878
		MAPE	2.406%	1.482%	1.482%	1.956%	2.209%	1.966%	1.446%	1.509%	1.354%
Surgutneftegas	1 day	RMSE	13.968	8.262	7.686	12.134	12.754	13.191	8.176	8.119	7.456
		MAPE	2.146%	1.208%	1.241%	1.843%	1.980%	1.922%	1.295%	1.140%	1.125%
	2 days	RMSE	15.814	9.158	9.162	14.547	14.435	14.407	8.726	9.019	8.788
		MAPE	2.840%	1.176%	1.288%	2.223%	2.498%	2.250%	1.310%	1.210%	1.264%
	1 week	RMSE	19.465	15.901	16.833	16.934	17.501	16.888	16.537	16.185	15.596
		MAPE	3.762%	2.414%	2.374%	2.697%	3.047%	2.691%	2.632%	2.348%	2.288%
Exxonmobil	1 day	RMSE	0.963	0.534	0.493	0.707	0.847	0.859	0.499	0.535	0.485
		MAPE	0.816%	0.480%	0.424%	0.617%	0.751%	0.749%	0.448%	0.466%	0.420%
	2 days	RMSE	1.099	0.636	0.665	0.816	0.893	0.912	0.654	0.657	0.665
		MAPE	0.936%	0.584%	0.570%	0.716%	0.751%	0.769%	0.610%	0.598%	0.571%
	1 week	RMSE	1.446	0.906	0.905	1.277	1.271	1.131	0.907	0.898	0.893
		MAPE	1.170%	0.790%	0.774%	1.037%	1.092%	1.024%	0.837%	0.783%	0.763%
PetroChina	1 day	RMSE	5.123	4.341	4.343	4.393	4.650	4.387	4.725	4.639	4.249
		MAPE	2.481%	2.005%	2.012%	2.238%	2.192%	2.239%	2.212%	2.157%	1.959%
	2 days	RMSE	6.576	4.412	4.420	5.212	5.625	5.020	4.597	4.455	4.432
		MAPE	3.064%	2.261%	2.302%	2.576%	2.780%	2.587%	2.225%	2.507%	2.401%
	1 week	RMSE	8.174	7.188	6.895	8.765	6.983	6.845	6.866	7.149	6.798
		MAPE	4.045%	2.830%	3.015%	4.032%	3.393%	3.223%	3.415%	2.969%	3.022%
Chevron	1 day	RMSE	1.391	0.889	0.996	1.335	1.271	1.234	0.977	0.927	0.952
		MAPE	1.121%	0.857%	0.864%	1.198%	1.044%	1.082%	0.964%	0.891%	0.849%
	2 days	RMSE	1.759	1.499	1.389	1.602	1.586	1.589	1.469	1.374	1.378
		MAPE	1.472%	1.136%	1.195%	1.453%	1.322%	1.356%	1.373%	1.119%	1.197%
	1 week	RMSE	3.298	2.621	2.563	3.098	2.906	2.945	2.475	2.520	2.362
		MAPE	2.817%	2.089%	2.201%	2.587%	2.446%	2.451%	2.286%	2.008%	2.019%

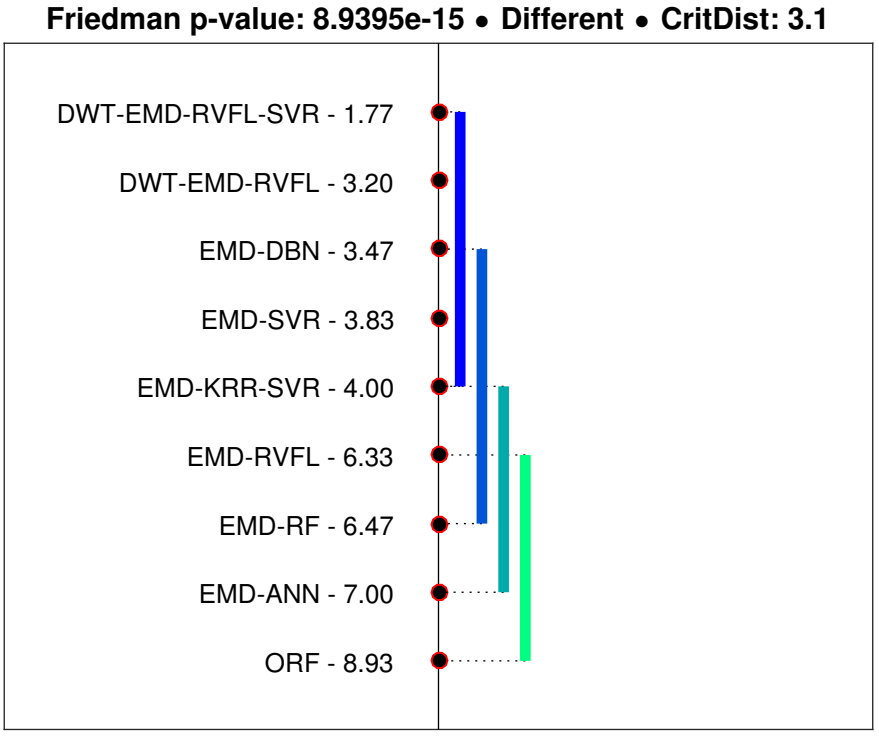


FIGURE 10.1: Nemenyi test for stock price forecasting based on RMSE. The critical distance is 3.1.

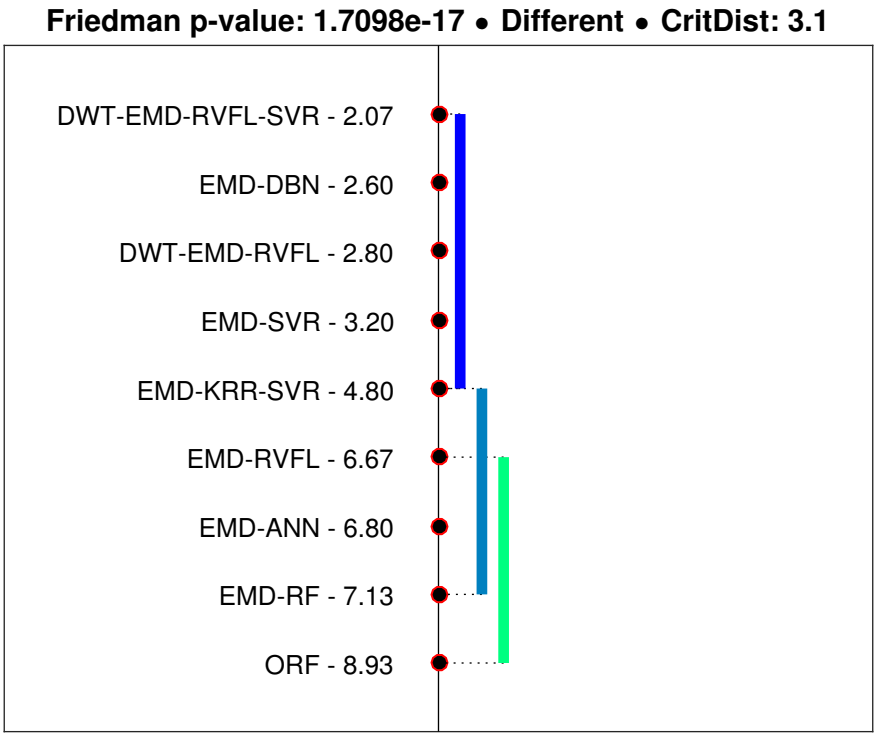


FIGURE 10.2: Nemenyi test for stock price forecasting based on MAPE. The critical distance is 3.1.

From the simulation and statistical testing results, several conclusions can be made about ensemble learning for short term stock price forecasting.

1. There exists significant differences among these ensemble learning methods based on the sufficient small Friedman p-value,

2. Ensemble deep learning method EMD-DBN achieves second place among all the ensemble learning models, which is only beaten by an ensemble increment learning model. This phenomenon demonstrates the attractiveness and potential of ensemble deep learning models for financial markets related time series forecasting.

3. Ensemble incremental learning is beneficial for short term stock price forecasting, which makes use of base learners to achieve comparable performance with deep learning models.

4. The incremental DWT-EMD-RVFL-SVR approach achieves the best rank and significantly outperforms EMD-RVFL, EMD-ANN, ORF and EMD-RF with a 95% confidence.

Part III

Surrogate for Chemical Plant Process Flow Modelling

Chapter 11

Machine learning approach for constructing surrogates of a biodiesel plant flow sheet model

Nowadays, it is more and more important for industries to have a better understanding, and thus optimization of daily operation activities, which can bring in the benefits like minimizing resource use and maximizing profit, as well as protecting the environment. Industrial 4.0 is a possible solution to achieve the goal with low cost, by constructing surrogate of physical models restricted to a predefined range of inputs, which have the same ability of industrial components to communicate with each other. There are huge amount of time series data generated by industrial systems, such as manufacturing machines, chemical plant, etc Constructing surrogate models is one of the important way to analyze such industrial systems. Based on the definition, regression includes two categories: both forecasting and function approximation. In this chapter, the surrogate models are trained as regression models using a number of input variables without considering time sequence dependence. For the future research works, time series forecasting/classification models will also be employed to help improve the surrogate models.

11.1 Introduction

As environmental concerns become more and more pressing, significant academic and industrial interest has been focused on the various ecologically friendly targets, such as reducing wastes and pollutants, reducing carbon footprints and creating cleaner manufacturing processes. As a result, eco-industrial parks (EIPs) have attracted public interests and became popular with the development of the concepts of “industrial ecology” [208], “industry symbiosis” [209], and “sustainable development” [210].

For the concept of EIP, as an industrial park, the businesses cooperate with each other, as well as the local community to reduce pollution and waste, efficiently share resources (such as information, materials, water, energy, infrastructure, and natural resources), and minimise environmental impact while simultaneously increasing business success [1]. For example, in Kalundborg, Denmark, a typical EIP has been established, which links a $1500MW$ coal-fired power plant with the community and other companies [209, 211]. In this EIP, the examples of resources exchanging include selling the steam from the power plant to a pharmaceutical and enzyme manufacturer named Novo Nordisk, and using surplus heat to heat 3500 local homes and a nearby fish farm. Meanwhile, the sludge from that farm is sold as a fertilizer. Furthermore, the waste from the power plant, which includes fly ash and clinker, is utilized for road building and cement production [212].

In recent years, numerous research works concerning various aspects of EIPs have been published, which mainly focus on optimal design of sustainable industrial activities by constructing mathematical models to create exchange networks of resources among the members of EIPs [213–216]. However, it is very difficult and expensive to perform the holistic modelling of complex and highly interconnected networks, such as EIPs, which include many physical models with disparate processes. This problem may be overcome by the concept of Industry 4.0 [1], constructing surrogate models of physical models restricted to a predefined range of inputs, which have the same ability of industrial components to communicate with each other. The surrogate models would make dynamic modelling and studies possible, and help reduce the computation time and memory significantly. For example, in [217], surrogate models are designed using ANN for building shell energy labelling. The experimental results indicate that ANN can represent the interaction between input and output data for a vast and diverse building stock. In [218], a global optimization method for general constrained grey-box models is presented, and is applied to the application of pressure swing adsorption. During the optimization procedures, the surrogate models are constructed to expedite the search towards optimal solutions. Figure 11.1 is the framework of EIP modelling based on Industry 4.0.

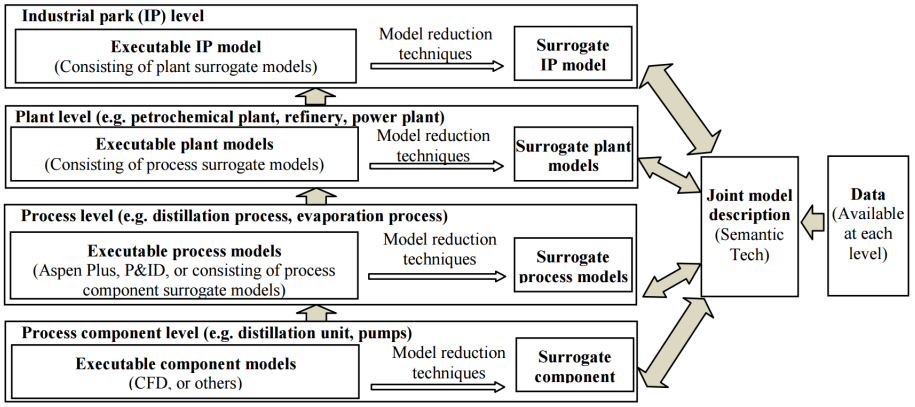


FIGURE 11.1: Framework of EIP modelling based on Industry 4.0. Adopted from [1].

A surrogate model is defined as an approximation of experimental and/or simulation data designed to provide answers when it is too expensive to directly measure the outcome of interest [219]. Due to the above motivation, there are two key requirements for the surrogate models: reasonable accuracy and significantly faster evaluation than physical model. The surrogate models can be used to obtain the nature of the input-output relationship, deal with noise or missing data, as well as help design and optimize experiments.

In the literature, surrogate models are widely applied in various research fields in engineering and science, which include modelling [220], sensitivity analysis [221], space exploration [222], parameter estimation [223], optimisation in areas ranging from circuit design through nanoparticle synthesis to flood monitoring [224]. Model selection is the most important part for constructing a surrogate model. There are numerous sampling and fitting techniques in the literature, which include response surface [225], polynomial fitting [2], kriging [226], artificial neural network (ANN) [227], support vector machine (SVM) [219] and so on. In [225], detailed reviews of data sampling and meta-model generation techniques are provided, along with the discussion about error measures including R^2, residue plots and root mean square error. In [219], numerous surrogate models are discussed with engineering case studies, including response surfaces, kriging, SVR and radial basis functions.

In [2], surrogate models are constructed for parametrisation of typical input-output relations within process flow sheet of a biodiesel plant. Two different surrogates are considered: polynomial response surfaces and high dimensional model representation (HDMR) fitting. Moreover, the effects of dimensionality, domain size, and surrogate type on the accuracy of surrogate models are also investigated in a variety of scenarios. In this paper, accurate surrogate models are constructed using machine learning algorithms, instead of polynomial and HDMR fitting, to analyze the relationship between 11 inputs and 6 outputs in a typical biodiesel plant. The same simulation scenarios as in [2] are considered: $1, 2, 6$ and 11 input variables and 3 domain sizes. Totally 5 different machine learning techniques (support vector regression (SVR), artificial neural network (ANN), deep belief network (DBN), random forests (RF) and random vector functional link network (RVFL)) are used for constructing surrogates. The performances of these machine learning algorithms are discussed according to both accuracy and efficiency.

The contribution of this work can be classified into three aspects. First of all, in our previous work [2], two different surrogate models are constructed for parametrization of typical input-output relations within process flow sheet of a biodiesel plant: polynomial response surfaces and high dimensional model representation (HDMR) fitting, as well as investigating the effects of dimensionality, domain size, and surrogate type on the accuracy of surrogate models. In this paper, we extend our work by exploring better surrogate models using machine learning algorithms instead of polynomial and HDMR fitting. Secondly, five different machine learning methods are employed and compared, especially including complicated structure models (deep

learning approaches) and efficient randomized neural networks (RVFL), which attempt to investigate the performance of machine learning methods for biodiesel plant flow sheet modeling based on both accuracy and efficiency. Last but not least, the best surrogate model constructed for biodiesel process modelling has been given with the optimal hyper-parameters, which can be treated as the benchmark for the future research on the same topic.

11.2 Experiment Setup

11.2.1 Data collection

In this study, for data collection and visualization, we used Model Development Suite (MoDS) [228] and custom-made Python 3.4 and R 3.2.2 scripts. Figure 11.2 shows the workflow of MoDS. We investigate four different dimension of input variables: $1, 2, 6$ and 11 in 3 different domain sizes. Totally 5 different machine learning methods were employed and compared. There are 400 points per input variable in each simulation to ensure that the number of training samples is sufficient. The training part was evaluated by R^2 and $\bar{R}^2$. Moreover, there are 100 data points per dimension for testing, which were measured by Root Mean Square Deviation (RMSD) and residuals. The definitions of input and output variables are summarized in Table 11.1 and Table 11.2, respectively.

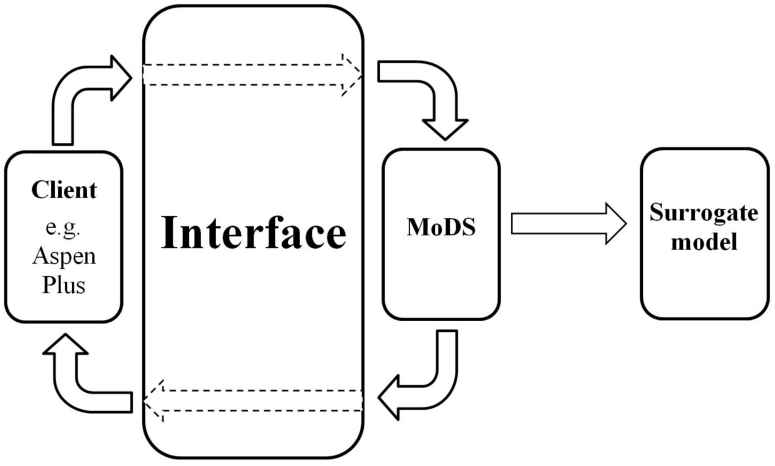


FIGURE 11.2: Model Development Suite work flow. Adopted from [2].

11.2.2 Data normalization

Before the machine learning algorithms being used to construct the surrogate models, all the training and testing values are linearly scaled to $[0, 1]$. The scaling formula is:

TABLE 11.1: Definitions and domain bounds of input variables

Name	Lower bounds	Upper bounds	Operation point
Molar flow of tripalmitine oil (kmol/h)	20, 22.5, 25	40, 37.5, 35	30
Temperature of tripalmitine oil (°C)	20, 22.5, 25	40, 37.5, 35	30
Operation temperature of CSTR 10D01 (°C)	44, 49, 54	64	60
Volume of CSTR 10D01 (m^3)	40, 43, 45	50, 49, 47	45
Operation temperature of flash drum 10D02 (°C)	80, 82.5, 85	100, 97.5, 95	90
Operation temperature of heater 10E01 (°C)	60, 62.5, 65	80, 77.5, 75	70
Molar flow of methanol (kmol/h)	150, 160, 170	210, 200, 190	180
Temperature of methanol (°C)	20, 22.5, 25	40, 37.5, 35	30
Operation temperature of decanter 10D02D (°C)	20, 22.5, 25	40, 37.5, 35	30
Operation temperature of heater 10E02 (°C)	80, 82.5, 85	100, 97.5, 95	90
Operation temperature of heater 10E03 (°C)	60, 62.5, 65	80, 77.5, 75	70

TABLE 11.2: Definitions of Output variables

Name
Heat duty of heater 10E01 (MW)
Heat duty of heater 10E02 (MW)
Heat duty of heater 10E03 (MW)
Heat duty of reactor 10D01 (MW)
Heat duty of flash drum 10D02 (MW)
Heat duty of decanter 10D02D (MW)

$$\bar{y}_i = \frac{y_{max} - y_i}{y_{max} - y_{min}} \tag{11.1}$$

11.2.3 Performance estimation

In this study, a number of error measures are used to evaluate the performance of surrogate models: R^2, root-mean-squared-deviation (RMSD) and residual plots. They are defined as follows:

$$\begin{aligned} RMSD &= \sqrt{\frac{1}{l_s}\sum_{j=1}^{l_s}(y'_j - y_j)^2} \\ R^2 &= 1 - \frac{\sum_{i=1}^{l}(y_i - y'_i)^2}{\sum_{i=1}^{l}(y_i - \bar{y})^2} \\ e_j &= y_j - y'_j \end{aligned} \tag{11.2}$$

where y'_i and y'_j are the predicted values of corresponding training data y_i and testing data y_j, respectively; $\bar{y}$ is the empirical mean of training data points, l is the number of training data samples, l_s is the number of testing data points, e_j is the residual for j^{th} testing data point, $i = 1, \ldots, l$ and $j = 1, \ldots, l_s$.

11.3 Results and Comparison

In this work, four scenarios with different input dimension are considered: 1, 2, 6 and 11 input variables. The surrogate models are constructed by five machine learning techniques: SVR, ANN, RF, DBN and RVFL. Same with [2], R^2 values were calculated using the training set to assess fit of the surrogates to the training data, while RMSD and residual plots were generated using the testing data.

11.3.1 Performance comparison using R^2 values for training data

R^2 is an error measure which compares the discrepancies between the predicted data and actual data with the discrepancies between the arithmetic average and actual data. Figure 11.3 shows the plots of R^2 values for the surrogates constructed for heat duty of reactor 10D01 with 11 input variables by various machine learning methods. As all the surrogate models perform quite well and achieve R^2 values higher than 0.99, it is very difficult to differentiate between the models by R^2 using training data.

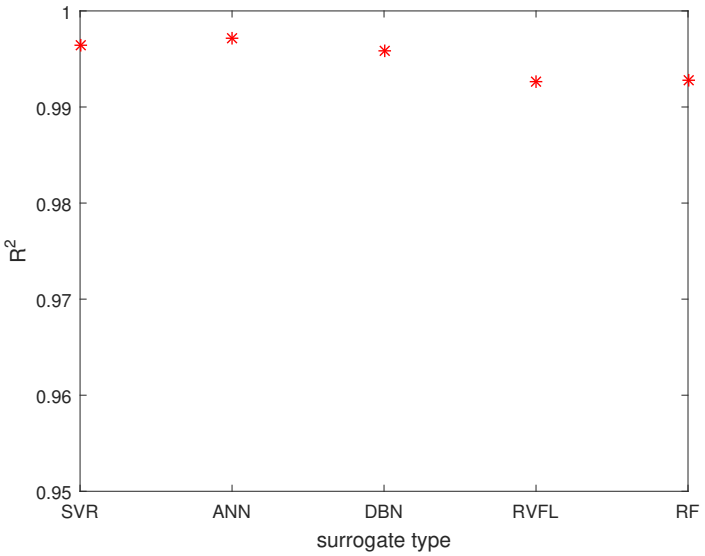


FIGURE 11.3: Plot of R^2 for the surrogate models

To take the model complexity into consideration, adjusted R^2 is also calculated, which is corrected for the number of fitted parameters relative to the number of data points. Here the number of data points is 4400 for the surrogates with 11 input variables. The results are shown in Figure 11.4. From the figure, we can clearly see that the adjusted R^2 values of SVR has been decreased significantly due to the high complexity of the model. However, all the models still achieve adjusted R^2 values higher than 0.985.

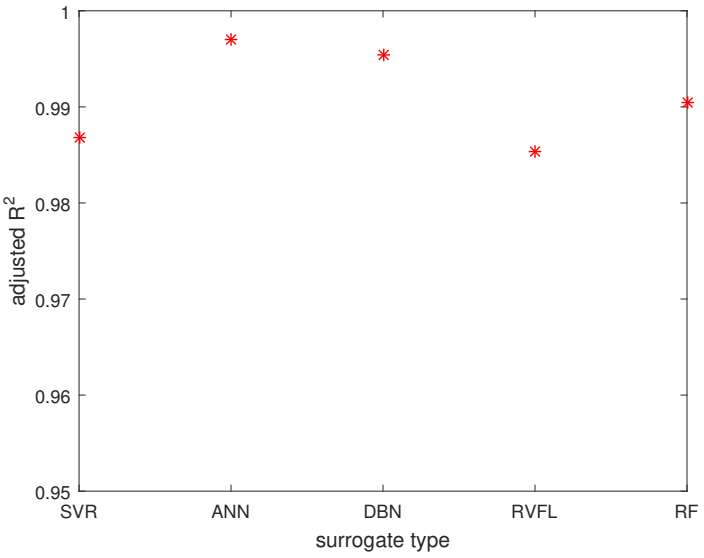


FIGURE 11.4: Plot of adjusted R^2 for the surrogate models

11.3.2 Performance comparison using RMSD values for testing data

The RMSD values of different machine learning algorithms with different input variables are recorded in Table 11.3, which can suggest a number of observations and conclusions. First of all, all the constructed surrogate models achieve at least a reasonable fit regardless of the domain size and number of dimensions, which proves the effectiveness of the benchmark machine learning models for surrogate fitting.

TABLE 11.3: Performance evaluation of surrogate models with RMSD

Dimension	Output	Surrogate model				
		SVR	ANN	DBN	RVFL	RF
11	10E01	**2.75E-04**	2.10E-03	1.99E-03	5.91E-03	1.69E-02
	10E02	1.43E-02	1.07E-02	**9.54E-03**	3.61E-02	5.27E-02
	10E03	**1.45E-03**	2.26E-03	3.17E-03	6.71E-03	2.15E-02
	10D01	**2.71E-03**	5.63E-03	6.51E-03	1.69E-02	3.16E-02
	10D02	1.86E-02	1.64E-02	**1.49E-02**	4.16E-02	4.17E-02
	10D02D	7.17E-03	6.93E-03	**6.13E-03**	1.18E-02	3.22E-02
6	10E01	**1.78E-04**	1.56E-03	7.63E-04	4.27E-03	1.42E-02
	10E02	**1.91E-04**	1.68E-03	1.40E-03	1.06E-02	1.35E-02
	10E03	**2.65E-04**	1.26E-03	1.07E-03	1.75E-03	6.64E-03
	10D01	**2.98E-04**	3.10E-03	3.00E-03	1.09E-02	2.35E-02
	10D02	**4.86E-03**	6.62E-03	5.77E-03	2.61E-02	1.80E-02
	10D02D	**1.02E-03**	3.19E-03	3.14E-03	7.71E-03	1.91E-02
2	10E01	**8.51E-05**	1.10E-03	9.50E-04	1.49E-03	4.83E-03
	10E02	**2.45E-04**	2.36E-03	1.94E-03	5.58E-03	1.13E-02
	10E03	**9.77E-05**	1.16E-03	1.01E-03	1.52E-03	5.25E-03
	10D01	**2.92E-04**	2.46E-03	2.42E-03	2.40E-03	1.48E-02
	10D02D	**2.62E-04**	2.64E-03	2.09E-03	2.44E-03	1.08E-02
1	10E01	**9.05E-05**	1.15E-03	1.62E-03	1.40E-03	2.24E-04
	10E02	**1.53E-04**	1.18E-03	1.66E-03	4.78E-04	2.50E-04
	10E03	**1.13E-04**	1.27E-03	1.80E-03	1.49E-03	2.39E-04
	10D01	**3.58E-04**	2.86E-03	4.10E-03	1.91E-03	5.96E-04
	10D02D	**1.88E-04**	2.56E-03	3.34E-03	2.87E-03	5.24E-04

Moreover, statistical tests are employed to give a detail analysis about the performance differences among all the learning models. The Friedman test ranks the algorithms for each dataset separately, and then assign average ranks in case of ties. The null-hypothesis states that all the algorithms have the same performance. If the null-hypothesis is rejected, in order to tell whether the performances of two among totally k learning models are significantly different, the Nemenyi post-hoc test is applied to compare all the learning models with each other. The comparison result of statistical test based on RMSD is shown in Figure 11.5.

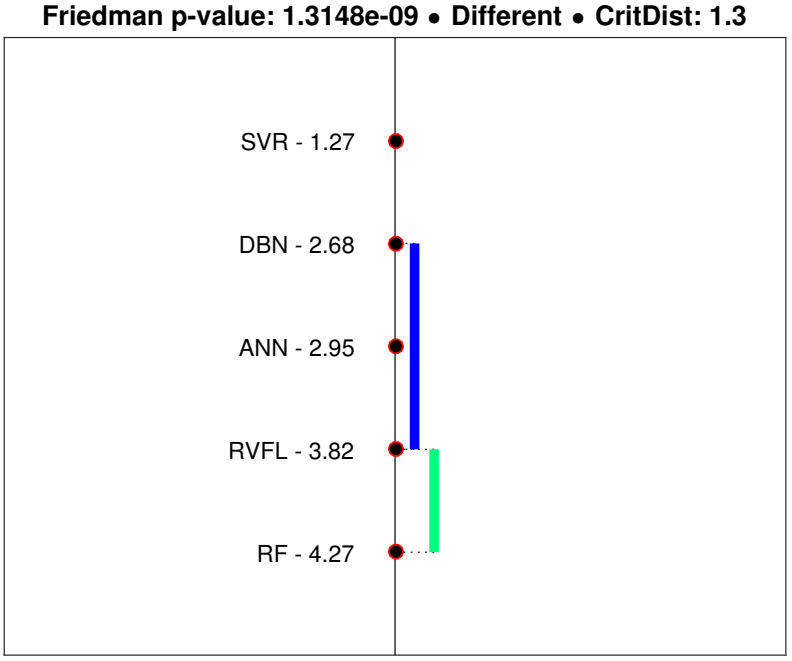


FIGURE 11.5: Nemenyi testing results for surrogate models based on RMSD. The models within a vertical line whose length is less than or equal to a critical distance have statistically the similar performance.

From the comparison results, several observations can be made. First of all, the SVR based surrogate model has achieved the best performance, followed by DBN and ANN. DBN, which belongs to deep learning category, has a similar performance compared with SLFN, and even performs worse in some cases. This phenomenon may be caused by the reason that the dataset is collected from a simulation system, which is not so complicated as a real plant system, and thus makes the deep neural networks overfitting the training data easily. Moreover, random forest, as a decision tree based ensemble method, has the limitation of accuracy for regression problems due to the reason that decision trees generate the predicted values from the mean or median of the samples in each leaf node. It is also worth noting that RVFL, as a non-iterative machine learning model with closed-form solutions, constructs surrogate models with reasonable accuracy and high efficiency, which will be proved in Section 11.3.4.

To have a comparison with the polynomial and HDMR fitting methods discussed in [2], the plots of RMSD values produced by different surrogates for heat duty of reactor 10D01 with respect to all 11 inputs are shown in Figure 11.6. Based on the results shown in [2], the polynomial fits of order 3 and HDMR model of 2^{nd} order with interactions, labeled as $P3$ and $H2b$, respectively, achieve the best performance among all the polynomial and HDMR variants. Hence we choose these two models to compare with machine learning methods. From the results shown in Figure 11.6, we can conclude that SVR, ANN and DBN significantly outperform polynomial and HDMR fitting for constructing surrogate models, while RVFL has similar performance with them.

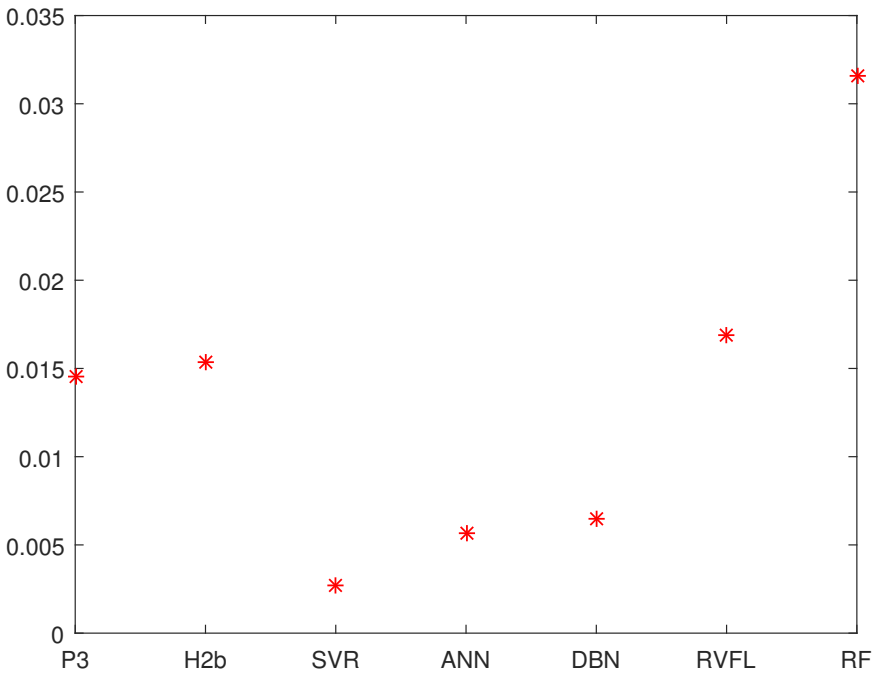
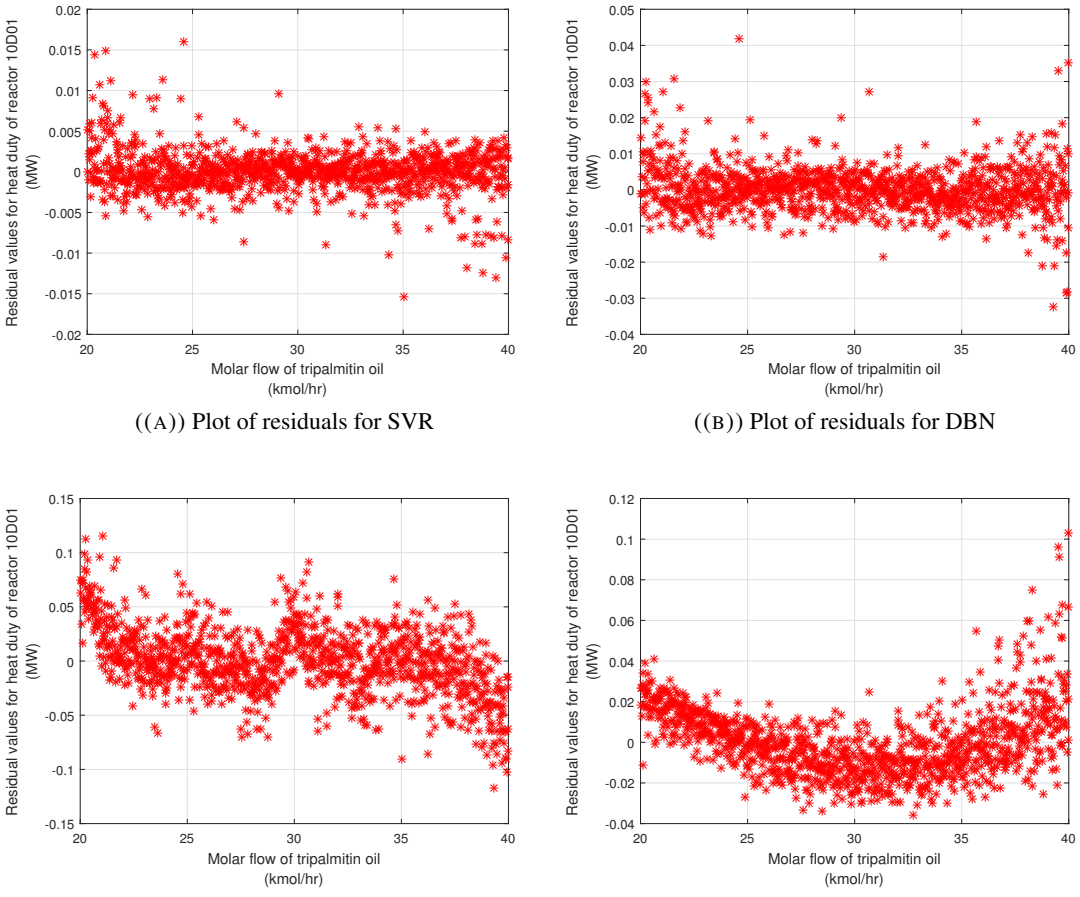


FIGURE 11.6: Plots of RMSD values produce by polynomial fitting, HDMR model and machine learning methods for heat duty (MW) of reactor 10D01 with respect to all 11 inputs.

11.3.3 Performance comparison using residual plots

As we have mentioned in Section 11.2.2, due to the ability to show the error size and distribution, residual plots are the most informative form of error measurement to understand whether the fit captures the true nature of the data. Figures 11.7 and 11.8 show the residue plots for 11-dimensional surrogates of heat duties of reactor 10D01 and heater 10E03, respectively. Meanwhile, for comparison, Figure 11.9 presents the residues plots for 1-dimensional surrogates of 10D01. Since the patterns of the residual plots are similar for surrogates constructed by DBN and ANN, only the residual plots for DBN are shown to simplify the comparison.

From Figures 11.7 and 11.8, we can see that all surrogate models do not follow a polynomial relation resulting in non-random distribution of the residuals for 11-dimensional inputs, which proves the effectiveness of machine learning models for constructing the surrogates. However, the plots for RF and RVFL still show some certain patterns for the distribution of the residues. Meanwhile, the comparison between Figures 11.7 and 11.9 shows that the non-random features are much more difficult to identify for surrogates with high dimensional input. Magnitude of residuals in all cases are relatively small indicating strong predictive powers of all the surrogate models. In conclusion, the residual plots confirm that SVR is one of the best methods for constructing surrogate models, followed by neural networks (e.g. ANN and DBN), while RF shows a stable performance ignoring the input dimension.



((A)) Plot of residuals for SVR

((B)) Plot of residuals for DBN

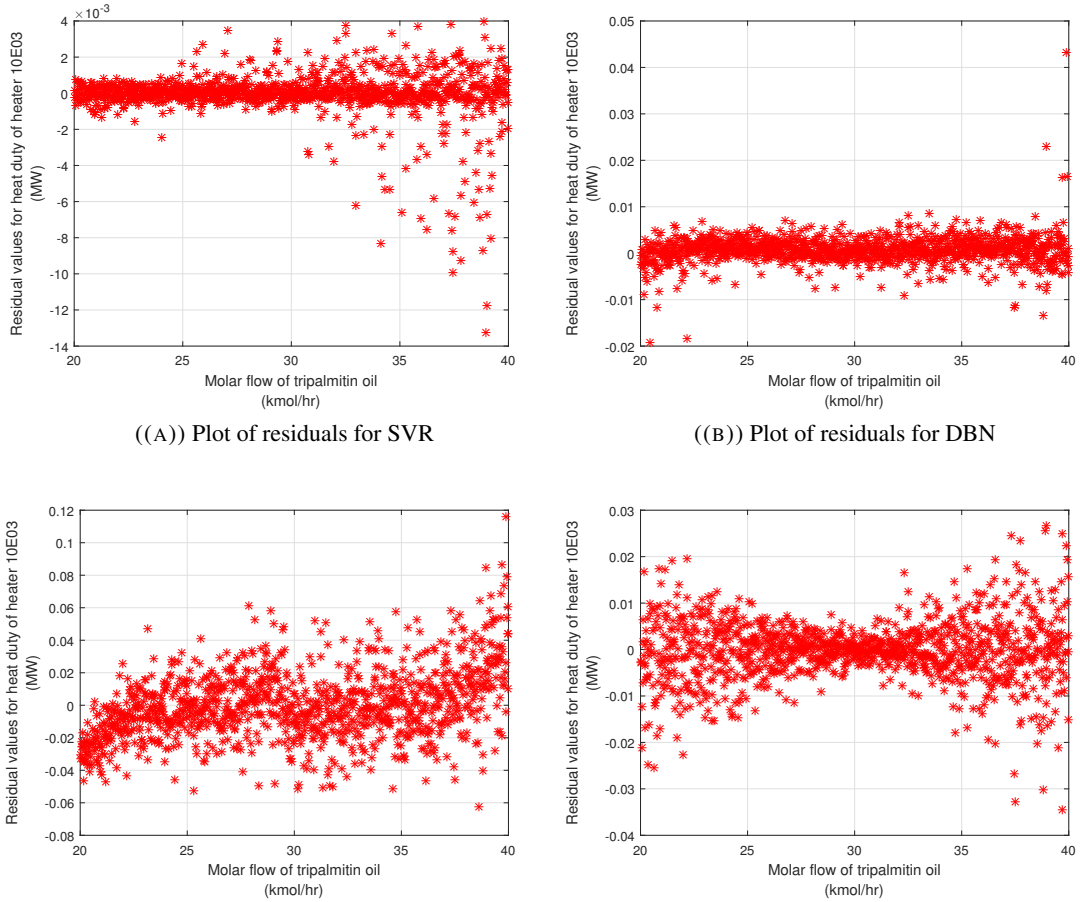
((C)) Plot of residuals for RF

((D)) Plot of residuals for RVFL

FIGURE 11.7: Plot of residuals against molar flow of tripalmitin oil for heat duty of reactor 10D01 produced for 11 inputs.

11.3.4 Computation time comparison

Figure 11.10(a) shows the computation time of benchmark machine learning methods for constructing 11-dimensional surrogates of heat duties of reactor 10D01, while the computation time



((A)) Plot of residuals for SVR

((B)) Plot of residuals for DBN

((C)) Plot of residuals for RF

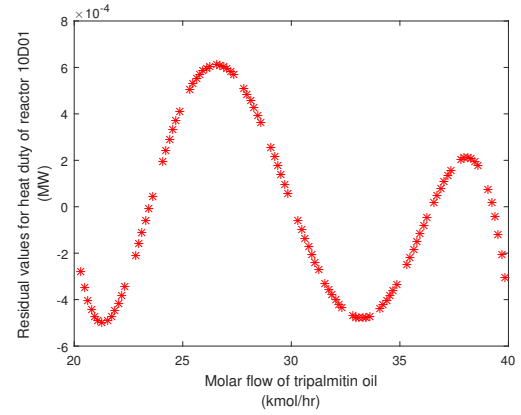
((D)) Plot of residuals for RVFL

FIGURE 11.8: Plot of residuals against molar flow of tripalmitin oil for heat duty of heater 10E03 produced for 11 inputs

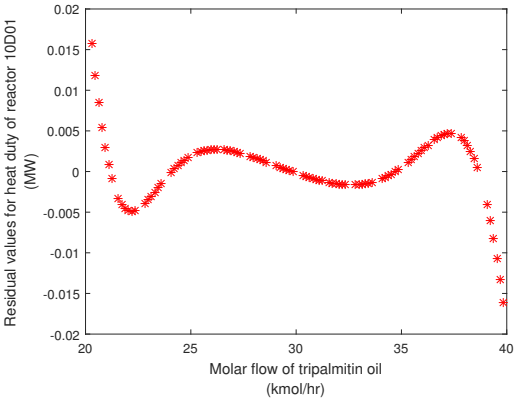
per evaluation is shown in Figure 11.10(b). Obviously, the computational speed of RVFL is superior than NNs and SVR. SVR requires a grid search on C and ϵ, and NNs are iteratively tuned by BP algorithm to convergence to the optimal weights. These repetitive parameter tuning processes cause NNs and SVR less efficient than RVFL, which has closed form solutions. Besides that, the RVFL based surrogate models can easily update the weights according to new input samples [170]. Therefore, RVFL is a good choice for surrogate models when the physical model is not very complicated, and high efficiency is required.

11.4 Summary for Surrogate Models

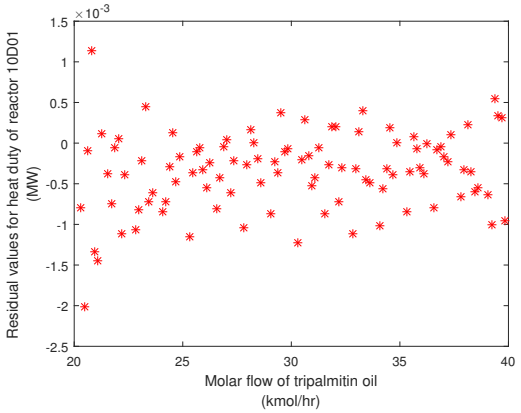
In this chapter, various machine learning techniques were investigated and employed for constructing surrogate models to analyze the input-output relations within process flow sheet of a biodiesel plant. The model under investigation includes a reaction and separation steps with



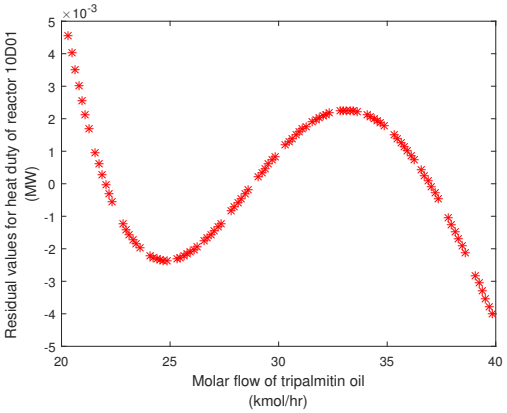
((A)) Plot of residuals for SVR



((B)) Plot of residuals for DBN

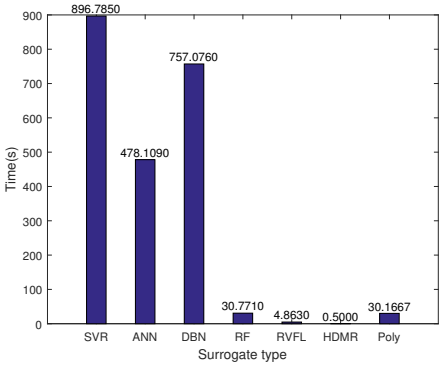


((C)) Plot of residuals for RF

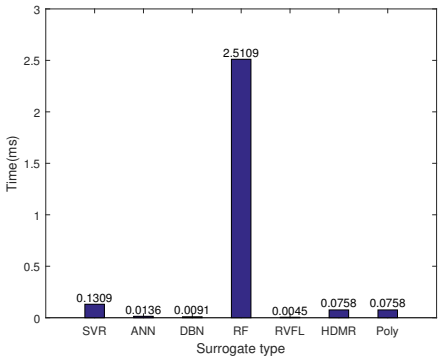


((D)) Plot of residuals for RVFL

FIGURE 11.9: Plot of residuals against molar flow of tripalmitin oil for heat duty of reactor 10D01 produced for 1 inputs.



((A)) Computation time for training the surrogate model



((B)) Computation time per evaluation

FIGURE 11.10: Training and evaluation time of learning models for constructing 11-dimensional surrogates of heat duties of reactor 10D01.

auxiliary equipment and was solved for steady-state operation. A variety of scenarios were considered: $1, 2, 6$ and 11 input variables were changed simultaneously, 3 domain sizes of input

variables were considered and 5 different surrogates (support vector regression (SVR), artificial neural network (ANN), deep belief network (DBN), random forests (RF), and random vector functional link network (RVFL)) were used. Each simulation produced 400 points per input variable used for training the surrogate models. Meanwhile, 100 points per dimension were generated for testing. The performance of surrogate models were evaluated by three error measures: R^2, RMSD and residuals. According to the simulation results, several observations can be summarized as follows:

1. According to both R^2 (with values in excess of 0.99) and RMSD (with very small values from 10^{-2} to 10^{-4}), all machine learning technique based surrogates achieve a good performance regardless of the domain size and number of dimensions.

2. From Figure 11.6 to 11.9, we can see that SVR, followed by DBN and ANN, achieves the best performance for constructing surrogate models in the context of biodiesel plant modeling, significantly outperforms polynomial and HDMR fitting, and also captures a random distribution of residuals for multiple dimensional surrogates.

3. DBN, which belongs to deep learning category, has a similar performance compared with ANN, and even performs worse in some cases. This phenomenon may be caused by the reason that the dataset is collected from a simulation system, which is not so complicated as a real plant system, and thus makes the deep neural networks overfitting the training data easily.

4. Random Forests, as a decision tree based method, has the limitation of accuracy for regression problems due to the reason that decision trees generate the predicted values from the mean or median of the samples in each leaf node. It is also worth noting that RF achieves relatively random distribution of residuals even for 1-dimensional surrogate.

5. The surrogate models constructed using RVFL have reasonable accuracy and high efficiency due to the reason that RVFL is a non-iterative machine learning model with closed-form solutions. Hence, RVFL is a good choice for surrogate models when the physical model is not very complicated, and high efficiency is required.

Chapter 12

Conclusions and Future Work

12.1 Conclusions

This thesis has addressed one of the most attractive research field of machine learning: power systems and financial markets related time series forecasting. Various ensemble methods have been investigated and developed, which includes decision tree ensembles, empirical mode decomposition based ensemble methods and wavelet decomposition based ensemble methods. Four kinds of time series forecasting problems are covered: electric load forecasting, wind power ramp forecasting, electricity price forecasting and stock market price forecasting. In other words, this thesis has focused on the methodologies to improve the forecasting accuracy under the help of ensemble methods.

The topic of the first part of this thesis is ensemble methods for power system related time series forecasting. In Chapter 3, a decision tree ensemble method is presented, named as oblique random forest with least square estimation. The proposed oblique RF has better performance compared with the original RF in both generic TS assessment and short term electricity load demand forecasting. The model complexity is also discussed, which is given as $O(Nn^3)$. In Chapter 4, we present two ensemble deep learning methods for short term electric load forecasting: ensemble deep belief network (EDBN) and EMD based ensemble deep belief network (EMD-DBN). The simulation results demonstrate the advantages of the ensemble deep learning models successfully. Moreover, in Chapter 5, we investigate the ensemble method for incremental learning, which makes use of a non-iterative model with high efficiency: Random Vector Functional Link (RVFL) network. The proposed DWT-EMD based ensemble approach outperforms EMD based and single structure models. On the other hand, in Chapter 6, ensemble learning methods are employed to deal with an important challenge on wind power utilization: wind power ramp forecasting. Three kinds of experiments are conducted: wind power forecasting, wind power ramp forecasting and wind power ramp classification. The computation

speed is also compared among the benchmark models. The experimental results demonstrate the effectiveness of the proposed methods.

In the second part of this thesis, ensemble learning for financial markets related time series forecasting is investigated and discussed. Especially, two kinds of financial time series forecasting problems are considered: electricity price and stock price forecasting. In Chapter 8, an ensemble kernel machine is proposed for short-term electricity price forecasting, which is composed of EMD, KRR and SVR. In Chapter 9, sharing the similar idea of the incremental ensemble model in Chapter 5, a fusion incremental learning approach is presented for short-term stock price forecasting, which is composed of Discrete Wavelet Transform (DWT), Empirical Mode Decomposition (EMD), Random Vector Functional Link (RVFL) network and Support Vector Regression (SVR). Besides the historical data of stock market closing price, ten indicators are also included to improve the performance of the fusion model. The simulation results demonstrate the advantages of ensemble learning methods for financial market related time series forecasting.

Moreover, in Chapter 7 and Chapter 10, two overall comparison experiments are conducted for electric load forecasting and stock price forecasting. All the ensemble methods mentioned in this thesis are compared and ranked based on Friedman test and Nemenyi post-hoc testing. The simulation results demonstrate the advantages of the ensemble incremental learning models for both power systems and financial markets related time series forecasting. The combination of ensemble learning and deep learning also achieves good performance, yet paying the price of the model complexity. The non-iterative model RVFL can boost its learning ability under the help of ensemble learning and incremental learning strategy; while still keeps the overall computation time reasonable. It is also worth noting that decision tree based ensemble methods cannot beat the neural network based ensemble models for time series forecasting problems investigated in this thesis.

Last but not least, in Chapter 11, the input-output relations within process flow sheet of a biodiesel plant are analyzed using surrogate models constructed by various machine learning methods. A variety of scenarios are considered: $1, 2, 6$ and 11 input variables are changed simultaneously, 3 domain sizes of input variables are considered and 5 different surrogates (support vector regression (SVR), artificial neural network (ANN), deep belief network (DBN), random forests (RF), and random vector functional link network (RVFL)) are used. Heat duties of equipment within the plant are used for considered outputs. The simulation results show the attractiveness of machine learning methods for constructing surrogate models based on three error measures: root-mean-squared-deviation (RMSD), R^2 and residuals. Moreover, a comparison among polynomial response surfaces fitting, high dimensional model representation (HDMR) and machine learning models is conducted based on statistical testing. The efficiency of learning

methods is also compared. The comparison results show that SVR achieves the best performance based on error measures, followed by DBN and ANN; while RVFL is the most efficient model.

Generally speaking, deep learning models combined with ensemble strategies have their advantages dealing with complicated real world time series forecasting tasks, especially for financial market related data with numerous external influence factors. On the other hand, when the problem is not very complicated, or there is strict limitation on computation complexity, non-iterative models with incremental learning are good choice for time series forecasting. A more detailed list of important conclusions and findings is shown below:

1. Ensemble learning methods are proven to be effective for various power system related time series forecasting tasks, including electric load forecasting, wind power forecasting, etc...
2. Based on no free lunch (NFL) theorem, there exists no learning algorithm that can be universally good. What we can do is to strategically optimize our ensemble model focusing on specific type of application tasks.
3. Empirical Mode Decomposition and its improved variants EEMD and CEEMDAN can benefit time series data analysis and feature extraction procedures significantly.
4. Deep learning models can also benefit from ensemble methods for time series forecasting. LSTM, as the most successful type of deep recurrent neural network to deal with long term memory, is one of the best choice for time series forecasting, especially when the accuracy is considered to be more important than efficiency for performance evaluation.
5. Random Forests, and decision tree ensembles, work for short term time series forecasting tasks quite well, and can outperform many non-ensemble benchmark models, including SVR and ANN. Oblique Random Forests further improve the performance by employing various data splitting methods of decision trees.
6. Non-iterative models, such as RVFL, is suitable for medium size datasets with high computation speed, and can be used as good base learners in ensemble models. For example, after decomposed by EMD, time series signal is transformed to a group of sub-signals, which are easier to be analyzed and modeled by fast models.
7. Incremental learning is beneficial for short term time series forecasting regarding to both accuracy and efficiency. The model can be updated by a relatively fast learning mechanism based on new incoming stream data to maintain good performance.
8. Time series data from financial markets generally possesses more complicated pattern compared with power system related data, because there exists strong human factors which bring in high level noise and randomness that make financial TS data hard to predict.

12.2 Future Work

In reality, time series signals are often influenced by numerous factors, such as weather conditions and economic fluctuations for electricity load demand time series. Therefore, for future work, additional nonlinear features need to be considered in constructing a more complex model. That is to say, multivariate time series forecasting models shall be constructed instead of univariate models. The potential learning ability of deep learning methods shall be well suited to such complex problems. Therefore, more deep learning methods and ensemble algorithms shall be developed for time series forecasting.

For oblique random forests, in addition to the least square method in Chapter 3 and MPSVM in benchmark, more hyperplane selection methods, including impurity score based methods, can be developed and tested in the context of time series forecasting. Furthermore, based on the fact that both deep learning (DBN in this study) and ensemble method (RF based methods) achieve better performance compared with basic learning models, hybrid ensemble deep learning models can be developed by combining the concept of deep learning and random forests.

Moreover, since the ensemble deep learning model is more time consuming when compared with the single structure model, optimization techniques can be designed to simplify the structure and increase the efficiency of deep learning models. When computing time is taken into account for the evaluation of performance, the non BP-based algorithms shall be considered, such as KRR and RVFL network. Ensemble models can be designed based on KRR or RVFL which can obtain comparable prediction results while having the advantage of fast training. Moreover, RF for regression shall also be tested for time series forecasting. For wind power forecasting, the ramp classification problem can be extended from binary class to multiple class. Some examples of additional classes are up-ramp and down-ramp events, and strong ramp and weak ramp events. For stock price forecasting, we can design models focused on predicting the direction of movement of the stock market index, which is much more meaningful for the development of effective market trading strategies.

Generally speaking, the EMD and DWT based decomposition approach can be combined with various learning algorithms; while these outcome ensemble models can be tested on various applications, including renewable energy and financial data. Moreover, probabilistic load forecasting methods, which provides time series forecasting output in the form of intervals, scenarios, density functions or probabilities, can also been developed by combining the proposed accurate point forecasting models with good classification methods.

For surrogate models, more machine learning algorithms should be investigated for constructing surrogates of more complex chemical models. For example, deep recurrent neural networks can be applied for constructing surrogate models to simulate a hybrid chemical model with feedback loops generated by a number of interconnected models. Moreover, electrical models

and the interactions with chemical models within EIPs can also be simulated and analyzed by surrogates.

List of Publications

Journal Articles:

- **X. Qiu**, Y. Ren, P. N. Suganthan, and G. A. J. Amaratunga, "Empirical mode decomposition based ensemble deep learning for load demand time series forecasting," in *Applied Soft Computing*, vol. 54, pp. 246–255, 2017.
- **X. Qiu**, L. Zhang, P. N. Suganthan, and G. A. J. Amaratunga, "Oblique random forest ensemble via least square estimation for time series forecasting," in *Information Sciences*, vol.420, pp.249–262, 2017.
- **X. Qiu**, P. N. Suganthan, and G. A. J. Amaratunga, "Ensemble Incremental Learning Random Vector Functional Link Network for Short-term Electricity Load Demand Forecasting," *Knowledge-based Systems,* (Accepted).
- **X. Qiu**, P. N. Suganthan, and G. A. J. Amaratunga, "Fusion of Multiple Indicators with Ensemble Incremental Learning Techniques for Stock Price Forecasting," *Information Fusion,* (Submitted).
- **X. Qiu**, J. J. Sikorski, S. S. Garud, P. N. Suganthan, and M. Kraft, "Machine learning approach for constructing surrogates of a biodiesel plant flow sheet model," *Computers & Chemical Engineering,* (Submitted).

Conference Papers:

- **X. Qiu**, P. N. Suganthan, and G. A. J. Amaratunga, "Short-term Wind Power Ramp Forecasting with Empirical Mode Decomposition based Ensemble Learning Techniques," in: *Proc. IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL'17)*, Hawaii, US, Dec. 2017.
- **X. Qiu**, P. N. Suganthan, and G. A. J. Amaratunga, "Short-term Electricity Price Forecasting with Empirical Mode Decomposition based Ensemble Kernel Machines," *Procedia Computer Science*, vol. 208, pp. 1308–1317, 2017.

- **X. Qiu**, H. Zhu, P. N. Suganthan, and G. A. J. Amaratunga, "Stock Price Forecasting with Empirical Mode Decomposition Based Ensemble ν-Support Vector Regression Model," in: *Proc. International Conference on Computational Intelligence, Communications, and Business Analytics*, Sep. 2017.

- **X. Qiu**, P. N. Suganthan, and G. A. J. Amaratunga, "Electricity Load Demand Time Series Forecasting with Empirical Mode Decomposition based Random Vector Functional Link Network," in: *Proc. IEEE Conference on Systems, Man and Cybernetics (SMC2016)*, Budapest, Hungary, Oct. 2016.

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

- 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.

- Y. Ren, **X. Qiu**, P. N. Suganthan, N. Srikanth, and G. Amaratunga, "Detecting wind power ramp with random vector functional link (RVFL) network," in: *Proc. IEEE Symposium Series on Computational Intelligence (CIEL'15)*, Cape Town, South Africa, Dec. 2015.

- A. C. Palaninathan, **X. Qiu**, and P. N. Suganthan, "Heterogeneous ensemble for power load demand forecasting," in: *Proc. TENCON 2016 - 2016 IEEE Region 10 Conference*, Singapore, Nov. 2016.

Bibliography

[1] M. Pan, J. Sikorski, C. A. Kastner, J. Akroyd, S. Mosbach, R. Lau, M. Kraft, Applying industry 4.0 to the jurong island eco-industrial park, Energy Procedia 75 (2015) 1536–1541.

[2] J. J. Sikorski, G. Brownbridge, S. S. Garud, S. Mosbach, I. A. Karimi, M. Kraft, Parameterisation of a biodiesel plant process flow sheet model, Computers & Chemical Engineering 95 (2016) 108–122.

[3] W.-C. Hong, Electric load forecasting by seasonal recurrent svr (support vector regression) with chaotic artificial bee colony algorithm, Energy 36 (2011) 5568–5578.

[4] J. Wang, W. Zhu, W. Zhang, D. Sun, A trend fixed on firstly and seasonal adjustment model combined with the ε-SVR for short-term forecasting of electricity demand, Energy Policy 37 (2009) 4901–4909.

[5] J. Che, J. Wang, G. Wang, An adaptive fuzzy combination model based on self-organizing map and support vector regression for electric load forecasting, Energy 37 (2012) 657–664.

[6] I. Koprinska, M. Rana, A. Troncoso, F. Martínez-Álvarez, Combining pattern sequence similarity with neural networks for forecasting electricity demand time series, in: International Joint Conference on Neural Networks (IJCNN), 2013.

[7] S. Fan, R. Hyndman, Short-term load forecasting based on a semi-parametric additive model, IEEE Trans. Power Syst. 27 (1) (2012) 134–141.

[8] Y. Kara, M. A. Boyacioglu, Ömer Kaan Baykan, Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange, Expert Systems with Applications 38 (2011) 5311–5319.

[9] W.-C. Hong, Y. Dong, L.-Y. Chen, S.-Y. Wei, Seasonal support vector regression with chaotic genetic algorithm in electric load forecasting, in: Proceedings of the 2012 Sixth International Conference on Genetic and Evolutionary Computing, IEEE Computer Society, 2012, pp. 124–127.

[10] J. C. Sousa, H. M. Jorge, L. P. Neves, Short-term load forecasting based on support vector regression and load profiling, International Journal of Energy Research 38 (3) (2014) 350–362.

[11] T. Ouyang, X. Zha, L. Qin, A survey of wind power ramp forecasting, Energy and Power Engineering 5 (2013) 368–372.

[12] H. Zareipour, D. Huang, 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.

[13] M. Cui, D. Ke, Y. Sun, D. Gan, J. Zhang, B.-M. Hodge, Wind power ramp event forecasting using a stochastic scenario generation method, IEEE Trans.Sustainable Energy 6 (2) (2015) 422–433.

[14] N. Francis, Predicting sudden changes in wind power generation, North American Windpower 5 (2008) 58–60.

[15] R. Girard, A. Bossavy, G. Kariniotakis, Forecasting ramps of wind power production at different time scales, in: European Wind Energy Conference, 2011.

[16] X. Qiu, Y. Ren, P. N. Suganthan, G. A. J. Amaratumga, Empirical mode decomposition based ensemble deep learning for load demand time series forecasting, Applied Soft Computing 54 (2017) 246–255.

[17] S. Barak, A. Arjmand, S. Ortobelli, Fusion of multiple diverse predictors in stock market, Information Fusion 36 (2017) 90–102.

[18] K. He, R. Zha, J. Wu, K. K. Lai, Multivariate EMD-based modeling and forecasting of crude oil price, Sustainability 8 (4) (2016) 387.

[19] I. Koprinska, M. Rana, V. G. Agelidis, Correlation and instance based feature selection for electricity load forecasting, Knowledge-Based Systems 82 (2015) 29–40.

[20] H. Raza, G. Prasad, Y. Li, EWMA model based shift-detection methods for detecting covariate shifts in non-stationary environments, Pattern Recognition 48 (3) (2015) 659–669.

[21] A. Gharehbaghi, P. Ask, A. Babic, A pattern recognition framework for detecting dynamic changes on cyclic time series, Pattern Recognition 48 (3) (2015) 696–708.

[22] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, Time Series Analysis: Forecasting and Control, Wiley Series in Probability and Statistics, Wiley, 2008.

[23] J. Bargur, A. Mandel, M. ha-Tekhniyon le-mehkar u-fituah (Haifa Israel), I. M. ha-energyah veha tashtit, Energy Consumption and Economic Growth in Israel: Trend Analysis (1960-1979), Ministry of Energy and Infrastructure, 1981.

[24] C. C. Holt, Forecasting seasonals and trends by exponentially weighted moving averages, International Journal of Forecasting 20 (2004) 5–10.

[25] G. E. P. Box, G. M. Jenkins, Time series analysis: forecasting and control, Holden-Day series in time series analysis and digital processing, Holden-Day, 1976.

[26] A. D. Papalexopoulos, T. C. Hesterberg, A regression-based approach to short-term system load forecasting, IEEE Transactions on Power Systems 5 (1990) 1535–1547.

[27] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297.

[28] L. Suganthi, A. A. Samuel, Energy models for demand forecastinga review, Renewable and Sustainable Energy Reviews 16 (2) (2012) 1223–1240.

[29] J. Che, J. Wang, Short-term electricity prices forecasting based on support vector regression and auto-regressive integrated moving average modeling, Energy Conversion and Management 51 (2010) 1911–1917.

[30] T. Hofmann, B. Schlkopf, A. J. Smola, Kernel methods in machine learning, Annals of Statistics 36 (3) (2008) 1171–1220.

[31] L. Zhang, P. N. Suganthan, Robust visual tracking via co-trained kernelized correlation filters, Pattern Recognition 69 (2017) 82–93.

[32] G. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for Deep Belief Nets, Neural Computation 18 (7) (2006) 1527–1554.

[33] D. C. Ciresan, U. Meier, L. M. Gambardella, J. Schmidhuber, Deep, big, simple neural nets for handwritten digit recognition, Neural Computation 22 (12) (2010) 3207–3220.

[34] X. Qiu, L. Zhang, Y. Ren, P. N. Suganthan, G. Amaratunga, Ensemble deep learning for regression and time series forecasting, in: Proc. IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL2014), Orlando, US, 2014.

[35] E. Busseti, I. Osband, S. Wong, Deep learning for time series modeling, Tech. rep., Technical report, Stanford University (2012).

[36] A. Abdullah, R. C. Veltkamp, M. A. Wiering, An ensemble of deep support vector machines for image categorization, in: Proc. IEEE International Conference of Soft Computing and Pattern Recognition (SOCPAR'09), 2009, pp. 301–306.

[37] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, The Journal of Machine Learning Research 11 (2010) 3371–3408.

[38] F. Agostinelli, M. R. Anderson, H. Lee, Adaptive multi-column deep neural networks with application to robust image denoising, in: Proc. Advances in Neural Information Processing Systems (NIPS'13), 2013, pp. 1493–1501.

[39] D. Cireşan, U. Meier, J. Schmidhuber, Multi-column deep neural networks for image classification, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12), 2012, pp. 3642–3649.

[40] D. Cireşan, U. Meier, J. Masci, J. Schmidhuber, Multi-column deep neural network for traffic sign classification, Neural Networks 32 (2012) 333–338.

[41] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.

[42] T. Kuremoto, S. Kimura, K. Kobayashi, M. Obayashi, Time series forecasting using restricted boltzmann machine, in: Emerging Intelligent Computing Technology and Applications, Springer, 2012, pp. 17–22.

[43] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32.

[44] M. Fernández-Delgado, E. Cernadas, S. Barro, D. Amorim, Do we need hundreds of classifiers to solve real world classification problems?, The Journal of Machine Learning Research 15 (1) (2014) 3133–3181.

[45] L. Breiman, J. H. Freidman, R. A. Olshen, C. J. Stone, Classification and regression trees, The Wadsworth and Brooks-Cole statistics-probability series, Taylor & Francis, 1984.

[46] H. Jiang, Y. Deng, H.-S. Chen, L. Tao, Q. Sha, J. Chen, C.-J. Tsai, S. Zhang, Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes, BMC Bioinformatics 5 (1) (2004) 81.

[47] T. K. Ho, The random subspace method for constructing decision forests, Pattern Analysis and Machine Intelligence 20 (8) (1998) 832–844.

[48] B. H. Menze, B. M. Kelm, R. Masuch, U. Himmelreich, P. Bachert, W. Petrich, F. A. Hamprecht, A comparison of random forest and its gini importance with standard chemometric methods for the feature selection and classification of spectral data, BMC Bioinformatics 10 (1) (2009) 213.

[49] L. Zhang, P. N. Suganthan, Oblique decision tree ensemble via multisurface proximal support vector machine, IEEE Transactions on Cybernetics 45 (10) (2015) 2165–2176.

[50] B. Li, G. Yang, R. Wan, X. Dai, Y. Zhang, Comparison of random forests and other statistical methods for the prediction of lake water level: a case study of the Poyang Lake in China, Hydrology Research 47 (2016) 69–83.

[51] T. J. Hall, C. N. Mutchler, G. J. Bloy, R. N. Thessin, S. K. Gaffney, J. J. Lareau, Performance of observation-based prediction algorithms for very short-range, probabilistic clear-sky condition forecasting, Journal of Applied Meteorology and Climatology 50 (2011) 3–19.

[52] Y.-H. Pao, S. M. Phillips, D. J. Sobajic, Neural-net computing and the intelligent control of systems, International Journal of Control 56 (1992) 263–289.

[53] Y.-H. Pao, G.-H. Park, D. J. Sobajic, Learning and generalization characteristics of the random vector functional-link net, Neurocomputing 6 (1994) 163–180.

[54] Y. Ren, P. N. Suganthan, N. Srikanth, G. Amaratunga, Random vector functional link network for short-term electricity load demand forecasting, Information Sciences 367-368 (2016) 1078–1093.

[55] W. F. Schmidt, M. A. Kraaijveld, R. P. W. Duin, Feedforward neural networks with random weights, in: Proceedings of the IAPR International Conference on Pattern Recognition Conference B: Pattern Recognition Methodology and Systems, 1992, pp. 1–4.

[56] L. Zhang, P. N. Suganthan, A comprehensive evaluation of random vector functional link networks, Information Sciences 367-368 (2016) 1094–1105.

[57] L. Breiman, Bias, variance, and arcing classifiers, Tech. rep. (1996).

[58] S. Chatterjee, A. Dash, S. Bandopadhyay, Ensemble support vector machine algorithm for reliability estimation of a mining machine, Quality and Reliability Engineering International 31 (8) (2015) 1503–1516.

[59] T. G. Dietterich, Ensemble methods in machine learning, in: Multiple classifier systems, Springer, 2000, pp. 1–15.

[60] J. M. Moreira, C. Soares, A. M. Jorge, J. F. de Sousa, Ensemble approaches for regression: A survey, ACM Computing Surveys 45 (1) (2012) 1–10.

[61] Y. Ren, P. N. Suganthan, N. Srikanth, Ensemble methods for wind and solar power forecasting: a state-of-the-art review, Renewable Sustain. Energy Rev. 50 (2015) 82–91.

[62] Y. Ren, X. Qiu, 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, 2014.

[63] Y. Ren, P. N. Suganthan, N. Srikanth, A novel empirical mode decomposition with support vector regression for wind speed forecasting, IEEE Transactions on Neural Network and Learning Systems 27 (8) (2016) 1793–1798.

[64] J. Shi, J. Guo, S. Zheng, Evaluation of hybrid forecasting approaches for wind speed and power generation time series, Renewable and Sustainable Energy Reviews 16 (2012) 3471–3480.

[65] J. Wu, C. C. Keong, Prediction of hourly solar radiation using a novel hybrid model of ARMA and TDNN, Solar Energy 85 (5) (2011) 808–817.

[66] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, MIT Press, 2000.

[67] D. Benaouda, F. Murtagh, J.-L. Starck, O. Renaud, Wavelet-based nonlinear multiscale decomposition model for electricity load forecasting, Neurocomputing 70 (2006) 139–154.

[68] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, H. H. Liu, The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis, in: Roy. Soc. London A, Vol. 454, 1998, pp. 903–995.

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

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

[71] A. U. Haque, P. Mandal, J. Meng, A. K. Srivastava, T.-L. Tseng, T. Senjyu, A novel hybrid approach based on wavelet transform and fuzzy ARTMAP networks for predicting wind farm power production, IEEE Transactions on Industry Applications 49 (2013) 2253–2261.

[72] J. Wu, C. C. Keong, Prediction of hourly solar radiation using a novel hybrid model of ARMA and TDNN, Solar Energy 85 (5) (2011) 808–817.

[73] X. Geng, K. Smith-Miles, Incremental Learning, Springer US, Boston, MA, 2009, pp. 731–735.

[74] J. L. Elman, Learning and development in neural networks: the importance of starting small, Cognition 48 (1993) 71–99.

[75] G. Grmanová, P. Laurinec, V. Rozinajová, A. B. Ezzeddine, M. Lucká, P. Lacko, P. Vrablecová, P. Návrat, Incremental ensemble learning for electricity load forecasting, Acta Polytechnica Hungarica 13 (2016) 97–117.

[76] Y. Yang, J. Che, Y. Li, Y. Zhao, S. Zhu, An incremental electric load forecasting model based on support vector regression, Energy 113 (2016) 796–808.

[77] X. Qiu, L. Zhang, P. N. Suganthan, G. A. J. Amaratunga, Oblique random forest ensemble via least square estimation for time series forecasting, Information Sciences 420 (2017) 249–262.

[78] X. Qiu, P. N. Suganthan, G. A. J. Amaratunga, Ensemble incremental learning random vector functional link network for short-term electric load forecasting, Knowledge-Based Systems 145 (2018) 182–196.

[79] X. Qiu, Y. Ren, P. N. Suganthan, G. A. J. Amaratunga, Short-term wind power ramp forecasting with empirical mode decomposition based ensemble learning techniques, in: Proc. IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, Hawaii, USA, 2017.

[80] X. Qiu, P. N. Suganthan, G. A. J. Amaratunga, Short-term electricity price forecasting with empirical mode decomposition based ensemble kernel machines, Procedia Computer Science 108 (2017) 1308–1317.

[81] S. B. Kotsiantis, D. Kanellopoulos, P. E. Pintelas, Data preprocessing for supervised learning, International Journal of Computer Science 1 (2) (2006) 111–117.

[82] K. Lakshminarayan, S. A. Harp, T. Samad, Imputation of missing data in industrial databases, Applied Intelligence 11 (3) (1999) 259–275.

[83] Z. Wu, 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, The Royal Society, 2004, pp. 1597–1611.

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

[85] Y. Ye, Q. Wu, J. Z. Huang, M. K. Ng, X. Li, Stratified sampling for feature subspace selection in random forests for high dimensional data, Pattern Recognition 46 (3) (2013) 769–787.

[86] I. Borg, P. J. F. Groenen, Modern Multidimensional Scaling: Theory and Applications, Springer series in statistics, Springer, 1997.

[87] P. Whittle, Hypothesis Testing in Time Series Analysis, Almquist and Wicksell, 1951.

[88] P. Whittle, Prediction and Regulation, English Universities Press, 1963.

[89] P. Whittle, Prediction and Regulation by Linear Least-Square Methods, University of Minnesota Press, 1983.

[90] E. Hannan, M. Deistler, The statistical theory of linear systems, Wiley series in probability and mathematical statistics: Probability and mathematical statistics, Wiley, 1988.

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

[92] L. Zhang, P. N. Suganthan, A survey of randomized algorithms for training neural networks, Information Sciences 364–365 (2016) 146–155.

[93] Y. Ren, X. Qiu, P. N. Suganthan, N. Srikanth, G. Amaratunga, Detecting wind power ramp with random vector functional link (rvfl) network, in: Proc. IEEE Symposium Series on Computational Intelligence (CIEL2015), Cape Town, South Africa, 2015.

[94] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, V. Vapnik, Support vector regression machines, Advances in Neural Information Processing Systems 9 (1997) 155–161.

[95] B. Schölkopf, A. J. Smola, R. C. Williamson, P. L. Bartlett, New support vector algorithms, Neural Computation 12 (2000) 1207–1245.

[96] C.-C. Chang, C.-J. Lin, Training ν-support vector regression: Theory and algorithms, Neural Computation 14 (2002) 1959–1977.

[97] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[98] L. Breiman, Bagging predictors, Machine Learning 24 (2) (1996) 123–140.

[99] L. Zhang, P. N. Suganthan, Random forests with ensemble of feature spaces, Pattern Recognition 47 (2014) 3429–3437.

[100] N. Manwani, P. Sastry, Geometric decision tree, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 42 (1) (2012) 181–192.

[101] W. Pedrycz, Z. A. Sosnowski, Genetically optimized fuzzy decision trees, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 35 (3) (2005) 633–641.

[102] S.-H. Cha, C. Tappert, A genetic algorithm for constructing compact binary decision trees, Journal of Pattern Recognition Research 4 (1) (2009) 1–13.

[103] L. Rokach, O. Maimon, Top-down induction of decision trees classifiers-a survey, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 35 (4) (2005) 476–487.

[104] G. Nicolas, T. P. Robinson, G. R. W. Wint, G. Conchedda, G. Cinardi, M. Gilbert, Using random forest to improve the downscaling of global livestock census data, PLOS ONE 11 (2016) 1–16.

[105] X. Jiang, M. Abdel-Aty, J. Hu, J. Lee, Investigating macro-level hotzone identification and variable importance using big data: A random forest models approach, Neurocomputing 181 (2016) 53–63.

[106] Y. Bengio, Learning deep architectures for ai, Foundations and Trends in Machine Learning 2 (2009) 1–127.

[107] H. A. Song, S.-Y. Lee, Hierarchical representation using nmf, Neural Information Processing 8226 (2013) 466–473.

[108] G. E. Hinton, R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.

[109] T. Yamashita, M. Tanaka, E. Yoshida, Y. Yamauchi, H. Fujiyoshi, To be Bernoulli or to be Gaussian, for a restricted Boltzmann machine, in: 22nd International Conference on Pattern Recognition, 2014, pp. 1520–1525.

[110] G. E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Computation 14 (8) (2002) 1771–1800.

[111] G. Hinton, A practical guide to training restricted boltzmann machines, Momentum 9 (1) (2010) 926.

[112] L. Zhang, Ensemble classification and theirapplications to visual tracking, Ph.D. thesis, Nanyang Technological University (2016).

[113] D. Opitz, R. Maclin, Popular ensemble methods: An empirical study, Journal of Artificial Intelligence Research 11 (1999) 169–198.

[114] S. Geman, E. Bienenstock, R. Doursat, Neural networks and the bias/variance dilemma, Neural Computation 4 (1992) 1–58.

[115] A. C. Palaninathan, X. Qiu, P. N. Suganthan, Heterogeneous ensemble for power load demand forecasting, in: Proc. TENCON 2016 - 2016 IEEE Region 10 Conference, Singapore, Singapore, 2016.

[116] 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.

[117] P. Büchlmann, B. Yu, Analyzing bagging, Annals of Statistics (2002) 927–961.

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

[119] T. G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization, Machine Learning 40 (2) (2000) 139–157.

[120] T. Chen, J. Ren, Bagging for gaussian process regression, Neurocomputing 72 (7) (2009) 1605–1610.

[121] S. Basterrech, V. Snášel, Time-series forecasting using bagging techniques and reservoir computing, in: Proc. International Conference of Soft Computing and Pattern Recognition (SoCPaR2013), Hanoi, Vietnam, 2013.

[122] Y. Ganjisaffar, R. Caruana, C. V. Lopes, Bagging gradient-boosted trees for high precision, low variance ranking models, in: Proceedings of the 34th international ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2011, pp. 85–94.

[123] D. Shrestha, D. Solomatine, Experiments with adaboost.rt, an improved boosting scheme for regression, Neural Computation (7) (2006) 1678–1710.

[124] A. Vezhnevets, V. Vezhnevets, Modest adaboost-teaching adaboost to generalize better, in: Graphicon, Vol. 12, 2005, pp. 987–997.

[125] S. Avidan, SpatialBoost: Adding spatial reasoning to adaboost, in: Proc. Computer Vision (ECCV'06), 2006, pp. 386–396.

[126] P. Büchlmann, B. Yu, Stochastic gradient boosting, Computational Statistics Data Analysis (2002) 927–961.

[127] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors), The Annals of Statistics (2) (2000) 337–407.

[128] S. D. P. Kankanala, A. Pahwa, Adaboost+ : An ensemble learning approach for estimating weather-related outages in distribution systems, IEEE Trans. Power Syst. (1) (2014) 359–367.

[129] R. Feely, Predicting stock market volatility using neural networks, Master's thesis, Trinity College Dublin (2000).

[130] R. E. Schapire, Y. Singer, Improved boosting algorithms using confidence-rated predictions, Machine Learning 37 (3) (1999) 297–336.

[131] B. Z. J. Wu, K. Wang, Application of adaboost-based bp neural network for short-term wind speed forecast, Power System Technology 36 (9) (2012) 221225.

[132] D. H. Wolpert, Stacked generalization, Neural Networks 5 (2) (1992) 241–259.

[133] L. Bo, L. Xinjun, Z. Zhiyan, Novel algorithm for constructing support vector machine regression ensemble, Journal of Systems Engineering and Electronics 17 (3) (2006) 541–545.

[134] K. Lu, L. Wang, A novel nonlinear combination model based on support vector machine for rainfall prediction, in: Proc. IEEE International Joint Conference on Computational Sciences and Optimization (CSO'11), 2011, pp. 1343–1346.

[135] L. Wang, J. Wu, Application of hybrid RBF neural network ensemble model based on wavelet support vector machine regression in rainfall time series forecasting, in: Proc. IEEE International Joint Conference on Computational Sciences and Optimization (CSO'12), 2012, pp. 867–871.

[136] J.-S. Chou, A.-D. Pham, Enhanced artificial intelligence for ensemble approach to predicting high performance concrete compressive strength, Construction and Building Materials 49 (2013) 554–563.

[137] J. Wu, A novel artificial neural network ensemble model based on k-nearest neighbor nonparametric estimation of regression function and its application for rainfall forecasting, in: Proc. International Joint Conference on Computational Sciences and Optimization (CSO'09), Vol. 2, 2009, pp. 44–48.

[138] A. Grossmann, J. Morlet, Decomposition of hardy functions into square integrable wavelets of constant shape, SIAM Journal on Mathematical Analysis 15 (1984) 723–736.

[139] D. C. Kiplangat, K. Asokan, K. S. Kumar, Improved week-ahead predictions of wind speed using simple linear models with wavelet decomposition, Renewable Energy 93 (2016) 38–44.

[140] D. Percival, A. Walden, Wavelet Methods for Time Series Analysis, Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, 2006.

[141] D. B. Percival, M. Wang, J. E. Overland, An introduction to wavelet analysis with applications to vegetation time series, Community Ecology 5 (1) (2004) 19–30.

[142] H. Liu, H.-Q. Tian, C. Chen, Y.-F. Li, A hybrid statistical method to predict wind speed and wind power, Renewable Energy 35 (2010) 1857–1861.

[143] J. P. S. Catalao, H. M. I. Pousinho, V. M. F. Mendes, Hybrid wavelet-PSO-ANFIS approach for short-term electricity prices forecasting, IEEE Transactions on Power Systems 26 (2011) 137–144.

[144] N. M. Pindoriya, S. N. Singh, An adaptive wavelet neural network-based energy price forecasting in electricity markets, IEEE Transactions on Power Systems 23 (2008) 1423–1432.

[145] J.C.Cao, S.H.Cao, Study of forecasting solar irradiance using neural networks with preprocessing sample data by wavelet analysis, Energy 31 (2006) 3435–3445.

[146] M. E. Torres, M. A. Colominas, G. Schlotthauer, P. Flandrin, A complete ensemble empirical mode decomposition with adaptive noise, in: Proc. 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, Prague, 2011, pp. 4144–4147.

[147] Z. Hu, Y. Bao, T. Xiong, A hybrid anfis model based on empirical mode decomposition for stock time series forecasting, Applied Soft Computing 42 (2016) 368–376.

[148] L. Ghelardoni, A. Ghio, D. Anguita, Energy load forecasting using empirical mode decomposition and support vector regression, IEEE Transactions on Smart Grid 4 (1) (2013) 549–556.

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

[150] H. K. Alfares, M. Nazeeruddin, Electric load forecasting: literature survey and classification of methods, International Journal of Systems Science 33 (2002) 23–34.

[151] T. Hong, Crystal ball lessons in predictive analytics, EnergyBiz (2015) 35–37.

[152] S. K. Murthy, S. Kasif, S. Salzberg, R. Beigel, Oc1: A randomized algorithm for building oblique decision trees, in: Proceedings of AAAI, Vol. 93, Citeseer, 1993, pp. 322–327.

[153] L. Zhang, Y. Ren, P. Suganthan, Towards generating random forests via extremely randomized trees, in: Neural Networks (IJCNN), 2014 International Joint Conference on, IEEE, 2014, pp. 2645–2652.

[154] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Machine Learning 63 (1) (2006) 3–42.

[155] C.-C. Chang, C.-J. Lin, Libsvm: a library for support vector machines, ACM Transactions on Intelligent Systems and Technology (TIST) 2 (3) (2011) 27.

[156] R. B. Palm, Prediction as a candidate for learning deep hierarchical models of data, Master's thesis (2012).

[157] P. A. Morettin, C. Toloi, Análise de séries temporais, Blucher, 2006.

[158] S. Haykin, Neural Networks: A Comprehensive Foundation, International edition, Prentice Hall, 1999.

[159] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.

[160] Australian energy market operator (Dec. 2013).
URL http://www.aemo.com.au/

[161] J. Xie, B. Xu, Z. Chuang, Horizontal and vertical ensemble with deep representation for classification, arXiv preprint arXiv:1306.2759.

[162] L. Breiman, J. H. Freidman, R. A. Olshen, C. J. Stone, Classification and regression trees, Wadsworth, 1984.

[163] L. Torgo, Regression datasets (2014).
URL http://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html

[164] J. FRIEDMAN, Multivariate adaptative regression splines, in: Annals of Statistics, Vol. 19, 1991, pp. 1–141.

[165] L. Breiman, Bagging predictors, in: Machine Learning, Vol. 24, Kluwer Academic Publishers, 1996, pp. 123–140.

[166] R. K. Pace, R. Barry, Sparse spatial autoregressions, in: Statistics and Probability Letters, Vol. 33, StatLib repository, 1997, pp. 291–297.

[167] E. Romero, D. Toppo, Comparing support vector machines and feedforward neural networks with similar hidden-layer weights, IEEE Transactions on Neural Networks 18 (3) (2007) 959–963.

[168] L. Ye, P. Liu, Combined model based on emd-svm for short-term wind power prediction, in: Proc. Chinese Society for Electrical Engineering (CSEE), Vol. 31, 2011, pp. 102–108.

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

[170] C. Chen, J. Wan, A rapid learning and dynamic stepwise updating algorithm for flat neural networks and the application to time-series prediction, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 29 (1) (1999) 62–72.

[171] Y. Chen, M. Q. Feng, A technique to improve the empirical mode decomposition in the hilbert-huang transform, Earthquake Engineering and Engineering Vibration 2 (2003) 796–808.

[172] Z. Wu, N. E. Huang, Ensemble empirical mode decomposition: A noise-assisted data analysis method, Advances in Adaptive Data Analysis 1 (2009) 1–41.

[173] Australian energy market operator (Dec. 2016).
URL http://www.aemo.com.au/

[174] Australian bureau of meteorology (Jun. 2017).
URL http://www.bom.gov.au/

[175] X. Qiu, Y. Ren, P. N. Suganthan, G. A. J. Amaratunga, Empirical mode decomposition based ensemble deep learning for load demand time series forecasting, Applied Soft Computing 54 (2017) 246–255.

[176] S. Haben, J. Ward, D. V. Greetham, C. Singleton, P. Grindrod, A new error measure for forecasts of household-level, high resolution electrical energy consumption, International Journal of Forecasting 30 (2014) 246–256.

[177] T. Hong, P. Pinson, S. Fan, Global energy forecasting competition 2012, International Journal of Forecasting 30 (2) (2014) 357–363.

[178] Z. Fan, Y. Zuo, D. Jiang, X. Cai, Prediction of acute hypotensive episodes using random forest based on genetic programming, in: Proc. IEEE Conference on Evolutionary Computation (CEC2015), 2015.

[179] X. Qiu, P. N. Suganthan, G. Amaratunga, Electricity load demand time series forecasting with empirical mode decomposition based random vector functional link network, in: Proc. IEEE Conference on Systems, Man and Cybernetics (SMC2016), Budapest, Hungary, 2016.

[180] R.-A. Hooshmand, H. Amooshahi, M. Parastegari, A hybrid intelligent algorithms based short-term load forecasting approach, International Journal of Electrical Power & Energy Systems 45 (2013) 313–324.

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

[182] H. Zheng, A. Kusiak, Prediction of wind farm power ramp rates: A data-mining approach, Journal of Solar Energy Engineering 131 (3) (2009) 031011–1–031011–8.

[183] A. Bossavy, R. Girard, 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.

[184] A. Bossavy, R. Girard, G. Kariniotakis, 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.

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

[186] S. Linden, B. Myers, S. E. Haupt, Observation-based wind-power ramp forecast system, in: Proc. American Meteorological Sciety Annual Meeting, 2012.

[187] S. Soman, H. Zareipour, O. Malik, P. Mandal, A review of wind power and wind speed forecasting methods with different time horizons, in: North American Power Symposium (NAPS2010), Arlington, TX, 2010, pp. 1–8.

[188] Wind power generation data (Apr. 2015).
URL www.elia.be/en/grid-data/power-generation/wind-power

[189] H. He, E. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering 21 (9) (2009) 1263–1284.

[190] J. Olamaee, M. Mohammadi, A. Noruzi, S. M. H. Hosseini, Day-ahead price forecasting based on hybrid prediction model, Complexity 21 (S2) (2016) 156–164.

[191] M. D. Felice, X. Yao, Short-term load forecasting with neural network ensembles: a comparative study [application notes], IEEE Computational Intelligence Magazine 6 (2011) 47–56.

[192] R. Weron, Electricity price forecasting: a review of the state-of-the-art with a look into the future, International Journal of Forecasting 30 (2014) 1030–1081.

[193] R. Kohavi, F. Provost, Glossary of terms, Machine Learning 30 (1998) 271–274.

[194] G. A. Darbellay, M. Slama, Forecasting the short-term demand for electricity: Do neural networks stand a better chance?, International Journal of Forecasting 16 (2000) 71–83.

[195] L. C. Ying, M. C. Pan, Using adaptive network based fuzzy inference system to forecast regional electricity loads, Energy Conversion and Management 49 (2008) 205–211.

[196] S. Van Vaerenbergh, Kernel methods for nonlinear identification, equalization and separation of signals, Ph.D. thesis, University of Cantabria, software available at https://github.com/steven2358/kmbox (Feb. 2010).

[197] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, Journal of the American Statistical Association 32 (200) (1937) 675–701.

[198] P. Nemenyi, Distribution-free Multiple Comparisons, Princeton University, 1963.

[199] R. F. Engle, Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation, Econometrica 50 (1982) 987–1008.

[200] T. Bollerslev, Generalized autoregressive conditional heteroskedasticity, Journal of Econometrics 31 (1986) 307–327.

[201] S. Asadi, E. Hadavandi, F. Mehmanpazir, M. M. Nakhostin, Hybridization of evolutionary LevenbergMarquardt neural networks and data pre-processing for stock market prediction, Knowledge-Based Systems 35 (2012) 245–258.

[202] A. Kazem, E. Sharifi, F. K. Hussain, M. Saberi, O. K. Hussain, Support vector regression with chaos-based firefly algorithm for stock market price forecasting, Applied Soft Computing 13 (2013) 947–958.

[203] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, in: F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc., 2012, pp. 1097–1105.

[204] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735–1780.

[205] X. Ding, Y. Zhang, T. Liu, J. Duan, Deep learning for event-driven stock prediction, in: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015), AAAI Press, 2015, pp. 2327–2333.

[206] R. Akita, A. Yoshihara, T. Matsubara, K. Uehara, Deep learning for stock prediction using numerical and textual information, in: Proc. IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS2016), Okayama, Japan, 2016.

[207] Yahoo finance (Apr. 2017).
URL http://www.finance.yahoo.com/

[208] C. Hoffman, The Industrial Ecology of Small and Intermediate-sized Technical Companies: Implications for Regional Economic Development, Program on the Role of Growth Centers in Regional Economic Development: Discussion Paper, Center for Economic Development, University of Texas, 1971.

[209] M. R. Chertow, Industrial symbiosis: Literature and taxonomy, Annual Review of Energy and the Environment 25 (2000) 313–337.

[210] G. Brundtland, M. Khalid, S. Agnelli, S. Al-Athel, B. Chidzero, L. Fadika, Our Common Future: The World Commission on Environment and Development, Oxford University Press, 1987.

[211] P. Desrochers, Cities and industrial symbiosis: Some historical perspectives and policy implications, Journal of Industrial Ecology 5 (4) (2001) 29–44.

[212] J. Ehrenfeld, N. Gertler, Industrial ecology in practice: The evolution of interdependence at kalundborg, Journal of Industrial Ecology 1 (1) (1997) 67–79.

[213] E. Cimren, J. Fiksel, M. E. Posner, K. Sikdar, Material flow optimization in by-product synergy networks, Journal of Industrial Ecology 15 (2) (2011) 315–332.

[214] I. Kantor, M. Fowler, A. Elkamel, Optimized production of hydrogen in an eco-park network accounting for life-cycle emissions and profit, International Journal of Hydrogen Energy 37 (2012) 5347–5359.

[215] M. Karlsson, The mind method: A decision support for optimization of industrial energy systems principles and case studies, Applied Energy 88 (2011) 577–589.

[216] Z. W. Liao, J. T. Wu, B. B. Jiang, J. D. Wang, Y. R. Yang, Design methodology for flexible multiple plant water networks, Industrial & Engineering Chemistry Research 46 (14) (2007) 4954–4963.

[217] A. P. Melo, D. Cóstola, R. Lamberts, J. L. M. Hensen, Development of surrogate models using artificial neural network for building shell energy labelling, Energy Policy 69 (2014) 457–466.

[218] F. Boukouvala, M. M. F. Hasan, C. A. Floudas, Global optimization of general constrained grey-box models: new method and its application to constrained PDEs for pressure swing adsorption, Journal of Global Optimization 67 (2017) 3 – 42.

[219] A. Forrester, A. Sóbester, A. Keane, A. I. of Aeronautics, Astronautics, Engineering Design Via Surrogate Modelling: A Practical Guide, Progress in Astronautics and Aeronautics, J. Wiley, 2008.

[220] N. Chen, K. Wang, C. Xiao, J. Gong, A heterogeneous sensor web node meta-model for the management of a flood monitoring system, Environmental Modelling & Software 54 (2014) 222–237.

[221] P. Azadi, G. P. Brownbridge, S. Mosbach, O. R. Inderwildi, M. Kraft, Production of biorenewable hydrogen and syngas via algae gasification: A sensitivity analysis, Energy Procedia 61 (2014) 2767–2770.

[222] P. Geyer, A. Schlüter, Automated metamodel generation for design space exploration and decision-making a novel method supporting performance-oriented building design and retrofitting, Applied Energy 119 (2014) 537–556.

[223] C. A. Kastner, A. Braumann, P. L. Man, S. Mosbach, G. P. Brownbridge, J. Akroydand, M. Kraft, C. Himawan, Bayesian parameter estimation for a jet-milling model using metropolishastings and wanglandau sampling, Chemical Engineering Science 89 (2013) 244–257.

[224] E. Roux, P.-O. Bouchard, Kriging metamodel global optimization of clinching joining processes accounting for ductile damage, Journal of Materials Processing Technology 213 (2013) 1038–1047.

[225] T. Simpson, J. Poplinski, P. N. Koch, J. Allen, Metamodels for computer-based engineering design: Survey and recommendations, Engineering with Computers 17 (2) (2001) 129–150.

[226] J. P. Kleijnen, Kriging metamodeling in simulation: A review, European Journal of Operational Research 192 (2009) 707–716.

[227] S. B. Crary, Design of computer experiments for metamodel generation, Analog Integrated Circuits and Signal Processing 32 (1) (2002) 7–16.

[228] Microsoft COM technical overview (Mar. 2017).
URL https://developer.microsoft.com/en-us/windows/desktop