Techniques to Improve Forecasting Models: Applications to Energy Demand and Price

THI HANG NGUYEN

Doctor Of Philosophy

ASTON UNIVERSITY

August 2010

Aston University

Techniques to Improve Forecasting Models: Applications to Energy Demand and Price

Thi Hang Nguyen

Doctor Of Philosophy, 2010

Thesis Summary

This thesis is a study of three techniques to improve performance of some standard forecasting models, application to the energy demand and prices. We focus on forecasting demand and price one-day ahead. First, the wavelet transform was used as a pre-processing procedure with two approaches: multicomponent-forecasts and direct-forecasts. We have empirically compared these approaches and found that the former consistently outperformed the latter. Second, adaptive models were introduced to continuously update model parameters in the testing period by combining filters with standard forecasting methods. Among these adaptive models, the adaptive LR-GARCH model was proposed for the first time in the thesis. Third, with regard to noise distributions of the dependent variables in the forecasting models, we used either Gaussian or Student-t distributions. This thesis proposed a novel algorithm to infer parameters of Student-t noise models. The method is an extension of earlier work for models that are linear in parameters to the non-linear multilayer perceptron. Therefore, the proposed method broadens the range of models that can use a Student-t noise distribution.

Because these techniques cannot stand alone, they must be combined with prediction models to improve their performance. We combined these techniques with some standard forecasting models: multilayer perceptron, radial basis functions, linear regression, and linear regression with GARCH.

These techniques and forecasting models were applied to two datasets from the UK energy markets: daily electricity demand (which is stationary) and gas forward prices (non-stationary). The results showed that these techniques provided good improvement to prediction performance.

Keywords: adaptive model, wavelet transform, Student-*t* noise, neural network, time series model.

Contents

List of Figures

List of Tables

Acknowledgements

I wish to express my gratitude to my supervisor, Prof. Ian Nabney, for his great support and guidance through my project. I am grateful to him for giving invaluable advice, helpful suggestions and kind encouragement. I would like to thank him for his Netlab toolbox that was used as a part of my codes in this thesis. Without his support I would never achieved what I have done in the thesis.

I would like to thank E.ON for their financial support. I am also grateful to E.ON staff, notably Greg Payne, David Turner, Sally Friend, Matthew Cullen, Nick Sillito, John Bateman, Stuart Griffiths, Daniel Crispin and David Jones, who provided the data, valuable advice and support. Especially, I am highly appreciated Greg Payne for his helpful discussions as well as his great assistance. Thanks are also due to Sally Friend, Matthew Cullen, and Nick Sillito for their support to help me getting the funding for my study.

I also sincerely thank to the examiners, Prof. James Taylor and Prof. David Lowe, for their insightful comments and advice which have significantly improved the thesis.

I would like to thank the members of the Non-linearity and Complexity Research Group (NCRG), especially Prof. David Lowe, Prof. David Saad, Dr Juan Neiroti, and Dr. Gabriele Migliorini for their lectures that helped me a lot in shaping the initial ideas and fundamental knowledge of this thesis. I extend my appreciation to Prof. David Lowe for his advice and his providing me essential research papers at the beginning of my research. They were very helpful to me. I thank Dr. Dan Cornford for his suggestions and advice in my first year viva and his Matlab code for the partial auto-correlation function. Many thanks to Vicky Bond, Kanchan Patel and Dr. Laura Rebollo-Neira.

I would thank to all my classmates and the PhD students in NCRG. I would also like to thank Kevin Murphy for his Matlab code on Kalman filter (available at http://www.cs.ubc.ca/ murphyk/Software/Kalman/kalman.html). I adapted some of these codes to my thesis.

Finally, I must extend my thanks to my parents, my sister and to Thanh, my beloved husband. Their love, encouragement and support are invaluable to me.

1 Introduction

1.1 Motivation of the thesis

Prediction is known to be an important tool and is applied in a wide range of domains, including business, government, economics, finance, medicine, industry, etc. Forecasts do not stand alone, but are part of a broader business process: managing, controlling, planning or scheduling systems. For example, long-term forecasts of telecommunication demand are used for network planning, a company predicts their product sales for each month in order to schedule production and determine staff requirement (Montgomery et al., 2008). Applications of predictions are various, but the main objective is to reduce the risk in decision making.

In the UK energy markets, prediction is mostly aimed at power/gas demand and prices. We focus on the wholesale energy prices rather than retail prices. A special characteristic of electricity is that it cannot be stored easily[1]. Therefore, an electricity demand forecast is valuable for power generators who can use such forecasts to effectively schedule operations of their power stations to match generation capacity with demand. Electricity demand

[1] There are some large hydro schemes to store electricity, but they only amount to a small fraction of overall demand.

forecasting is also considered one of the fundamental pieces of information for trading in the energy market because the power price depends on demand.

Accurate electricity/gas price forecasting is vital for traders in the energy market. The fluctuations in electricity demand and the requirement of balancing between demand and supply are the principal causes of high volatility in wholesale electricity prices. This makes accurate price forecasts even more important. If a market participant makes an accurate forecast of the market price, it can develop a strategy to maximise its own profits and minimise risk due to price spikes by appropriate trading in forward contracts. An energy generator can plan its actions to maximise benefits or utilities by reducing/increasing its generation. In addition, understanding the process of forward price development can help the generators make additional profits through trading on the forward market.

Forecasting problems have been investigated for decades and numerous statistical models have been developed. However, forecasting energy prices presents a number of challenges because of the volatility characteristics of the prices. Although many efforts have been made to develop prediction methods, current methods still have significant deficiencies in this domain. As a result, the challenge of developing new methods able to better solve difficult problems still attracts the interest of researchers. In addition, much work remains to be done in this area since some models still require training algorithms for new characteristic encountered (e.g. a model with a new noise distribution rather than the usual Gaussian).

E.ON UK is a part of the E.ON Group, generating and distributing electricity, and retailing electricity and gas. The organisation explores promising new techniques to apply to its core business. This project is a part of their program: forecasting energy demand and forward prices. The main objective of this thesis is to develop techniques to improve performance of some existing forecasting models. We focus on prediction applications in the energy sector with one-day-ahead horizons. These forecasts are used for trading on the forward market and scheduling power plants. These forecasting algorithms are tested on two datasets: daily electricity demand and gas forward price. These datasets were selected to meet the needs of E.ON and also to investigate both stationary and non-stationary data.

1.2 Prediction problem

1.2.1 Scope of the thesis

Prediction is defined to be the process of estimating future values (which have been not observed yet) of a variable given known values of that variable and (perhaps) other related variables[2]. The variables are numerical quantities and they can be electricity demand, stock price, telecommunication demand, population, unemployment rate, or any numerical quantity, depending on the application domain. The variable whose future values we wish to predict is called the *target variable*. The forecast values are normally derived from a model (with defined parameters) and inputs.

The range of prediction problems is large and diverse. We can classify prediction problems using several characteristics. First, forecasting can be classified by *forecast horizon*, which is the time ahead that we want to forecast values. It can be a short-term, medium-term, or long-term forecast. The categorisation of lengths of forecast horizons depends on the application domain. For example, in the energy demand/price prediction, the short-term involves forecasting a few time periods (minutes, hours, days, or weeks) ahead. The medium-term forecast can be several months to a year in the future and the long-term can extend to several years. Second, with regard to the choice of the type of *forecasting model*, there are various forms that can be used to generate input-output mappings for prediction: neural networks, time-series models, mathematical formulae, etc. These models allow us to compute forecasts from the observed or known values. The third aspect of the prediction problem relates to the *form* of forecast. Usually, the forecast is the expectation (or mean) of a future value. Some applications require not only a forecast of the mean but also a measure of uncertainty, such as standard derivation, confidence interval, or full probability distribution.

Within the scope of this thesis, we are interested in predicting the one-day-ahead value of energy demand and price. We focus on predicting means of the target variables but not their uncertainty measures.

1.2.2 Formal definition

The prediction problem within the scope of our thesis can be formalised as follows:

[2] The term 'prediction' has a very broad meaning. It can refer to predicting the possibility of an event happening (e.g. earthquake or volcanic eruption) or actual occurrences (e.g. lottery). However, we are here interested in the future values of numerically measured quantities.

$$\begin{aligned}\text{given} \quad & \{y_1, y_2, \ldots, y_t\} \text{ and } \mathbf{x}_{t+1} \\ \text{estimate} \quad & \widehat{y}_{t+1} \\ \text{subject to minimising} \quad & E(\widehat{y}_{t+1}, y_{t+1})\end{aligned}$$

where

- $\{y_1, y_2, \ldots, y_t\}$ are the observations of the *target variable* y at time step $1, 2, \ldots, t$.
- $\mathbf{x}_{t+1}$ are the observations of the *input vector* $\mathbf{x}$ at time step $t+1$. The input vector can include some observed values of target y and (optionally) known values of related variables up to time t.
- $\widehat{y}_{t+1}$ is a *forecast value* of y at time step $t+1$. This forecast is made at time step t.
- y_{t+1} is the real value of y at time step $t+1$.
- $E(\widehat{y}_{t+1}, y_{t+1})$ is an *error function* measuring how close the forecast $\widehat{y}_{t+1}$ and the real value y_{t+1} is. Some error functions are defined in Section 3.5.

The main job of prediction is to statistically estimate a functional relationship between input vector $\mathbf{x}$ and forecast:

$$\widehat{y} = f(\mathbf{x}, \theta),$$

where function $f(\mathbf{x}, \theta)$ is called the *forecasting* or *prediction model* and θ is the set of model parameters. The process of finding the structure of the function f and estimating parameters θ is called training. In this thesis, we used statistical models in which the target y is assumed to be corrupted by a zero-mean *noise* random variable ε:

$$y = f(\mathbf{x}, \theta) + \varepsilon, \tag{1.1}$$

where the noise distribution is usually assumed to be Gaussian.

1.2.3 Forecasting process

A forecasting process has six major steps as follows[3]:

[3] This forecasting process is based on a standard data mining process CRoss-Industry Standard Process for Data Mining (CRISP-DM) at http://www.crisp-dm.org/index.htm.

- *Specification*: define the problem and clarify the objective by answering some questions: what do we want to forecast, what is the time horizon of the forecast, and what is the form of the forecast? We have answered these questions in Section 1.2.1.

- *Data understanding*: in this phase we analysed the data that E.ON had provided in order to understand the relationships between variables and recognise underlying patterns (e.g. trends, seasonality) in the data. These variables include the targets and exogenous variables which are potential inputs for prediction models (see Chapter 2). We also did some data cleaning: replacing missing values by the average of two adjacent values, identifying and replacing the outliers. Few values in electricity demand time series are abnormally high, and we replaced them by values from the previous week.

- *Data pre-processing*: create input vectors. We must form input variables that are appropriate for the target variable and model structure. This is one of the main areas of study in this thesis. We used some processing procedures to select the input variables for each kind of prediction model (see Section 3.4). This step also included some analytical processes to enhance prediction accuracy, such as applying a wavelet transform to derive new attributes for input vectors (to be presented in Chapter 4), replacing irregular data on holidays in electricity demand time series (see Section 2.5.1), transforming temperature to be quasi-linear with electricity demand, and using dummy variables to represent non-numerical data such as the day of the week (see Section 3.6.1 for more detail).

- *Modelling:* build forecasting models from training sets. This step defines forecasting model structures and fits the models to the data. Fitting means we estimate their parameters using some optimisation criterion. We use various statistical models, including machine learning, time series, and financial stochastic models (to be presented in Chapter 3). Chapter 5 and 6 propose two techniques to improve performance of these prediction models.

- *Evaluation:* verify the models developed in step 4. This step measures how close the forecasts are to the corresponding real values. The model accuracy should be tested on an out-of-sample dataset because the error on the training data cannot be relied upon to estimate the accuracy of a model. Hence we have to divide each dataset into a training set and a test set. The training set is used in step 4 to develop models while the test set is used in this step to evaluate the models. We used some standard error measures and also defined a new one to evaluate and compare prediction performance (see Section 3.5).

- *Deployment:* once a model is confirmed to be reasonably accurate, it can be used in the real world. This step is the responsibility of E.ON with the support of Aston in technology transfer. All the software developed in the project has been delivered to E.ON with documentation and E.ON staff have received some training in the methodology. They will decide how and when the knowledge in this thesis will be used for their real world applications.

1.3 Major contributions

In this thesis, we focus on the data pre-processing, modelling, and evaluation steps. The main contents of the thesis are three techniques to improve the accuracy of some prediction models. Note that these improvement techniques cannot stand alone, but they have to be combined with the prediction models to make them more successful.

This thesis provides an empirical comparison of a set of forecasting frameworks in order to explore the following issues:

- Because the above improvement techniques cannot do prediction by themselves, we used some machine learning and time series models as standard forecasting models (i.e. the form of function f in Equation (1.1)): a multilayer perception (MLP), a radial basis function (RBF), a linear regression (LR), and a linear regression with generalised autoregressive conditional heteroscedastic (LR-GARCH) model. These forecasting models are basically used to test the performance of the improvement techniques.

- With regard to a transformation of the target variable prior to modelling, this thesis uses the wavelet transform (WT) to generate new variables for input vector $\mathbf{x}$ in Equation (1.1). We compared performance of the prediction models without WT and two combination methods:

    - *Multicomponent forecast*: a WT decomposes the target value y into wavelet components, and then each component is forecast with a separate model. The forecast value of y is computed by an inverse wavelet transform.

  - *Direct forecast*: using the components of the WT as input variables to a single forecast model to directly predict the target.

- With regard to model parameters (θ in Equation (1.1)), they are either estimated just once or continuously updated in the testing period. We evaluate the performance of the standard forecasting models with two variations:

  - *Fixed models*, i.e. models whose parameters are fixed after training on a training set.
  - *Adaptive models*, i.e. hybrid of filters (extended Kalman filter (EKF) or particle filter (PF)) and forecast models, where parameters are estimated on a training set and then adapted continuously on the test set using the filter.

In terms of this factor, we also proposed adaptive models for the financial stochastic models.

- With regard to the noise distributions of the dependent variables in the forecasting models (i.e. ε in Equation (1.1)), we use either *Gaussian* distributions or *Student-t* distributions.

By combining the above factors, there are 60 different prediction frameworks. We tested these prediction frameworks for forecasting one-day-ahead electricity demand and one-day-ahead gas forward price in the UK market. Two large datasets are used: (1) electricity daily demand with 821 observations, and (2) 24 sub-datasets of gas forward prices.

Compared with earlier work, our thesis makes the following contributions. First, we propose new forecasting frameworks with various combinations of different methods: wavelet transform, a range of machine learning/time series models, filters, and different noise distributions. Second, although combining WT with a time series or neural network model has already appeared, previous papers only used either multicomponent-forecast or direct-forecast. In this thesis, we use both types of forecast and compare their prediction accuracy, which provides an answer to the question of which is better for energy datasets.

The experimental results on the UK data showed that multicomponent-forecasts consistently outperform direct-forecasts and the models without WT. Third, we combine filters (EKF/PF) with machine learning/time series models to create adaptive models, whose parameters are updated online during forecasts. Among these adaptive models, the adaptive LR-GARCH and adaptive financial stochastic models are proposed for the first time in this thesis. Moreover, we use not only the EKF for adaptive models as earlier authors but also the PF. The benefits of using the PF are that it makes no *a priori* assumption of Gaussian noise and also that it is not necessary to linearise the prediction models. Fourth, we consider the use of either Gaussian or Student-t as noise distribution in prediction models. Student-t noise showed good effects on the gas price data whose residuals are well known to be fat-tailed distributions. We proposed a novel training algorithm for Student-t models. This algorithm is an extension of earlier work (Tipping and Lawrence, 2005) for models that are linear in parameters to the non-linear MLP. Therefore, our proposed training technique broadens the range of models with Student-t noise model. Finally, besides historical data of a target variable (e.g. electricity demand or gas forward price) and its WT components, a number of exogenous variables (e.g. temperature, wind speed, day pattern, electricity supply and electricity price etc.) are also considered as input variables. Some pre-processing procedures (presented in Section 3.4) are used to choose the relevant input variables for each forecasting model.

1.4 Structure of the thesis

This thesis is organised as follows:

Chapter 2 presents an overview of the UK energy market. We focus on energy demand and prices because they are target variables of our prediction tasks. We also describe the related variables which are potential inputs for these tasks. In Section 2.5, the two datasets which were used to test the performance of proposed algorithms in this thesis are described.

In Chapter 3, we present an initial analysis and some results on the input variable selection and the standard forecasting models. They are basic steps to develop the improvements in the next chapters.

Chapters 4, 5 and 6 study three techniques to enhance the performance of the models presented in Chapter 3. The techniques are aimed at different aspects of the prediction task: pre-processing data, parameter estimation, and noise distribution. In Chapter 4, wavelet transforms are used as a pre-processing procedure. This chapter studies the question of which types of WT can be used in forecasting applications, discusses different approaches for using WT in prediction, and empirically compares their performances.

Chapter 5 discusses another technique to make these forecasting models more accurate: adaptive models. We identify the situations in which this technique is effective.

In Chapter 6, we investigate noise distribution issues. This chapter discusses the need to use models with Student-*t* noise for energy price time series. Then we propose a novel methodology for inferring parameters of Student-*t* probabilistic models.

Chapter 7 carries out an empirical comparison to evaluate the effectiveness of the above improvement techniques when they separately combined with standard prediction models as well as the benefit when they are cumulatively combined together.

In chapter 8, we conclude the thesis, summarising algorithms proposed in the thesis and their performance. We suggest several related research topics which may be pursued in the future to improve and extend the methods described in this work.

1.5 Publications resulting from this thesis

There are some publications resulting from this work as follows:

1. H. T. Nguyen and I. T. Nabney. Combining the wavelet transform and forecasting models to predict gas forward prices. In *The Seventh International Conference on Machine Learning and Applications, ICMLA'08*, pages 311-317, 2008.

2. H. T. Nguyen and I. T. Nabney. Energy forward price prediction with a hybrid adaptive model. In *IEEE Symposium on Computational Intelligence for Financial Engineering, CIFEr 2009*, pages 66-71, 2009.

3. H. T. Nguyen and I. T. Nabney. Energy demand and price forecasts using wavelet transforms and adaptive machine learning models. *Energy,* 35 (9), pages 3674-3685, 2010.

4. H. T. Nguyen and I. T. Nabney. Variational inference for Student-*t* MLP models. *Neurocomputing,* 73(16-18), pages 2989-2997, 2010.

1.6 Abbreviations

ACF	autocorrelation function
ANN	artificial neural network
AR	autoregressive
ARMA	autoregressive moving average
ARIMA	autoregressive integrated moving average
ARD	automatic relevance determination
BM	benchmark model
CD	correct direction
CM	correlation matrix
EK	Kalman filter
EKF	extended Kalman filter
FT	Fourier transform
GARCH	generalised autoregressive conditional heteroscedastic
KF	Kalman filter
IR	improvement ratio
LR	linear regression
MAE	mean absolute error
MAP	maximum a posterior
MAPE	mean absolute percent error
MLP	multilayer perceptron
NLL	negative log likelihood
NMAE	normalised mean absolute error
NRMSE	normalised root mean squared error
PACF	partial autocorrelation function
PF	particle filter
RBF	radial basis function
RHWT	redundant Haar wavelet transform
RMSE	root mean squared errors
ROC	renewable obligation certificate
RW	random walk model
SAP	system average prices
SCG	scaled conjugate gradient
SMP	system marginal prices
SSM	state space model
WT	wavelet transform

2 Energy markets

The datasets used in this thesis were derived from real information from the UK energy markets, which was provided by E.ON. The proposed algorithms were tested on two forecasting problems: daily electricity demand and gas forward prices. This chapter gives an overview of the UK energy markets and the datasets.

2.1 Introduction

In the UK, electricity comes from a number of generating sources, including coal, oil, gas, nuclear, solar, biomass, wind and hydro. The contributions of each of these fuels have changed over time due to different factors: economic, political, and technological. Coal, gas, and nuclear power stations provide the majority of the generating capacity (see Figure 2.1). If we take the period 1970 - 2007 as a whole, coal has been the predominant fuel, generating 54% of all the electricity in the UK, followed by nuclear (21%), natural gas (13%) and oil (10%) (Davies, 2009). From the 1980s the contribution of coal reduced due to a broader trend whereby the UK economy moved from traditional heavy industries like

Figure 2.1: Fuel mix for electricity generation in the UK (Source: Davies (2009)).

coal to a more service-based economy. Since 1990, North Sea gas has been used to supply the UK with cheap fuel for electricity generation. In addition, the development of gas-power generation technologies has made gas more and more attractive and it has become more important in the UK electricity industry. These facts motivate the transition of the generating source from coal to gas. Gas made a very small contribution in 1990, but it is now dominant. In 2005, gas power contributed 39% of all electricity in the UK, which surpassed coal (35%) and nuclear (20%) (Wiltsher et al., 2006).

In recent years, the problem of global warming has attracted a great deal of attention and is considered as one of the biggest environmental challenges facing the world. Moreover, the government is also concerned with the insecurity of future supplies of natural gas. These factors motivate the development of renewable energy, for example small hydroelectric plant (run of water), wind (both onshore and offshore), wave power, etc. However, the capacity of electricity from renewable sources is still insignificant due to restrictions of the technologies and the expensive producing cost. In 2007, renewable energy contributed only 5% of all electricity generated in the UK (Davies, 2009).

In 2002 the Renewable Obligation Certificate (ROC) was introduced, which is aimed at increasing the amount of electricity generated from renewable energy sources and reducing CO_2 emissions. All companies are required to supply a minimum proportion of their electricity from renewable sources (in 2009, this proportion was 9.1% (Davies, 2009)). A company can meet this requirement by either generating renewable energy or buying ROCs from other companies. ROCs can be traded in the open market by bid/offer mechanisms. At the end of each year, if a company does not have enough ROCs to meet the requirement, they will get a penalty: they must pay into a "buy out" fund. Then, this "buy out" will provide funds to for the companies which have presented sufficient ROCs for that year.

The most important objective of suppliers is to get the greatest benefit from their business. The suppliers are encouraged to balance the energy supply and demand by controlling power stations or trading energy contracts. If a supplier fails to balance supply and demand, fines can be imposed, and National Grid as the system operator[1] in Great Britain sells/buys energy to correct aggregate imbalances. In long term plans, the suppliers can make profits from buying/selling power stations or planning to build new power stations (which is a complex task because they must pay construction and running fees, compute effectiveness of investment, and ask for approval from the government). In addition, they may also make additional profits through trading on the energy forward market.

There are two trading levels in the energy markets, i.e. wholesale and retail. In the wholesale markets, the participants (including system operators, generators, suppliers and traders) trade a large amount of gas and electricity among themselves. Then, these suppliers sell their purchased gas/electricity to their customers (i.e. consumers which can be residents or business users) in retail markets. A supplier is not necessarily a generator. In this thesis we focus on the wholesale markets only. Trading in the wholesale markets is based on forward contracts and spot markets. The forward contract is for future delivery of energy (e.g. a month ahead or season ahead) while the spot market is for buying or selling energy within the day. We will discuss these in more detail in Section 2.3.

The energy suppliers also have to manage spikes in the market. Electricity and gas prices sometimes show very large and unpredictable spikes, which may result from various reasons: shutting down a power station, extremely cold weather, etc. If the suppliers can predict these opportunities, they can get a large benefit. If a company owns a power plant, it can increase its profits by switching the plant on when electricity prices are high relative to the price of inputs.

[1] The system operator is an organisation which is responsible for controlling and managing the operation of gas or electricity markets in the most effective and economic maner. They also ensure a continuing supply-demand balance.

2.2 Energy demand

In this thesis, the electricity and gas demand on a day are taken to be the total consumption of all users in Great Britain (including industry, the commercial sector and the residential sector) over the whole day. The basic unit for electricity and gas demand are Megawatt hours (MWh) and therm respectively.

Figure 2.2 shows daily electricity demand from 7^{th} October 2004 to 3^{rd} May 2007. The data has a yearly seasonality pattern, caused by temperature variations. Electricity demand is highly sensitive to the weather conditions. It is normally higher during the colder part of the year, mainly because of the use of heating. The figure also shows that there are some days which have much smaller demand than the adjacent days, for example observations around time steps 180, 450 and 810. They are public holidays (Christmas, Easter and Bank holidays). Note that there is a large difference in demand between years. For example, observations around time step 100 and those around time step 465 (see Figure 2.2) present data on the same periods of two adjacent years (2005 and 2006), but the values of these two observation groups are differ by about 1.1×10^5 (MWh), i.e. approximately 10% of the largest consumption days in 2005 .

If we "zoom in" on a shorter period for more detail (see Figure 2.3), we see that daily electricity demand also has a periodicity of seven days with lower consumption at weekends. Most offices and schools close on weekends, when their energy consumption is much reduced. Consequently, the total electricity demand drops significantly at weekends. There are normally peaks on Tuesday or Wednesday. The lowest demand is on Sunday.

Daily gas demand shows only annual seasonality, but not weekly seasonality. It is still sensitive to temperature: gas demand in winter is higher than in the summer (see Figure 2.4).

Unlike most commodities, electricity has a distinctive character, i.e. it cannot be stored easily. It is nearly impossible to produce a large amount of electricity at a time of low demand, hold it and consume it later. This means that supply has to match consumption at (nearly) all times. Therefore, electricity generators and suppliers need to have good predictions of demand; then they can save money by planning/scheduling the power stations based on these predictions; the suppliers have good plans of buying/selling electricity in the wholesale market.

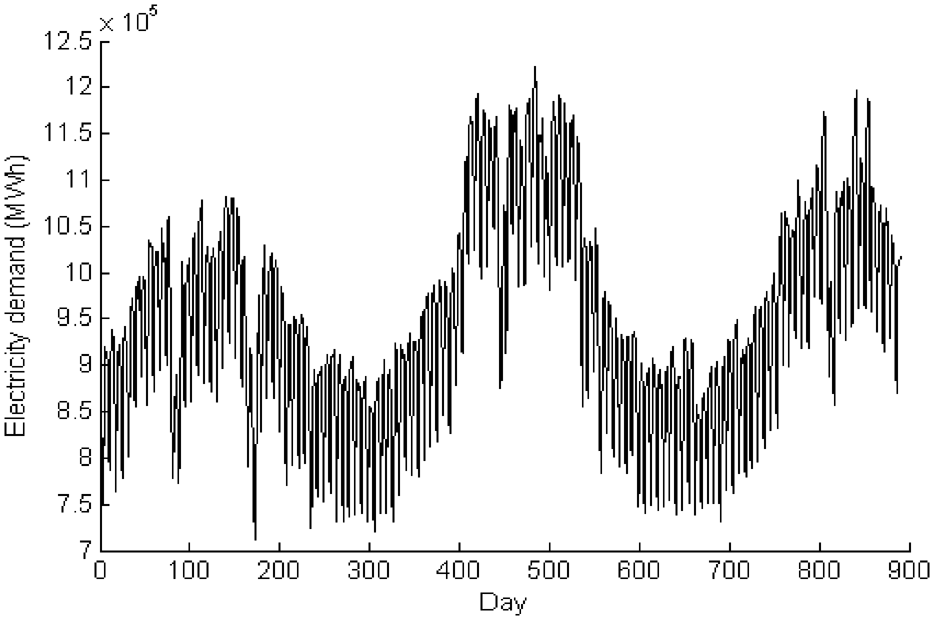


Figure 2.2: Daily electricity demand from 7th October 2004 to 3rd May 2007.

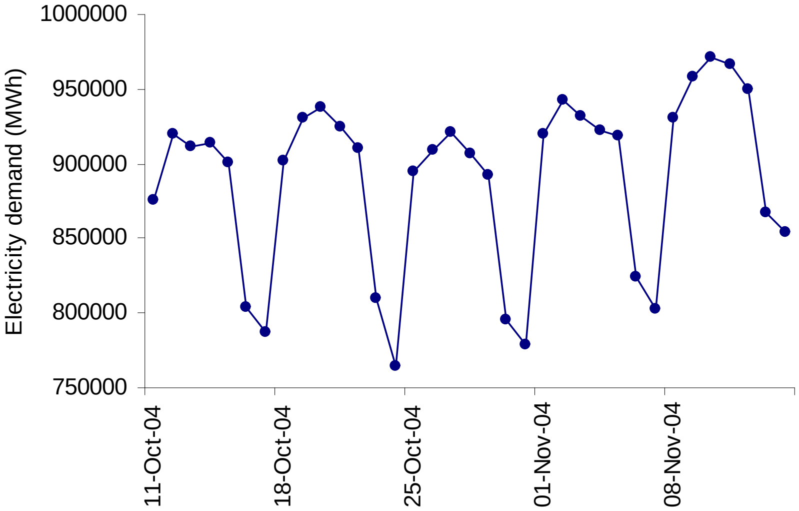


Figure 2.3: Daily electricity demand from Monday 11th October 2004 to Sunday 14th November 2004.

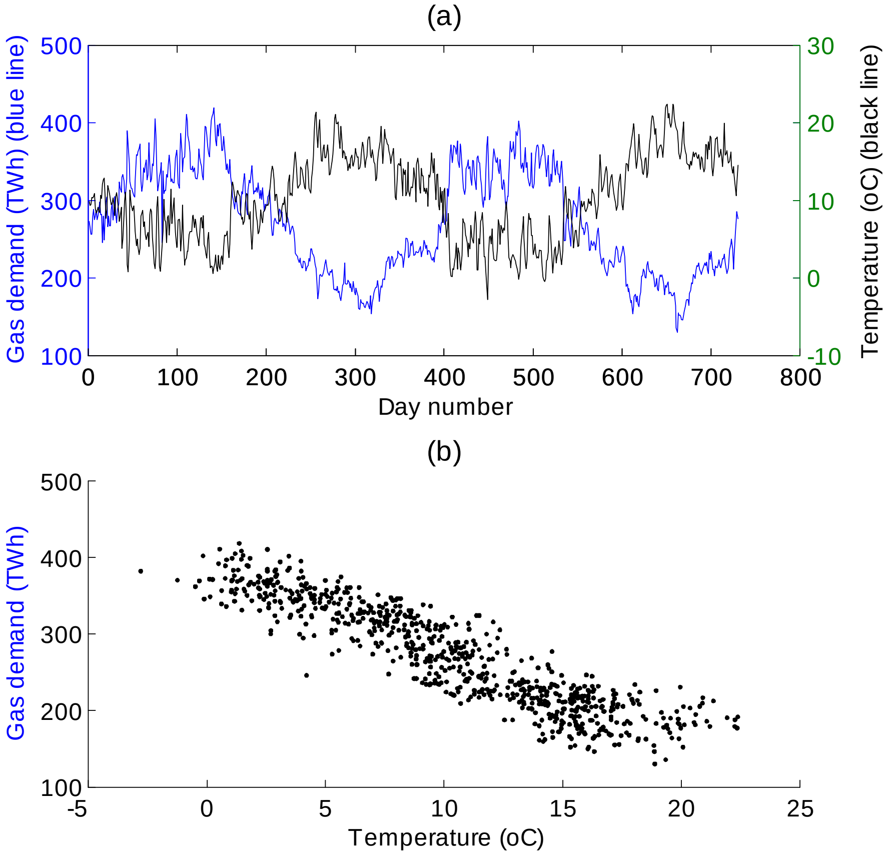


Figure 2.4: (a) Daily gas demand and temperature during the period from 7th October 2004 to 6th October 2006. The blue line is the gas demand and the black line is the temperature. (b) Daily gas demand plotted against temperature.

2.3 Energy forward prices

Forward contracts play an important role in the wholesale market. They are an agreement between two participants in the market to sell/buy a given amount of electricity or gas for delivery over a specified future time period at a certain price. A difference between electricity/gas contracts and most other commodity contracts is that the delivery time of an electricity or gas contract is a period of time rather than at a specific future time point. This is the nature of all electricity contracts because electricity is non-storeable and it is beneficial only if it is used over a period of time. The main objective of suppliers and generators trading the forward contracts is to reduce or avoid risks that they may face due to price changes in the future. Because forward contracts are not traded at weekends and public holidays, each year has approximately 250 trading days.

There are various types of forward contracts depending on the length of the delivery period: monthly, seasonal, quarterly and annual contracts. The delivery period of a monthly gas forward contract normally starts at the beginning of a month and stops at the end of that month. For example, the delivery period of Oct-2006 gas product is from 1^{st} October 2006 to 31^{st} October 2006. No forward contracts is traded once its delivery period has begun. A monthly gas forward contract is available for trading during the contract trading period. In the data that we received from E.ON, the trading period was six months for trading prior to 30^{th} April 2007 but has been five months since that date[2]. Therefore, there are five or six months of daily price data (approximately 110-130 data points) for each monthly gas product. In the above example, Oct-2006 product can be traded on all days from 3^{rd} April 2006 until 29^{th} September 2006 (except public holidays and weekends). At the beginning of every month, a monthly contract stops being traded and a new monthly contract is listed.

Unlike gas, where forward products match the standard calendar, the delivery periods of electricity forward products follow a specific calendar, called the Electricity Forwards Agreement (EFA) calendar. This calendar lists a standard set of (monthly/quarterly/seasonal) products that can be traded on the electricity forwards markets; a month in EFA starts on a Monday and it is a rolling cycle of 4-4-5 weeks (with a 53^{rd} week added every so often

[2] The gas forward data is taken from Heren (http://www.icis.com/heren/) during the period before 30^{th} April 2007, but after this date we get information from another source, Argus (http://www.argusmedia.com/). This data source does not make too much difference, other than the fact that Argus only publishes prices for five monthly products at a time, instead of six from Heren.

Type of contract	Length of delivery period	Trading period
Electricity forward price		
Monthly	28, 35 or 42 days	4 months
Quarterly	91 or 98 days	3 quarters
Seasonal	182 or 189 days	5 seasons
Gas forward price		
Monthly	28 - 31 days	6 months (before 30^{th} April 2007) 5 months (after 1^{st} May 2007)
Quarterly	89 - 92 days	11 quarters
Seasonal	172 - 183 days	5 seasons
Annual	365 days	2 years

Table 2.1: Characteristics of gas and electricity forward products in the UK market.

to keep the years aligned). Table 2.1 shows the lengths of delivery periods and trading period of forward contracts in the UK gas and electricity markets.

All forward prices provided by E.ON are daily sampled time series. Because the wholesale markets are very competitive and busy with many participants, every day there are a number of transactions for each product (i.e. same delivery period). However, their prices are not the same. Both parties of a contract can negotiate to set the forward price. The load specified in transactions are also different. In this thesis, the price of a type of product on a certain day is the "close market price", which is the average of prices of all transactions that take place on that day, weighted by load. Gas and electricity prices are quoted in p/therm and £/MWh respectively.

Table 2.2 contains an example of closing prices of all the monthly gas forward products traded from 20^{th} September 2006 to 10^{th} October 2006 in order to show how multiple contracts are available at different times. Forward prices of different products (i.e. different delivery period) are different, even if they are traded on the same day. For example, the price of the Nov-2006 product was 56.05 (p/therm) on 28^{th} September 2006 whereas the price of the Dec-2006 product traded on the same day was 71 (p/therm). The price of products delivering in a cold period are normally higher than those in a hotter period.

There exist shorter delivery period contracts: *weekend ahead* and *weekday ahead,* whose delivery periods are the next working day or the next weekend respectively. In the gas market, there are also within-day contracts which sell or buy gas for delivering gas on the same day. In the electricity market, there are spot contracts which are for delivering power on every half hour within the day of trading; however, we decided not to

Date/ Product	Oct-06	Nov-06	Dec-06	Jan-06	Feb-06	Mar-06	Apr-06
20/09/2006	37.500	58.050	72.750	77.050	75.450	64.400	
21/09/2006	36.450	57.900	72.750	77.500	75.750	64.600	
22/09/2006	34.950	57.125	72.200	77.525	75.725	65.050	
25/09/2006	35.525	57.125	72.275	77.650	75.750	64.550	
26/09/2006	34.850	55.750	70.750	76.300	74.550	63.050	
27/09/2006	33.475	55.125	70.050	75.975	74.125	62.475	
28/09/2006	33.850	56.050	71.000	77.125	75.000	63.400	
29/09/2006	33.500	54.850	70.000	76.550	74.550	63.150	
02/10/2006		51.750	67.600	74.400	72.950	61.000	46.500
03/10/2006		50.450	65.500	72.500	71.350	60.250	45.575
04/10/2006		51.450	65.500	72.200	71.150	60.150	45.300
05/10/2006		52.100	66.150	72.100	70.950	60.350	45.600
06/10/2006		52.125	66.200	71.575	70.450	60.200	45.375
09/10/2006		52.775	67.550	73.700	71.375	60.925	45.450
10/10/2006		51.950	66.875	72.550	70.900	60.425	45.500

Table 2.2: Closing prices (p/therm) of all monthly gas forward products in the period from 20th September 2006 to 10th October 2006.

study this kind of product as it was of less interest to the trading team at E.ON.

Figure 2.5 shows hourly sampled electricity demand during a week. Due to the characteristics of human behaviour, demand is normally higher during working hours (7am - 7pm Monday-Friday). This period of the day is called peak hours. The remaining periods (i.e. 7pm-7am on Monday-Friday & all-day weekend) are off-peak hours. The electricity demand for peak hours can be 50% greater than for off-peak time. Because electricity is nearly non-storable, the power stations which are used to provide “excess” electricity on peak hours have to be flexible on starting and shutting down to balance demand and supply. Therefore, it is more expensive to produce these extra demands on peak hours. Consequently, two versions of electricity forward contracts are available: base load contracts which guarantee to deliver continuous electricity for the whole day (24x7), and peak load contracts which provide electricity in peak hours only. All forward contracts are for delivering electricity as a constant flow during the delivery period. Unlike electricity, gas is more storable: thus there is no concept of base load and peak load for gas products.

Figure 2.6(a) shows an example of the main fuels used to generate electricity within a day. There are different requirements for plants which are devoted to base load and those to peak load. The base load plants have to be able to work continuously for long periods (except time for maintenance or repair). They usually run on low-cost fuels such as nuclear, coal, or hydro (run of water). As mentioned in the previous paragraph, the

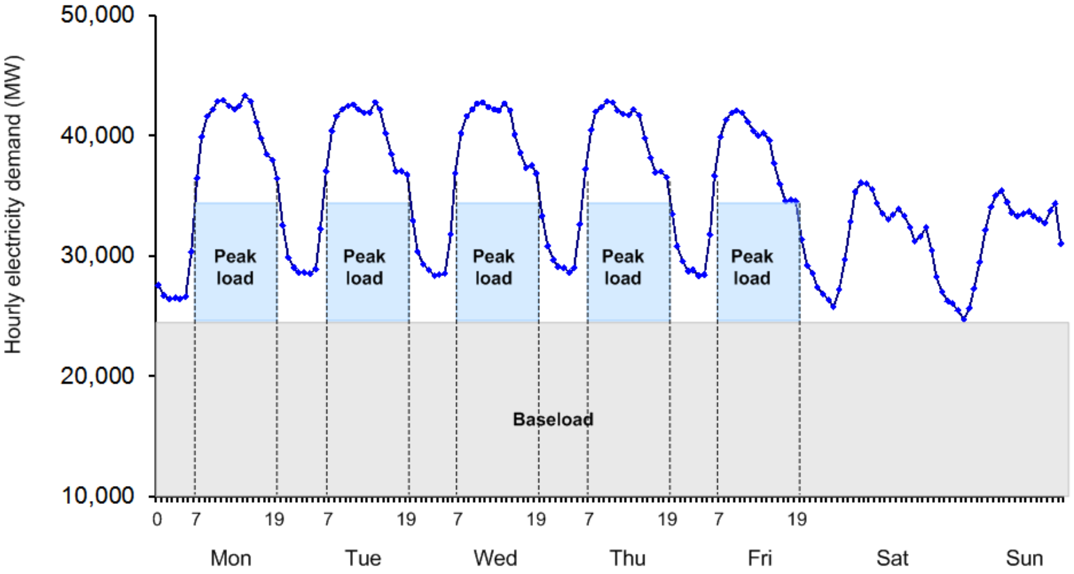


Figure 2.5: Base load and peak load in electricity forward contracts. Data (blue line) is electricity demand (MW), which was sampled at every hour, from Monday 10^{th} July 2006 to Sunday 16^{th} July 2006.

plants for peak load must have the capability of quick start and stop to meet short-term changes in demand. Fuels for peak load can be gas, oil, and coal. Coal and gas have been used to generate both base load and peak load while nuclear fuel can provide the base load only. The figure also shows the increasing generation cost of different fuels. Demand is filled by the cheapest available generations. Of course, peak load plants tend to have higher cost of generation than base load plants, and so prices of peak load contracts are higher than prices of base load contracts.

Figure 2.6(b) shows an example of how a company buys power to meet the demand on a day. The blue and grey regions present load which has been bought from forward contracts in advance. The orange part needs to be bought within the day from the spot markets to balance the supply and demand.

2.4 Related data

We were also provided with a number of exogenous data streams which might drive the gas/electricity demand and price, including:

- *Weather*: sunset time, wind speed, temperature. Temperature (in degrees centigrade) data was measured every hour at 12 weather stations around Great Britain. The temperature used in this thesis is calculated by averaging the temperatures of

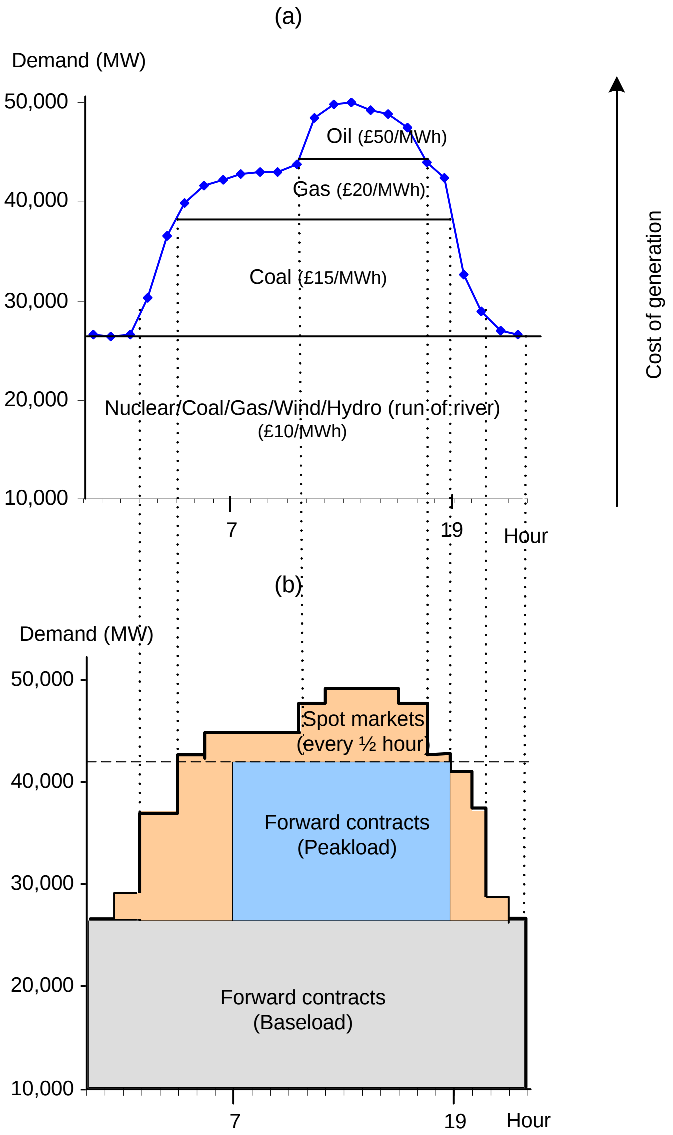


Figure 2.6: An example of (a) Main fuels used for generating electricity, (b) Trading electricity on the spot markets and forward markets to meet the demand.

the whole day for each station to get the daily average temperature of each station, and then averaging these daily average temperatures of all stations. It would be better had the averaged temperature variable been weighted according to the population sizes. This will be discussed in Section 8.2 on page 152. Wind speed was measured in the same way as temperature. Among these weather factors, temperature has the most impact on energy consumption. Figure 2.7 shows the relationship between the daily electricity demand and temperature. In general demand falls when temperature increases, and vice versa. It would have been more appropriate if we use weather forecasts for electricity demand forecasting as in some previous paper (Taylor and Buizza, 2003, Cancelo et al., 2008). However, temperature forecasts were not available during the project, so we used historical actual temperatures for predicting electricity demand (to be described in more detail in Section 3.6.1 on page 62).

- *Day*: the day of the week. This factor affects the customer behaviours and office/school activities. Electricity consumption significantly drops at the weekend.
- *Electricity supply*: total of electricity generated from all power stations in Great Britain.
- *Electricity Interconnector flow:* amount of electricity exported/imported to outside Great Britain. Great Britain currently has interconnections to France and Northern Ireland (Crouch, 2010).
- *Gas demand*: the total amount of gas consumed in Great Britain.
- *System Average Prices (SAP) of ga*s: In the gas within-day market, which sells or buys gas to deliver gas on the same day, there are numerous transactions between different participants on the same day. Because the buyers and sellers can negotiate to get agreements in price and volume of gas, there is no fixed price for all transactions even on the same day of trading. SAP on a day is defined as the weighted average value of the price of all transactions on that day.
- *System Marginal Prices (SMP) of gas*: SMP is based upon actions of the system operator in balancing the system. Although gas can be stored, the pressure in gas storages and pipes have to be keep a range of level, which should not be too high or

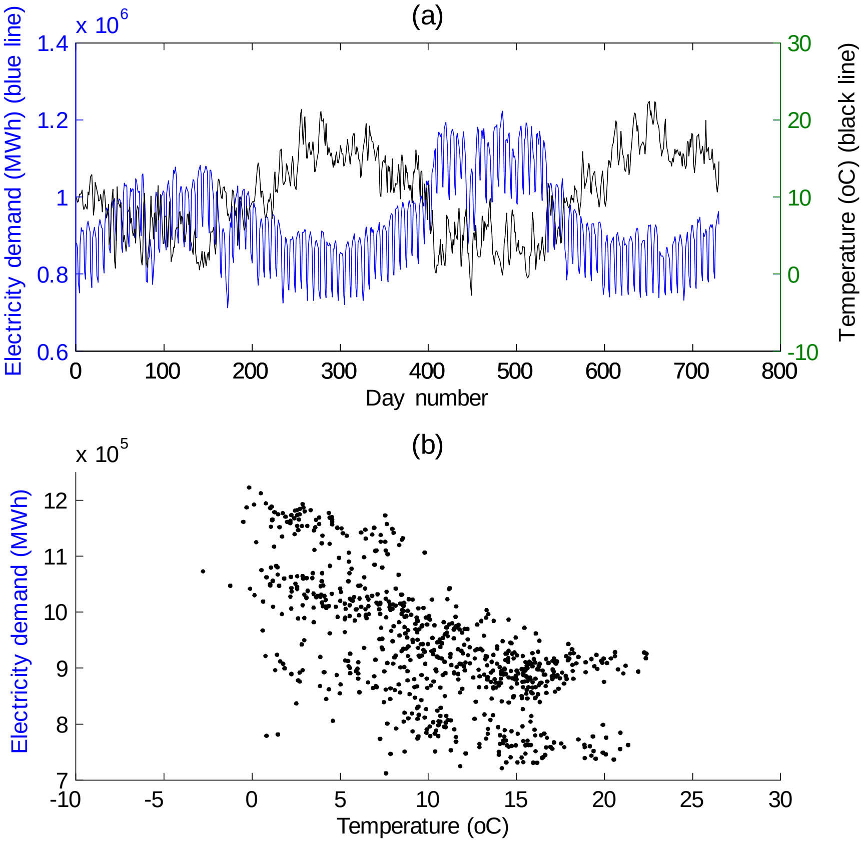


Figure 2.7: (a) Electricity demand and temperature during the period from 7th October 2004 to 6th October 2006. The blue line is the electricity demand and the black line is the temperature. (b) Daily electricity demand plotted against temperature.

too low. Therefore, the suppliers are encouraged to balance the energy supply and demand by trading energy contracts. If suppliers fail to balance supply and demand, the system operator sells/buys energy to correct the imbalances. In addition the system operator impose fines on the failed suppliers. The fines depends on how much the imbalance is. The fine price in case of over supplement is called SMP sell and the fine price in case of over demand is called SMP buy:

  - SMP sell is the lower of (SAP − 0.95p/therm) and (the lowest priced action of the system operator).
  - SMP buy is the greater of (SAP + 0.84p/therm) and (the highest priced action of the system operator).

- *GBP:USD rate:* the exchange rate.
- *Oil spot price.*
- *Events:* events that affect the electricity/gas prices, for example closed storage facilities, unusually cold weather, high continental price, maintenance, etc. According to information provided by E.ON, in the days when the events happen, the price significantly changes. The shorter-delivery period products (such as within-day or day-ahead product) are normally more affected than the long-term forward products (like seasonal or annual contracts) because most of these events affect energy supply/demand only temporarily.

2.5 Datasets in experiments

We evaluated the performance of the algorithms on two problems: forecasting the daily electricity demand and forecasting the prices of monthly gas forward products. These datasets were chosen as requested by E.ON. Moreover we selected them because one of them consists of forward prices which are normally non-stationary and the other is demand which is more stationary[3]. Both datasets were taken from the UK energy market.

[3] The time series is weak stationary if the mean and variance do not depend on time. In the electricity demand dataset, because the electricity demand time series has a longest seasonality of one year, we compute the mean and variance of a one-year window. Then we slide the window to each time step. The mean and the standard derivation do not change by much, less than 0.03% and 0.005% of the largest value in the demand time series respectively. Therefore, the electricity demand time series is quite stationary. Conversely, it is clear that the means of gas forward price time series do significantly change over time.

2.5.1 Electricity demand dataset

The first dataset contains observations of the daily total electricity demand of all users in Great Britain, from 7^{th} October 2004 to 3^{rd} May 2007. The data has yearly seasonality, caused by temperature, and weekly seasonality, caused by human behaviour and economic activity. There are about eight public holidays (Christmas, Easter and Bank holidays) per year. Because of economic activity, the electricity consumption on the public holidays, the days between Good Friday and Easter Monday, and the days between Christmas and New Year are significantly smaller than on other days, even much smaller than on weekends (see observations around time steps 180, 450, and 810 in Figure 2.2). This affects the overall performance of prediction models. We classed these days as special days.

There are two approaches to deal with this issue. The first is to include the special days in the dataset, but introduce a dummy variable (equal to 1 for special days and 0 otherwise). The second is to smooth out the demand on the special days by replacing the demand on those days by the electricity demand on the same day of the closest previous week, which is not a special day. Then, we performed the pre-processing procedures and create input-target pairs. Note that until this step, we have not removed data on the special days, therefore the periodicity is still maintained. After that, if the target of an observation (i.e. input-target pairs) is a special day, we removed that observation out of the dataset. This approach is similar to the smoothing method presented in (Taylor, 2008). In this paper the author performed smoothing out the value on the special days prior to fitting models and predicting; and on the test set the errors associated with the public holidays are excluded from the overall errors of the model. Since the main objective of this thesis is to evaluate the effects of a range of improvements (i.e. wavelet transform, filters, and Student-t noise) on the standard prediction models, we can select any of the above approaches. In this thesis, we chose the second approach.

Figure 2.8 shows the target time series of the first dataset after removing special days, containing 821 observations. The first 525 observations were used as a training set and the last 296 observations were used as the test set.

2.5.2 Gas forward price dataset

The second dataset (see Figure 2.9) contains daily prices of monthly gas forward products from Jun-2006 to May-2008 and is sampled from 1^{st} December 2005 to 30^{th} April 2008.

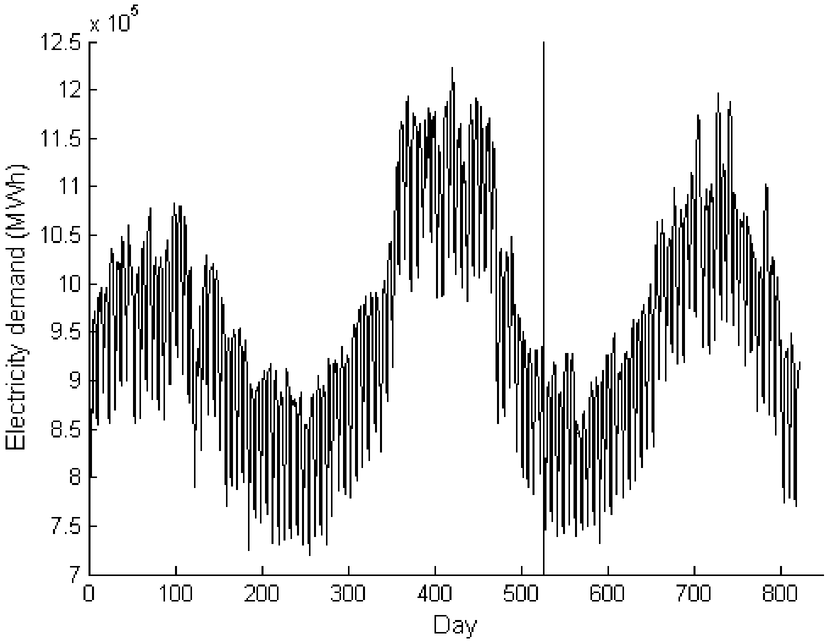


Figure 2.8: Dataset 1: daily electricity demand (MWh). The training set is the earlier section and the test set is the later section. Note that the observations on public holiday has been removed.

As mentioned in Section 2.3 (page 28) there are approximately 110-130 data points for each monthly gas product time series.

We created 24 sub-datasets: each sub-dataset corresponds to the price time series of a single product, in which the first two third of the time series was used as the training set and the remainder was used as the test set.

There exist some abnormal events which affected market behaviours. In the middle of March 2006, Rough, the UK's largest gas storage facility, was closed due to a fire. The facility remained closed for most of the summer, raising concerns about gas supply for the following winter. This made the gas prices of Oct-2006, Nov-2006, Dec-2006, Jan-2007, and Feb-2007 products spike upwards during the summer of 2006. The price gradually decreased after that.

Figure 2.9 shows that there was a large change in the way the market behaved around February 2007. In the period before this milestone, the price trend descends for a long period of time. After that, however, there was a distinct change in the market behaviour: the forward prices stabilised and started to climb. The change happened because the market hit a fundamental floor. This issue will potentially lead to a bad forecasting performance in the period right after February 2007 because their forecasting model should be trained

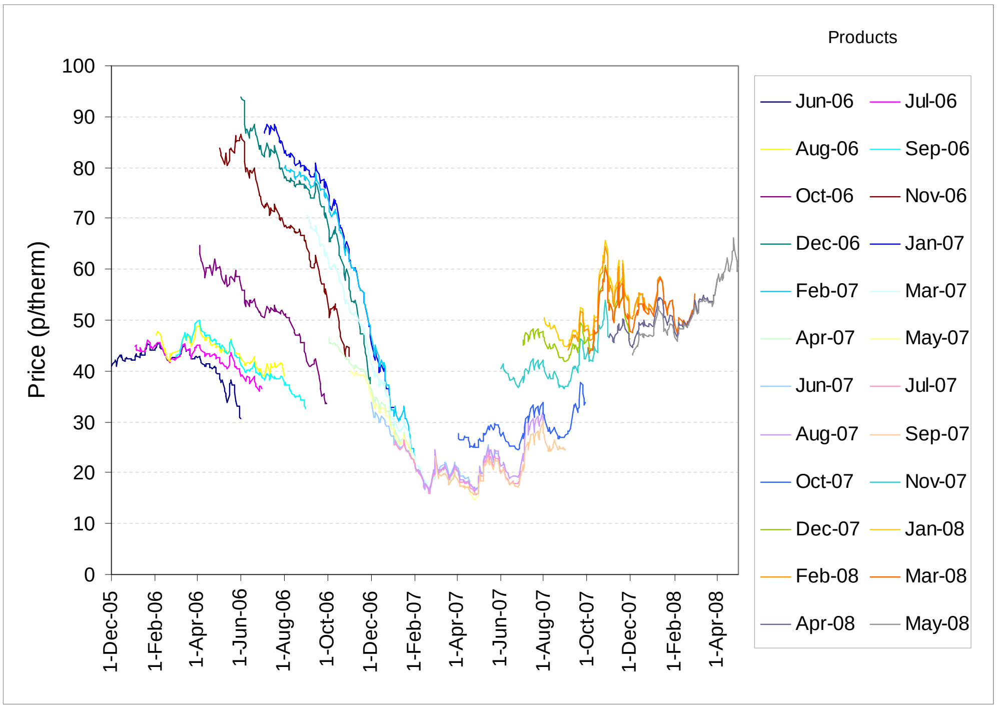


Figure 2.9: Dataset 2: Price (p/therm) of monthly forward gas products Jun-2006 to May-2008. Data is sampled from 1st December 2005 to 30th April 2008.

on data (from before this milestone) whose behaviour is completely different.

Another abnormality in the gas forward prices occurred in the period from September 2007 to February 2008. The volatility in this period was much higher than the other periods because the autumn of 2007 was a start of a huge bull-run[4] in commodities. The most notable bull-run is oil price that started at $70/barrel level and ended at $150/barrel. During this period from September 2007 to February 2008 oil went up from $70/barrel to $100/barrel, which made the volatility of the oil price very high. This affected the volatility of gas forward price as well. In addition, there were some problems with gas pipelines and platforms that contributed to high volatility.

2.6 Summary

This chapter presents an overview of the UK energy market. We have concentrated on defining and describing characteristics of energy demand and forward prices because they are selected to be the targets of predictions. The two datasets deriving from these variables were presented: daily electricity demand and monthly gas forward price. In the next

[4] A bull-run, also called a bull market, is associated with rising or being expected to rise the price in a financial market of a commodity.

chapters, we will study the prediction models and the improvement techniques. These two datasets will be used to test performance of these methods and techniques. This chapter also mentioned a range of related data, such as weather, oil price, GBP:USD rate, etc. Some of them may be highly relevant to the targets and will be considered as candidates for selecting the input variables of the prediction models (to be mentioned in detail in Chapter 3).

3 Predicting time series

3.1 Introduction

As mentioned in Chapter 1, the main objective of this thesis is to develop and evaluate three improvements on standard models for forecasting energy demand and price one-day-ahead. Note that these improvements are not forecasting models themselves. They are used to support some prediction models. We will test the performance of these improvement techniques on some standard forecasting models. This chapter provides the preliminary analysis and results of some standard forecasting models. The later chapters of this thesis will present different approaches to improve the performance achieved in this chapter. We also present the data pre-processing and evaluation in more details here.

Figure 3.1 shows the general process of building forecasting models, which involves two steps: pre-processing and modelling.

- *Step 1*: Analysing data to select the input vectors for each model. As we saw in Chapter 2, there is a large number of variables that are potential inputs for forecasting models. These variables are lags of different types of time series data,

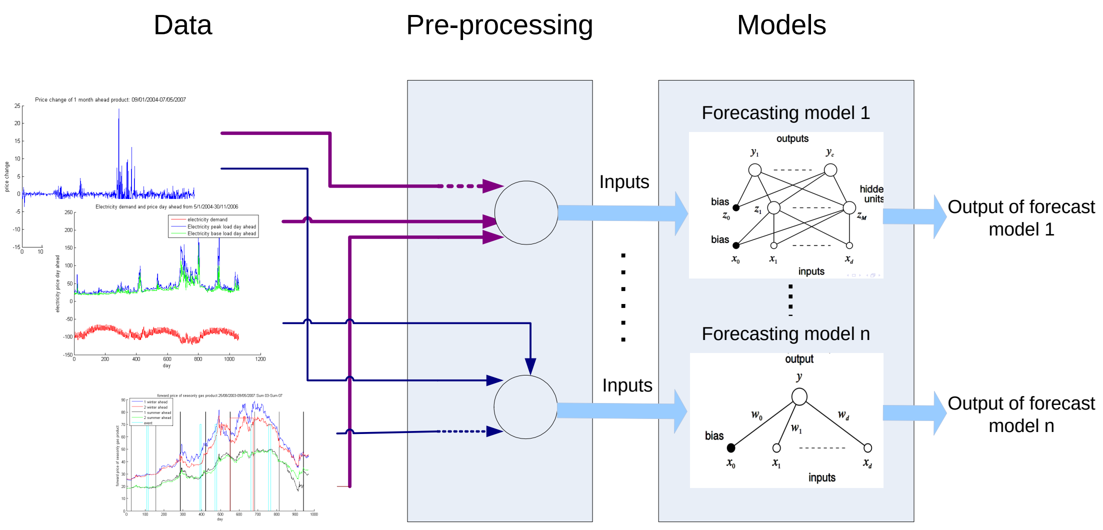

Data
Pre-processing
Models
Price change of 1 month ahead product: 09/01/2004-07/05/2007
price change
Electricity demand and price day ahead from 5/1/2004-30/11/2006
electricity demand
Electricity peak load day ahead
Electricity base load day ahead
electricity price day ahead
day
forward price of seasonly gas product:25/08/2003-09/05/2007:Sum 03-Sum 07
1 winter ahead
2 winter ahead
1 summer ahead
2 summer ahead
event
forward price of seasonly gas product
day
Inputs
Inputs
Forecasting model 1
outputs
y_1
y_c
hidden units
bias
z_0
z_1
z_M
bias
x_0
x_1
x_d
inputs
Forecasting model n
output
y
w_0
w_1
w_d
bias
x_0
x_1
x_d
inputs
Output of forecast model 1
Output of forecast model n

Figure 3.1: Building forecasting models.

such as oil price, temperature, gas forward price, etc. In this step we investigate the relationship between these variables and the target values. The variables that are highly relevant to the target values are selected. Techniques for selecting the inputs will be presented in Section 3.4. The outcomes of this step are matrices of inputs and targets for each dataset.

- *Step 2*: Training forecast models and forecasting the target values. The structures and training algorithms of the prediction models used in this step will be discussed in Section 3.3.

We empirically evaluate and compare the performances of these models and improvements by testing them on real data from the UK market. Section 3.5 describes some measures for evaluating the performance of these forecasting models.

3.2 Related work

Forecasting problems have been investigated for decades and numerous statistical models have been developed. Various forecast horizon values have been studied. Some researchers have forecast data on very short-term horizons (from minutes to hours ahead). These forecasts are very important for real-time scheduling of electricity generation. For example, Taylor (2008) used minute-by-minute data to predict electricity demand from 10 to 30 minutes ahead. In other studies, da Silva et al. (2008), Soares and Medeiros (2008), Taylor et al. (2006) and Nogales et al. (2002) used hourly and half-hourly data to predict power demand several hours to several days ahead. Hourly data was also used in Panagiotelis and Smith (2008) to forecast spot price data. Some others concentrated on studying longer term forecasts like months or years ahead. For instance, predicting monthly electricity demand of Eastern Saudi Arabia was mentioned in (Abdel-Aal and Al-Garni, 1997). Another example can be found in (Akay and Atak, 2007) where two datasets were involved: annual total electricity consumptions and industrial sector electricity consumption for Turkey. The data from 1970 to 2004 were used as training set; then they forecast consumption in the period from 2006 to 2015.

In terms of input variables, we can use historical lags of the time series itself or/and exogenous variables. The two most popular groups of exogenous variables in energy forecasting are calendar-related variables and climate-related variables. Examples of variables belonging to the first group (calendar-related) are public holidays, weekends, or daily/weekly/annual seasonality (Dordonnat et al., 2008, da Silva et al., 2008, Taylor, 2008). These variables were normally represented by dummy variables. Examples of the second group of exogenous variables (climate-related) are temperature, humidity, wind speed, cloudiness, and rainfall. The methods used for climate-related variables are diverse. For example, Taylor and Buizza (2003) used weather ensemble prediction, which includes 51 different scenarios of future values of weather, to predict demand. The results showed that using ensemble prediction outperformed the prediction using a traditional single weather point forecast. In the work by Yan (1998), various types of climate-related variables were combined to generate a single climate variable. In this group of climate-related variables, temperature was reported as the most important input for electricity demand (Moral-Carcedo and Vict̕ens-Otero, 2005, da Silva et al., 2008). Most of the studies on this variable (temperature) focus on analysing the non-linear relationship between electricity demand and temperature (Cancelo et al., 2008, Valor et al., 2001). In this thesis, beside these exogenous variables, we also consider other related variables, such as electricity/gas forward price, oil price or exchange rate, as candidates for inputs of forecasting models.

Although there are many input variables, not all of them are relevant to the target. Consequently, using all of them as inputs not only is computationally expensive but also potentially reduces the prediction accuracy of the forecast models. Therefore selecting appropriate input variables for each kind of model is very important. However, many papers in the literature have overlooked this issue. There are few published papers in energy prediction that have systematic analyses on selecting input variables. Mandal et al. (2005) proposed an approach to selecting input variables, in which the load and prices were forecast by choosing days that are similar to that of the forecast day. The selection of similar days was performed by an Euclidean norm with weighted factors. da Silva et al. (2008) presented two types of procedure for automatically selecting inputs to neural networks. The first type, based on filters, analyses input relevances using statistical tests. The second type, called Bayesian wrappers, evaluates the usefulness of each input by estimating the variance of the corresponding weights. This method is also known as automatic relevance determination (ARD) and it has been studied previously (Nabney, 2002, MacKay, 1994).

The most common topic in forecasting is modelling. A number of statistical methods have been proposed for energy price and demand forecasting. In general, we can classify the prediction methods into three groups: time series models, machine learning models, and financial models. The recent application of machine-learning and time series to these problems has given promising results. In the first group of forecasting models, most papers have studied autoregressive (AR) models (Nogales et al., 2002), autoregressive moving average (ARMA), and autoregressive integrated moving average (ARIMA) models (Contreras et al., 2003, Conejo et al., 2005, Zhou et al., 2006, Taylor, 2008). In the study by Zheng et al. (2005), Garcia et al. (2005), generalised autoregressive conditional heteroscedastic models (GARCH) have been used for price time series. The GARCH model takes into account fat-tailed behaviour and volatility clustering (i.e. the observation that large changes tend to be followed by large changes, of either sign, and small changes tend to be followed by small changes), which are two important features of financial time series Li et al. (2005). The work of Garcia et al. (2005) on forecasting next-day market clearing price of mainland Spain and California market showed that the GARCH model outperformed a general ARIMA model (which was studied in (Contreras et al., 2003) with the same data) when volatility and price spikes are present.

In the second group, artificial neural networks (ANNs) are the most common models for energy price/demand forecasting (Lowe and Webb, 1991, Hazarika and Lowe, 1997, Gao et al., 2000, Hippert et al., 2001, Mandal et al., 2005, da Silva et al., 2008, Jursa and Rohrig, 2008). ANNs are attractive for forecasting energy price/demand because they are non-linear and nonparametric. Beside these models, financial models, which are another approach based on economic processes for modelling electricity prices, have also been used for electricity price forecasting (Benth and Koekebakker, 2008).

Several additional procedures have been proposed to improve the prediction accuracy of forecast models. Gao et al. (2000) presented a regularisation method to avoid the overfitting problem, which normally causes large errors in unseen data. Some researchers have mentioned the use of multiple models. Guo and Luh (2004) believed that one single neural network might misrepresent parts of the input-output data mapping that could have been correctly represented by different networks. They showed how to generate a committee of neural networks for forecasting. A number of different neural networks were trained and used to predict the market clearing price separately. Then, the output was a weighted average of outputs of all the neural networks. The weighting coefficients were computed based on the current input data and the historical data. In addition, there is a large number of other papers on combining forecasts (Hoeting et al., 1999, Taylor and Bunn, 1999, de Menezes et al., 2000). Similarly, the cascaded neural networks were proposed in (Zhang et al., 2003).

Another approach for enhancing the prediction models is the use of pre-processing procedures to derive new input variables or select relevant input variables. They can also effectively construct new variables or reduce noise in the input data. Hazarika and Lowe (1997) used principal component analysis and wavelet transform (WT) to extract new inputs from the original demand time series. In the work by Amjady and Keynia (2009), the wavelet transform was combined with prediction models for electricity demand. The historical price series were decomposed using the wavelet transform into a set of series, which are called wavelet components. Each component was forecast by a single model and then the demand forecast was obtained by the inverse WT. Beside that the WT was also used as a denoising filter. In the work by Stevenson (2001), wavelet filter functions were used to de-noise the input vector before applying forecasting models.

Using Kalman filters (KF) or extended Kalman filters (EKF) to adjust the model parameters online have recently attracted the attention of forecasters. Lowe and McLachlan (1995), Nabney et al. (1996), Niranjan (1999) are among the first studies on this topic. The KF and EKF filters have been applied not only to neural network models but also to financial models. Lowe and McLachlan (1995) developed a prediction framework based on RBF models for predicting UK short-term electricity load demand. The online adjustment of bias by EKF allows the network to track the error more accurately. In the work by Niranjan (1999), the author proposed an approach to make the Black-Scholes model, which is a financial model for option pricing, into a dynamic system model. The EKF was used to re-estimate the model parameters recursively. The unobserved quantities in the Black-Scholes model (i.e. volatility of option price and the risk free interest rate) are the hidden space vector of the underlying system.

3.3 Standard forecasting models

This section presents some standard forecasting models, which are basically used as fundamentals to test the performance of the improvement techniques presented in Chapter 4, 5, and 6. As mention in Chapter 2, beside electricity demand and gas price, we have been provided with a number of external variables. These variables might be relevant to the target values and they might be helpful to predict the future values of the target time series. We would like to make the most use of these exogenous variables. Therefore, this thesis focuses on some multivariable forecasting models: MLP, RBF, LR and LR-GARCH. In addition, we present financial stochastic models for electricity forward price. The use of these financial stochastic models is motivated by the fact that these methods estimate not only the mean but also the variance of the forward contracts. Variance estimate is helpful in updating parameters of the adaptive models, which will be presented in Section 5.6 on page 107.

3.3.1 Linear regression

Model

Linear regression (LR) is a simple model where the output is a linear combination of inputs. The input vector of a LR can include both historical values of target variables and exogenous variables. This model is given by:

$$\widehat{y} = h_{lr}(x, \boldsymbol{\omega}, b) = \textstyle\sum_{i=1}^{d} \omega^{(i)} x^{(i)} + b = \boldsymbol{\omega}\mathbf{x} + b, \tag{3.1}$$

where $\widehat{y}$ represents the output of the model, $\boldsymbol{\omega} = \{\omega^{(1)}, \dots, \omega^{(d)}\}$ is the weight vector, b is the bias (or the intercept or the constant) and $\mathbf{x} = \{x^{(1)}, \dots, x^{(d)}\}$ represents the input vector.

Training a LR

In all the standard forecasting models in this chapter, we assume that the targets are corrupted by Gaussian noise with zero mean. This assumption of noise distribution is popular in the literature either because of arguments derived from the Central Limit Theorem[1] or just to simplify calculations. The conditional density of the target data y given the input $\mathbf{x}$ is given by:

$$p(y|\mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(\widehat{y}(\mathbf{x}) - y)^2}{2\sigma^2} \right\}. \tag{3.2}$$

LR parameters are inferred by maximising likelihood. In the training process, given training data $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_T, y_T)\}$, we need to estimate parameter vector $\mathbf{w} = [b, \omega^{(1)}, \ldots, \omega^{(d)}]$ subject to maximising the data likelihood $p(D|\mathbf{w})$. This is equivalent to minimising the negative log likelihood of the data (Nabney, 2002):

$$E = \sum_{t=1}^{T} (y_t - \widehat{y}(\mathbf{x}_t))^2 . \tag{3.3}$$

We also call E the sum-of-squares error. This cost function can be used for the MLP and RBF models as well. Let $\mathbf{X}$ denote the input matrix with a column of 1s added to the end, and Y the target vector. We can rewrite Equation (3.3) as follows:

$$E = (\mathbf{Y} - \mathbf{X}\mathbf{w})' (\mathbf{Y} - \mathbf{X}\mathbf{w}) .$$

This is a linear least-square problem: minimising E with respect to $\mathbf{w}$ can be solved by the pseudo-inverse[2] of $\mathbf{X}$. We set the derivative of E to zero and get:

$$\mathbf{w} = \mathbf{X}^{+}\mathbf{Y},$$

where $\mathbf{X}^{+}$ is the pseudo-inverse of $\mathbf{X}$.

[1] The Central Limit Theorem states that if S_n is the mean of n independent samples from an arbitrary distribution with a mean μ and variance σ^2, the distribution of S_n approaches a normal distribution with a mean μ and a variance σ^2/n as $n \longrightarrow \infty$.

[2] A pseudo-inverse is a matrix inverse-like object when the matrix may not be invertible. The pseudo-inverse X^{+} of X satisfies: $XX^{+}X = X$.

3.3.2 Multilayer perceptron

Model

A multilayer perceptron consists of a number of perceptrons organized in layers. Each perceptron has several inputs and one output which is a function of the inputs. It has been shown that networks with one hidden layer are capable of approximating any continuous functional mapping if the number of hidden units is large enough (Hornik et al., 1989). Therefore, only two-layer networks will be considered in this thesis.

For an MLP with two layers, d input variables $\mathbf{x} = \{x^{(1)}, \ldots, x^{(d)}\}$, M hidden units, and a single output $\widehat{y}$, the output is calculated as follows

$$a_j = \textstyle\sum_{i=1}^{d} \widetilde{\omega}_{ij} x^{(i)} + \widetilde{\omega}_{0j} \qquad j = 1, \ldots, M \tag{3.4}$$

$$\widehat{y} = \textstyle\sum_{j=1}^{M} \omega_j g(a_j) + \omega_0, \tag{3.5}$$

where $\{\widetilde{\omega}_{ij}\}$ and $\{\omega_j\}$ are the weights of the first and second layers respectively, $\{\widetilde{\omega}_{0j}\}$ and ω_0 are the bias of the first and second layers respectively, and the activation function $g(\cdot)$ is usually logistic sigmoid or tanh [3]. In this thesis, we used tanh activation.

Training an MLP model

We inferred MLP parameters by a maximum a posterior (MAP) method. We assume that the noise model for the target data follows a Gaussian distribution with zero mean and constant inverse variance β. Given training data $D = \{(x_1, y_1), (x_2, y_2), ..., (x_T, y_T)\}$, the cost function of the MLP model in the MAP method is defined by (Bishop, 2006):

$$E = \frac{\beta}{2} \sum_{t=1}^{T} (y_t - \widehat{y}(\mathbf{x}_t))^2 + \sum_{h=1}^{H} \left(-\frac{\alpha_h}{2} \sum_{\omega \in \mathcal{W}_h} \omega^2 \right), \tag{3.6}$$

where $\alpha_1, \ldots, \alpha_H$ are hyperparameters (discussed below).

The second term in Equation (3.6) is for regularisation. The equation is derived from an assumption that the weight prior $p(\boldsymbol{\omega}|\boldsymbol{\alpha})$ of the model is a Gaussian, where $\boldsymbol{\alpha}$ is called a hyperparameter. It is helpful to generalise the hyperparameter $\boldsymbol{\alpha}$ to multiple hyperparameters $\alpha_1, \ldots, \alpha_H$ corresponding to groups of weights $W_1, \ldots, W_H$. In theory, we can create groups of the weights in any way that we want. However, weights in the MLP

[3] These activation functions are defined by $tanh(x) = \left(e^x - e^{-x}\right) / \left(e^x + e^{-x}\right)$ and $sigmoid(x) = 1/\left(1 + e^{-x}\right)$.

are normally divided into four groups: first-layer weights, first-layer biases, second-layer weights, and second-layer biases. In addition, the first layer weights can be also divided into several groups: weights fanning out from a input variable are associated to a separate group. We used the latter grouping approach in our experiments because it is consistent with the automatic relevance determination (ARD) (MacKay, 1994), which will be used as one of the input selection methods in Section 3.4.4 (page 59).

There is a reason why we use the MAP instead of a maximum likelihood to train the MLP model. By using maximum likelihood, we often encounter overfitting: this is a problem where the model fits the noise in the training data rather than the underlying generator and may lead to large errors on unseen data. There are several approaches to overcome this problem, such as early stopping (Gao et al., 2000) or using a committee to combine different networks. In this thesis, we use weight decay to regularise the model by penalising large weights and imposing smoothness. The second term in Equation (3.6) of the MAP method penalises large weights.

We can use a non-linear optimisation algorithm (e.g. scaled conjugate gradient (SCG) (Møller, 1993)) to optimise E. The Bayesian evidence procedure is used to compute the optimal hyperparameters $\alpha_1, \ldots, \alpha_H$ and β (MacKay, 1992).

We used 10-fold cross-validation to select the number of hidden units of the MLP. In a k-fold cross-validation, the training set is divided into k nearly equally sized segments (or folds). We perform k iterations of training and validation. In each iteration, a single segment is used for validation and the remaining $k-1$ segments are used for training the model, so for each model there are k error values. The average of the errors is the cross-validation error of the model. This procedure is performed for the different MLP models with different numbers of basis functions. Since the cross-validated error of a model on the training set may be taken as an estimate for the error of the model on unseen data, the network structure corresponding to the smallest cross-validation error is chosen.

3.3.3 Radial basis functions

Model

The RBF is the main alternative to the MLP for non-linear modelling by neural networks. It was introduced by (Broomhead and Lowe, 1988). The outputs $\widehat{y}$ of an RBF model for an input $\mathbf{x}$ are given by:

$$\mathbf{r}_j = \left\|\mathbf{x} - \boldsymbol{\mu}_j\right\| \quad j = 0, \ldots, M, \tag{3.7}$$

$$\widehat{y} = \textstyle\sum_{j=1}^{M} \omega_j \phi_j(\mathbf{r}_j) + \omega_0, \tag{3.8}$$

where $\boldsymbol{\mu}_j$ are the cluster centres or first layer weights, and $\mathbf{r}_j$ is the distance between the input and the cluster centre $\boldsymbol{\mu}_j$. Here ϕ_j represents the basis functions and ω_j is the output-layer weight corresponding to the j^{th} basis function. In this thesis, the thin plate spline basis function was used because it was known to have better interpolation properties than the Gaussian basis function (Lowe, 1995):

$$\phi_j(\mathbf{r}_j) = \mathbf{r}_j^2 \log(\mathbf{r}_j).$$

Equation (3.8) can be rewritten in this form:

$$\widehat{y}(\mathbf{x}) = \boldsymbol{\phi}\mathbf{W}, \tag{3.9}$$

where $\boldsymbol{\phi} = [1, \phi_1, \phi_2, \ldots, \phi_M]$ is the design vector, and $\mathbf{W} = \{\omega_j\}$, $j = 0, 1, 2, \ldots, M$ is the output-layer weight vector. From this we can see that once the centres $\boldsymbol{\mu}_j$ are fixed, the RBF output is linear in parameters.

Training a RBF model

Given a training set $D = \{(\mathbf{x}_1, y_2), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_T, y_T)\}$, we need to estimate a set of parameters $\theta = \{\boldsymbol{\mu}_j, \omega_j\}$ subject to maximising the data likelihood $p(D|\theta)$. The cost function is the same as in Equation (3.3). It is possible to train RBF models by using standard non-linear optimisation algorithms in the same way as training MLP models. However, there is an alternative algorithm for training the RBF model, which is more used in practice, including two stages (Broomhead and Lowe, 1988):

1. *Optimise basis function centres* $\boldsymbol{\mu}_j$ $j = 1, \ldots, M$. We randomly choose a subset of the training data and use them as the basis function centres. This gives surprisingly successful result in practice (Nabney, 2002). Alternatively, we can compute these parameters in a more sophisticated way. Firstly, the dataset is clustered into a number of clusters, then the centres of these clusters can be used as the basis function

centres.

2. *Optimise the output weights*: when the basis function parameters are determined, the outputs of the RBF model are linear combinations of basis functions. We can extend Equation (3.9) to the whole dataset D as $\widehat{Y}(\mathbf{X}) = \Phi\mathbf{W}$, where $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ and $\widehat{Y}= \{\widehat{y}_1, \widehat{y}_2, \ldots, \widehat{y}_T\}$. Because the error function (3.3) is quadratic in the weights, its minimum can be found by using the pseudo-inverse of the design matrix $\mathbf{W} = \Phi^{+}\mathbf{Y}$, where $\mathbf{Y} = \{y_1, y_2, \ldots, y_T\}$.

The main advantage of the RBF is very fast training in comparison to MLP models because the pseudo-inverse in the RBF take significantly less time than the evidence procedure in training MLP models. We used 10-fold cross-validation to select the number of basis functions of RBF.

3.3.4 LR-GARCH

Model

In the forecast models described above, the errors are assumed to be homoscedastic (i.e. the variance of the residual is assumed to be independent of time). A generalised autoregressive conditional heteroscedastic (GARCH) can be used to model changes in the variance of the errors as a function of time. In this thesis we study an extended version of the GARCH model: a linear regression with GARCH model (LR-GARCH). In this model, the mean is modelled by a linear regression and the variance follows a GARCH. The LR-GARCH(r, m) model is given by:

$$\begin{aligned} y_t &= \widetilde{\beta} + \widehat{\beta}\mathbf{x}_t + \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{D}(0, n_t), && (3.10)\\ n_t &= \alpha_0 + \sum_{i=1}^{m} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{r} \gamma_j n_{t-j}, && (3.11) \end{aligned}$$

with constraints

$$\begin{aligned} \alpha_i, \gamma_j &> 0, && (3.12)\\ \textstyle\sum_{i=1}^{m} \alpha_i + \sum_{j=1}^{r} \gamma_j &< 1, && (3.13) \end{aligned}$$

where $\mathbf{x}_t$, y_t, and ε_t represent the input vector, target, and error of the model respectively, n_t is the variance of error ε_t, and $\beta = \{\widetilde{\beta}, \widehat{\beta}\}$ is the parameter vector of the output function.

$\alpha = \{\alpha_0, \alpha_1, ..., \alpha_m\}$ and $\gamma = \{\gamma_1, \gamma_2, ..., \gamma_r\}$, ε_t is i.i.d, with zero mean and variance n_t. ε_t can be a Gaussian or Student-t distribution. LR-GARCH is a generalisation of a linear time series model with homoschedastic disturbances in which the conditional variance n_t of the noise varies with information about errors and its variance up to time $t-1$. Term $\widetilde{\beta} + \widehat{\beta} x_t$ in Equation (3.10) is the same as the LR model. The error term ε_t, whose variance is defined by Equation (3.11), is a GARCH component. The GARCH model was first proposed in (Bollerslev, 1986) and is frequently used in financial forecasting.

The use of LR-GARCH was motivated from the fact that there are auto-correlations in the squared standardised residual of the LR model (see Figure C.1(a) in Appendix C on page 172). The GARCH component in the LR-GARCH model can capture these auto-correlations: when we fit the data with a model with the GARCH component, there is no longer auto-correlation in the squared standardised residual (see Figure C.1(b)).

Training a LR-GARCH model

Given a training dataset $D = \{(\mathbf{x}_1, y_2), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_T, y_T)\}$, the maximum likelihood method is used to estimate the parameters $\theta = \{\beta, \alpha, \gamma\}$ of the LR-GARCH model. If ε_t has a Gaussian distribution, the negative log likelihood of the LR-GARCH model is given by (we ignore the constant terms because they do not affect the optimisation procedure):

$$\begin{aligned} \mathcal{L}(\theta) &= -\log(p(D|\theta)) = \frac{1}{T}\sum_{t=1}^{T} e_t(\theta), \qquad (3.14) \\ e_t(\theta) &= \frac{1}{2}\log n_t + \frac{1}{2}\frac{\varepsilon_t^2}{n_t}. \end{aligned}$$

The cost function is non-linear; thus an iterative process is used to optimise the parameters. In this project, we used scaled conjugate gradient (SCG) (Møller, 1993). Derivatives of the negative log likelihood $\mathcal{L}(\theta)$ with respect to β and $\delta = \{\alpha, \gamma\}$ are given by:

$$\begin{aligned} \frac{\partial \mathcal{L}(\theta)}{\partial \delta} &= \frac{1}{2T}\sum_{t=1}^{T}\left[\frac{1}{n_t}\frac{\partial n_t}{\partial \delta}\left(1 - \frac{\varepsilon_t^2}{n_t}\right)\right], \qquad (3.15) \\ \frac{\partial \mathcal{L}(\theta)}{\partial \beta} &= \frac{1}{T}\sum_{t=1}^{T}\left[-\varepsilon_t \overline{\mathbf{x}}_t \frac{1}{n_t} + \frac{1}{2}\frac{1}{n_t}\frac{\partial n_t}{\partial \beta}\left(1 - \frac{\varepsilon_t^2}{n_t}\right)\right], \qquad (3.16) \end{aligned}$$

where:

$$\begin{aligned}
\frac{\partial n_t}{\partial \delta} &= z_t + \sum_{i=1}^{r} \gamma_i \frac{\partial n_{t-i}}{\partial \delta}, && (3.17)\\
\frac{\partial n_t}{\partial \beta} &= -2\sum_{j=1}^{m} \alpha_i \overline{\mathbf{x}}_{t-j}\varepsilon_{t-j} + \sum_{j=1}^{r} \gamma_j \frac{\partial n_{t-j}}{\partial \beta}, && (3.18)\\
z_t &= [1, \varepsilon_{t-1}^2, \ldots \varepsilon_{t-m}^2, n_{t-1}, \ldots, n_{t-r}], && (3.19)\\
\overline{\mathbf{x}} &= [1, \mathbf{x}]. && (3.20)
\end{aligned}$$

Constraints

Note that an LR-GARCH model has two constraints which are defined in Equations (3.12) and (3.13). Because the SCG algorithm can only optimise the cost function $\mathcal{L}(\theta)$ without taking into account these constraints, we cannot directly apply this algorithm to train a LR-GARCH model. Instead, we have to modify our approach. The first constraint can be removed by substituting $\alpha_i = \exp(\widehat{\alpha}_i)$, $\gamma_j = \exp(\widehat{\gamma}_j)$ and it is automatically satisfied. Substituting in Equation (3.14), we obtain $\mathcal{L}(\widehat{\theta})$. The cost function $\mathcal{L}(\widehat{\theta})$ needs to be optimised with respect to $\widehat{\alpha}_i, \widehat{\gamma}_j$ instead of α_i, γ_j. The derivatives of $\mathcal{L}(\widehat{\theta})$ can be computed from Equations (3.15 - 3.20) and the chain rule.

In order to satisfy constraint (3.13), we used the penalty function method (Fletcher, 1987). A penalty function is defined as follows:

$$g(\widehat{\theta}) = \textstyle\sum_{i=1}^{m} \exp(\widehat{\alpha}_i) + \sum_{j=1}^{r} \exp(\widehat{\gamma}_j) - 1 < 0,$$

where we denote $\widehat{\theta} = \{\beta, \widehat{\alpha}, \widehat{\gamma}\}$ the new parameters of LR-GARCH model which we optimise. We have to optimise the function $\mathcal{L}(\widehat{\theta})$ subject to the constraint $g(\widehat{\theta}) < 0$. A quadratic penalty function is constructed as follows:

$$P_{\mathcal{L}}(\widehat{\theta}, \lambda) = \begin{cases} \mathcal{L}(\widehat{\theta}) & \text{if } g(\widehat{\theta}) < 0 \\ \mathcal{L}(\widehat{\theta}) + \frac{\lambda}{2}\left[g(\widehat{\theta})\right]^2 & \text{if } g(\widehat{\theta}) > 0 \end{cases}$$

where λ is the penalty parameter. The value of $\widehat{\theta}$ that optimises the function $\mathcal{L}(\widehat{\theta})$ subject to constraint (3.13) is equal to the value of $\widehat{\theta}$ that optimises the function $P_{\mathcal{L}}(\widehat{\theta}, \lambda)$ when $\lambda \to \infty$. Then instead of optimising $\mathcal{L}(\widehat{\theta})$, we optimise the penalty function $P_{\mathcal{L}}(\widehat{\theta}, \lambda)$ with $\lambda \to \infty$. Choosing a large value of λ from the start, however, might not be effective.

Nabney (2002; page 72) pointed out that if we choose a large λ from the start, the condition number of the Hessian at the constrained optimum solution might become very high, and hence it might be difficult for algorithms like the SCG used in the thesis to find the constrained optimum solution. Nabney (2002) suggested a practical and more effective way to overcome this difficulty: starting from a moderately small value of λ, we carry out several iterations of optimisation for gradually increasing λ, with each iteration starting at the optimal solution found by the previous iteration. In the thesis we adopted this method to find the constrained optimum solution more effectively.

To implement this idea, we perform multiple iterations with increasing values of λ. In each iteration, SCG is used to optimise $P_{\mathcal{L}}(\widehat{\theta}, \lambda)$ with respect to $\widehat{\theta}$ given an assigned value of λ. SCG is a local optimisation algorithm and it requires an initial vector $\widehat{\theta}$ as an input argument. At the first iteration, λ is assigned a small value, say 15, we get the first optimal vector $\widehat{\theta}_1$. At iteration k, we use $\widehat{\theta}_{k-1}$ (i.e. output of iteration $k-1$) as initial vector, λ_k is assigned a value which is larger λ_{k-1}. In this thesis, $\lambda_1, \ldots, \lambda_k$ form a geometric progression with common ratio larger than 1 (say 10), so they have exponential growth towards positive infinity. The output $\widehat{\theta}$ of final iteration optimises the function $L(\widehat{\theta})$ subject to the constraint.

3.3.5 Financial stochastic models

Model

This section is about another approach to forecasting: financial stochastic models. Unlike machine learning and time series models where the output can be any type of time series, financial stochastic models are normally designed for a specific type of time series, such as a stock price, electricity price, etc. Economic processes influence the structure and form of financial models. In addition, the input variables in a machine learning models are like black boxes while the financial stochastic models consider the meaning of each input variable and indicate certain sets of variables as their input vector. An example of a financial model is Black–Scholes, which is a well-known model for option pricing.

Benth and Koekebakker (2008) presented stochastic dynamical models of electricity forward products. There is a difference in the terminology for derivative products in their paper and this thesis. The contracts, which are called “forwards” in this thesis, are named “swaps” in (Benth and Koekebakker, 2008). They refer to contracts which are for delivering electricity over a period. In (Benth and Koekebakker, 2008), “forward” refers to contracts with a fixed delivery time. In fact, these contracts do not exist in the market because electricity is only useful for practical purpose over a period of time, but Benth and Koekebakker (2008) introduced the concept of “fixed delivery time contract” for the purpose of deriving the equations of their “swap” contracts. In order to be consistent to the other parts of the thesis, we will call the period delivery contracts (i.e. “swaps” in Benth and Koekebakker (2008)) “forward” and the other “fixed delivery forward”.

Benth and Koekebakker (2008) presented six different stochastic models for log-returns of electricity forward prices. The log-return at time t of a forward contract whose delivery period is $[\mathcal{T}_1, \mathcal{T}_2]$ is defined by:

$$r_t(\mathcal{T}_1, \mathcal{T}_2) = \ln\left(\frac{p_{t+\Delta t}(\mathcal{T}_1, \mathcal{T}_2)}{p_t(\mathcal{T}_1, \mathcal{T}_2)}\right),$$

where $p_t(\mathcal{T}_1, \mathcal{T}_2)$ and $p_{t+\Delta t}(\mathcal{T}_1, \mathcal{T}_2)$ are the forward prices at time t and $t + \Delta t$ respectively, where Δt indicates a time unit. In this thesis, we use the time unit of one day because the price data has sampled daily.

In these stochastic models, log return $r_t(\mathcal{T}_1, \mathcal{T}_2)$ is modelled as a Gaussian distributed random variable $\mathcal{N}(m_t, \nu_t)$ with mean $m_t(\mathcal{T}_1, \mathcal{T}_2)$ and variance $\nu_t(\mathcal{T}_1, \mathcal{T}_2)$ as follows:

$$m_t(\mathcal{T}_1, \mathcal{T}_2) = \int_t^{t+1} \left(\lambda \Upsilon(s, \mathcal{T}_1, \mathcal{T}_2) - \frac{1}{2}\Upsilon^2(s, \mathcal{T}_1, \mathcal{T}_2)\right) ds \tag{3.21}$$

$$\nu_t(\mathcal{T}_1, \mathcal{T}_2) = \int_t^{t+1} \Upsilon^2\left(s, \mathcal{T}_1, \mathcal{T}_2\right)\, ds, \tag{3.22}$$

where λ is a constant, and $\Upsilon(t, \mathcal{T}_1, \mathcal{T}_2)$ is the forward volatility model. These equations of the mean and variance are derived from the assumption that the natural logarithm of electricity price is a Brownian motion (Benth and Koekebakker, 2008). There are six different forward volatility models corresponding to six different financial stochastic models (Table 3.1).

Note that these financial stochastic models are specific to electricity forward contract only because they capture an important property of electricity: it cannot be stored. These financial models provide not only the mean but also the volatility of forward prices, which cannot be obtained from the MLP and RBF models.

Model	$\Upsilon(t, \mathcal{T}_1, \mathcal{T}_2)$	Parameters	Constraint
E1	a	$\theta = \{a, \lambda\}$	$a \geq 0$
E2	$a\phi(\mathcal{T}_1, \mathcal{T}_2)$	$\theta = \{a, b, \lambda\}$	$a, b \geq 0$
E3	$a(t)\phi(\mathcal{T}_1, \mathcal{T}_2)$	$\theta = \{a, b, d, q, \lambda\}$	$a, b \geq 0$
E4	$a((1-c)\phi(\mathcal{T}_1, \mathcal{T}_2) + c)$	$\theta = \{a, b, c, \lambda\}$	$a, b \geq 0, 0 \leq c \leq 1$
E5	$a(t)((1-c)\phi(\mathcal{T}_1, \mathcal{T}_2) + c)$	$\theta = \{a, b, c, d, q, \lambda\}$	$a, b \geq 0, 0 \leq c \leq 1$
E6	$a\phi(\mathcal{T}_1, \mathcal{T}_2) + c(t)$	$\theta = \{a, b, c, d, q, \lambda\}$	$a, b \geq 0, 0 \leq c$
where: $a(t) = a + d\sin(2\pi t/250) - q\cos(2\pi t/250)$ $c(t) = c + d\sin(2\pi t/250) - q\cos(2\pi t/250)$ $\phi(\mathcal{T}_1, \mathcal{T}_2) = I.e^{bt}$; $I = \frac{e^{-b\mathcal{T}_1} - e^{-b\mathcal{T}_2}}{b(\mathcal{T}_2 - \mathcal{T}_1)}$			

Table 3.1: Financial stochastic models for electricity forward contracts.

Training financial stochastic models

The parameters of these models are estimated using maximum likelihood. We solve this problem for time series of each kind of product, for example price time series of the Win-2010 product. In each product, $\mathcal{T}_1$ and $\mathcal{T}_2$ are fixed, thus we can disregard these parameters in the subsequent analysis. Denote the set of parameters of a financial model by θ. Because log-return r_t is a Gaussian distributed random variable with mean m_t and variance ν_t, it can be rewritten in form of:

$$r_t = m_t(\theta) + \sqrt{\nu_t(\theta)}\varepsilon_t, \quad \varepsilon(t) \sim \mathcal{N}(0, 1). \tag{3.23}$$

We can convert a training set of forward prices to a set of log-returns: $D = \{r_1, ..., r_T\}$. Maximising likelihood is equivalent to minimising the negative log likelihood. Because r_t has a Gaussian distribution, the negative log likelihood is given by:

$$\mathcal{L}(\theta) = -\log P(D|\theta) = T\log\sqrt{2\pi} + \frac{1}{2}\sum_{t=1}^{T}\log\nu_t + \frac{1}{2}\sum_{t=1}^{T}\frac{(r_t - m_t)^2}{\nu_t}. \tag{3.24}$$

The cost function is nonlinear; thus an iterative process is used to optimise the parameters. In this thesis, we used scaled conjugate gradient (SCG) (Møller, 1993). Derivatives of the negative log likelihood $\mathcal{L}(\theta)$ with respect to θ are computed using the formula:

$$\frac{\partial\mathcal{L}(\theta)}{\partial\theta} = -\sum_{t=1}^{T}\frac{(r_t - m_t)}{v_t}\frac{\partial m_t}{\partial\theta} + \frac{1}{2}\sum_{t=1}^{T}\left[\frac{1}{v_t} - \frac{(r_t - m_t)^2}{v_t^2}\right]\frac{\partial v_t}{\partial\theta}.$$

Details of the equations for m_t, v_t , $\partial m_t/\partial\theta$, and $\partial v_t/\partial\theta$ for these models are given in Appendix A.

Constraints

The parameters in the financial models have several constraints as shown in Table 3.1. Because SCG optimises the cost function $\mathcal{L}(\theta)$ without taking into account these constraints, we cannot directly apply this algorithm to train these models. Instead, we have to use an alternative technique: substitution. In model E1-E5, we substitute parameters a, b, c by $\widehat{a}, \widehat{b}, \widehat{c}$, which are given by

$$a = e^{\widehat{a}},\ b = e^{\widehat{b}},\ c = \frac{e^{\widehat{c}}}{1 + e^{\widehat{c}}}\ .$$

We substitute parameters a, b, c in model E6 by $\widehat{a}, \widehat{b}, \widehat{c}$ defined by

$$a = e^{\widehat{a}},\ b = e^{\widehat{b}},\ c = e^{\widehat{c}}.$$

After these substitutions, the constraints are automatically satisfied by unconstrained variables $\widehat{a}$, $\widehat{b}$, $\widehat{c}$. Replacing these substituting equations to Equation (3.24), we obtain $\mathcal{L}(\widehat{\theta})$. The cost function $\mathcal{L}(\widehat{\theta})$ is now optimised with respect to $\widehat{a}, \widehat{b}, \widehat{c}, \lambda$ instead of a, b, c, λ. A list of the unconstrained parameters for financial models is shown in Table 3.2.

Partial derivatives of the negative log likelihood $\mathcal{L}(\widehat{\theta})$ with respect to the new parameters $\widehat{a}$, $\widehat{b}$, and $\widehat{c}$ can be computed from the partial derivatives of $\mathcal{L}(\theta)$ with respect to a, b, and c using the chain rule as follows:

$$\begin{aligned}
\frac{\partial \mathcal{L}(\widehat{\theta})}{\partial \widehat{a}} &= a\frac{\partial \mathcal{L}(\theta)}{\partial a}, \\
\frac{\partial \mathcal{L}(\widehat{\theta})}{\partial \widehat{b}} &= b\frac{\partial \mathcal{L}(\theta)}{\partial b}, \\
\frac{\partial \mathcal{L}(\widehat{\theta})}{\partial \widehat{c}} &= \left(c - c^2\right)\frac{\partial \mathcal{L}(\theta)}{\partial c} \text{ for models E1-E5,} \\
\frac{\partial \mathcal{L}(\widehat{\theta})}{\partial \widehat{c}} &= c\frac{\partial \mathcal{L}(\theta)}{\partial c} \text{ for model E6.}
\end{aligned}$$

3.4 Variable selection and pre-processing

Beside electricity demand and monthly forward gas prices, we were provided with a large number of exogenous variables which were potential candidates for inputs. However, only some of them are relevant. Using irrelevant variables will often reduce the performance of

Model	Unconstrained parameters
E1	$\widehat{\theta} = \{\widehat{a}, \lambda\}$
E2	$\widehat{\theta} = \{\widehat{a}, \widehat{b}, \lambda\}$
E3	$\widehat{\theta} = \{\widehat{a}, \widehat{b}, d, f, \lambda\}$
E4	$\widehat{\theta} = \{\widehat{a}, \widehat{b}, \widehat{c}, \lambda\}$
E5	$\widehat{\theta} = \{\widehat{a}, \widehat{b}, \widehat{c}, d, q, \lambda\}$
E6	$\widehat{\theta} = \{\widehat{a}, \widehat{b}, \widehat{c}, d, q, \lambda\}$

Table 3.2: Unconstrained parameters of financial stochastic models.

the forecasting models (da Silva et al., 2008). Therefore, this step is very important.

The potential inputs include electricity supply from Great Britain, real average temperature, wind speed, sunset time (giving seasonal information), SMP sell/buy of gas, gas demand, price of monthly/seasonal/annual base load/peak load electricity forward products, price of monthly/seasonal gas forward product, price of weekday ahead/weekend ahead gas product, SAP of gas, exchange rate GBP:USD, oil spot price, and day pattern (e.g.. day of the week). We do not have weather forecast data.

In the training phase, various measures were used to select the relevant input variables, including the correlation matrix (CM), autocorrelation function (ACF), partial autocorrelation function (PACF). These methods were used to select input variables for linear models (i.e. LR-GARCH and LR). We computed the CM of the target and exogenous variables: the exogenous variables which were highly correlated to the targets were chosen. We also computed the ACF and PACF of target time series. Lags with high correlations were selected as input variables. The number of inputs were selected by cross-validation.

Although these methods are simple and effective for selecting input variables for linear models, Drezga and Rahman (1998) reported that input variable selection procedures based on (linear) correlation analysis are not appropriate for non-linear models like MLP or RBF. To overcome this problem, some pre-processing procedures has been proposed for non-linear models (Nabney, 2002, da Silva et al., 2008, Ferreira and da Silva, 2007). In this thesis we used automatic relevance determination (ARD) for MLP and RBF models.

3.4.1 Correlation matrix

The correlation coefficient ρ_{xy} between two time series $\mathbf{x} = \{x_1, x_2, \ldots, x_T\}$ and $\mathbf{y} = \{y_1, y_2, \ldots, y_T\}$ is defined as

$$\rho_{xy} = \frac{Cov(\mathbf{x}, \mathbf{y})}{\sqrt{Var(\mathbf{x})Var(\mathbf{y})}},$$

where:

$$\begin{aligned} Cov(\mathbf{x}, \mathbf{y}) &= \frac{1}{T} \sum_{t=1}^{T} (x_t - \overline{x})(y_t - \overline{y}), \\ Var(\mathbf{x}) &= \sqrt{Cov(\mathbf{x}, \mathbf{x})}, \\ Var(\mathbf{y}) &= \sqrt{Cov(\mathbf{y}, \mathbf{y})}. \end{aligned}$$

The correlation matrix contains the correlation coefficients of variables. It shows the strength and the direction of possible linear relationships between random variables. Correlations have the properties $-1 \leq \rho_{xy} \leq 1$ and $\rho_{xy} = \rho_{yx}$. The larger the absolute value of ρ_{xy}, the higher correlation between the two variables $\mathbf{x}$ and $\mathbf{y}$. If $\rho_{xy} = 0$, the two variables are uncorrelated.

In this thesis, we compute the correlation matrix of the output (i.e. the quantity to be forecasted) and potential input variables. If a variable is highly correlated with the output, it should be chosen to be an input variable of the linear models. Selecting variables based on the magnitude of their correlation with target variable is based only on linear relationships.

3.4.2 Auto-correlation function

Let $\{x_1, x_2, \ldots, x_T\}$ be a time series. The lag-k auto-correlation of the time series is defined as (Bowerman and O'Connell, 1987):

$$\rho_k = \frac{\sum_{t=1}^{T-k} (x_t - \overline{x})(x_{t+k} - \overline{x})}{\sum_{t=1}^{T} (x_t - \overline{x})^2},$$

where

$$\overline{x} = \frac{1}{T} \sum_{t=1}^{T} x_t. \tag{3.25}$$

It is proved that $\rho_0 = 1$, and $-1 \leq \rho_k \leq 1$ for all k. The auto-correlation of a time series shows how well this time series matches a time-shifted version of itself. The graph of the auto-correlation at lags $k = 1, 2, ...$ is called auto-correlation function.

3.4.3 Partial auto-correlation function

The auto-correlation ρ_k measures the correlation between x_t and x_{t-k} regardless of their relationship with the intermediate variables $x_{t-1}, \ldots, x_{t-k+1}$. However, when deciding whether to add a lag to AR model, we should discount the effect of intermediate variables. This means that a further lagged variable x_{t-k} is only included in the model for predicting x_t if x_{t-k} is highly correlated with x_t and this correlation takes into account the intermediate variables $x_{t-1}, \ldots, x_{t-k+1}$. The partial auto-correlation function (PACF) is defined to measure such a relationship.

The lag-k partial auto-correlation of $\{x_1, x_2, \ldots, x_T\}$ and is defined as (Bowerman and O'Connell, 1987):

$$\rho_{1,1} = \rho_1, \tag{3.26}$$

$$\rho_{k,k} = \frac{\rho_k - \sum_{j=1}^{k-1} \rho_{k-1,j}\rho_{k-1}}{1 - \sum_{j=1}^{k-1} \rho_{k-1,j}\rho_j} \qquad k = 2, 3 \ldots, \tag{3.27}$$

where

$$\rho_{kj} = \rho_{k-1,j} - \rho_{k,k}\rho_{k-j,k-j} \quad \text{for } j = 1, 2, \ldots k-1. \tag{3.28}$$

The graph of the partial auto-correlation at lags $k = 1, 2, ...$ is called partial auto-correlation function. Similar to the auto-correlation, the partial auto-correlation varies between -1 and $+1$, with values near ± 1 indicating strong correlation.

3.4.4 Automatic relevance determination

Automatic relevance determination (ARD) (MacKay, 1994) is a Bayesian technique to evaluate the importance of each input variable for non-linear models. This technique is based on an assumption that the prior distributions of the parameters corresponding to the inputs are zero-mean Gaussian. A separate hyperparameter α_i is associated with each input. This hyperparameter is the inverse variance of the prior distribution of the weights fanning out from that input. The evidence procedure (MacKay, 1994) is used to optimise values of the hyperparameters. Note that the input variables are normalised before applying ARD. If a hyperparameter is small, it is likely that its associated input variable will have a large value. This means the corresponding input is important and should be included in the models. Conversely, if a hyperparameter is large, the corresponding input is not important; therefore we can omit it.

In this project, ARD is used to determined the most important variables for non-linear models (i.e. MLP and RBF) from the set of variables which are intuitively related to the target. All these potential variables were used as inputs to a non-linear model - the MLP. An iterative procedure is used: (1) an optimisation algorithm is used to optimise the parameters of the network given hyperparameters of the networks, (2) the evidence procedure is used to optimise the hyperparameters given a fixed set of weights. These two steps are repeated until convergence. Finally, we obtain the optimal values of the hyperparameters and the weights. The variables corresponding to the smallest hyperparameters are the most important inputs of the models, and therefore they are selected as inputs for the prediction models.

3.5 Model evaluations

3.5.1 Benchmark models

Because electricity demand is strongly seasonal with a period of one week, the benchmark model for this dataset is a model in which demand of a day is assumed to be the same as the demand of the same day in the previous week. Note that because of economic activity, the demand significantly drops on the public holidays and some days around them, we have to smooth the electricity demand data before applying the benchmark model.

A random walk (RW) model is used as a benchmark to evaluate the performance of forecasting monthly gas forward price. A RW is given by: $y_{t+1} = y_t + \varepsilon_{t+1}$, where ε_{t+1} is zero-mean noise. The model predicts that tomorrow's price on average will be equal to today's price on average.

3.5.2 Errors

Three types of prediction errors of the test sets were computed. They are the mean absolute percentage error (MAPE), mean absolute error (MAE), normalised mean absolute error (NMAE), root mean squared error (RMSE), and normalised root mean squared error

(NRMSE) which are defined by

$$
\begin{aligned}
e_{MAPE} &= \frac{1}{T}\sum\nolimits_{t=1}^{T}\left|\frac{y_t-\widehat{y}_t}{y_t}\right|\times 100\%,\\
e_{MAE} &= \frac{1}{T}\sum\nolimits_{t=1}^{T}\left|y_t-\widehat{y}_t\right|,\\
e_{NMAE} &= \frac{\sum_{t=1}^{T}\left|y_t-\widehat{y}_t\right|}{\sum_{t=1}^{T}\left|y_t-E[y]\right|},\\
e_{RMSE} &= \sqrt{\frac{1}{T}\sum\nolimits_{t=1}^{T}\left(y_t-\widehat{y}_t\right)^2},\\
e_{NRMSE} &= \sqrt{\frac{1}{T}\frac{\sum_{t=1}^{T}\left(y_t-\widehat{y}_t\right)^2}{Var(y)}},
\end{aligned}
$$

where y is the real demand/price, $\widehat{y}$ is the forecast demand/price, $E[y]$ is the mean of y, $Var(y)$ is the variance of y, and T is the number of observations in the test set. Note that MAE and RMSE are error measurements in absolute terms; NMAE and NRMSE are normalised versions of MAE and RMSE which are scale-free.

We also computed the improvement ratios (IR) of errors of a method compared with corresponding errors of the benchmark model (BM). For example, the IR of RMSE of a model M comparing with RMSE of the BM is given by:

$$IR_{RMSE}(M)=\frac{e_{RMSE}(BM)-e_{RMSE}(M)}{e_{RMSE}(BM)}\times 100\%.$$

Because the benchmark models basically show how predictable a dataset is, IR is used to evaluate performance of proposed models without being biased by data behaviour. It represents a data-free error measure of forecasting models. IR shows how good a method is compared to the benchmark model.

3.6 Experimental results

This section presents the preliminary results of the above pre-processing procedures and standard forecasting models: MLP, RBF, LR and LR-GARCH. The remaining of the thesis presents different approaches to improve the accuracy achieved in this section.

3.6.1 Pre-processing procedures on the electricity demand dataset

Because *day of week* is a periodic variable and the electricity demand has a period of a week, we represented day of the week by two dummy variables: $swd = \sin(2\pi i/7)$

and $cwd = \cos(2\pi i/7)$, where $i = 1$ to 7 correspond to Monday to Sunday respectively. Moreover, as can be seen in Section 2.5.1 on page 36, there are two and a half years of electricity demand data and there is a clear annual seasonality. Thus we represented days of year by two dummy variables: $syd = \sin(2\pi i/365)$ and $cyd = \cos(2\pi i/365)$, where $i = 1$ to 365 correspond to the first day and last day of the year. There is another approach to deal with weekly seasonality: using multi-equation models with different equations for each day of the week. This approach has been applied for hourly electricity demand in (Dordonnat et al., 2008, Soares and Medeiros, 2008). In their work, 24 models were developed, one for each hour of the day. We have implemented both approaches for the weekly pattern: however the results of first approach were better and are presented in the thesis.

Temperature is an important variable for electricity demand forecasting (Moral-Carcedo and Vict'ens-Otero, 2005). It would be best had we used the temperature forecast as in some previous works (RTE, 2005, Cancelo et al., 2008). However, these forecasts were not available during this study, and thus we used historical temperature in this thesis.

Temperature is known to have a non-linear relationship with electricity demand (Bessec and Fouquau, 2008, Henley and Peirson, 1997). Therefore, in linear models, instead of using real temperature (τ), we used a transformed value ($\widehat{\tau}$). The methodology for computing $\widehat{\tau}$ has been mentioned in several previous papers (Engle et al., 1986, Cancelo et al., 2008, Moral-Carcedo and Vict'ens-Otero, 2005). To define this transformation, we plotted a scatter plot of electricity demand versus temperature (Figure 3.2(a)). This plot shows that if we divide data into four groups: demand of working days with $\tau < 14^0C$ or $\tau \geq 14^0C$, and demand of weekends with $\tau < 14^0C$ or $\tau \geq 14^0C$, the demand within each group is approximately a linear function of temperature plus noise. For each group, we approximate the relationship between electricity demand and temperature by a linear function as follows:

$$\widehat{\tau} = \begin{cases} -18.89\tau + 1171.40 & \text{if working day and } \tau < 14^0C, \\ 2.09\tau + 862.69 & \text{if working day and } \tau \geq 14^0C, \\ -16.06\tau + 1009.86 & \text{if weekend and } \tau < 14^0C, \\ -1.09\tau + 781.45 & \text{if weekend and } \tau \geq 14^0C. \end{cases}$$

The linear approximations for temperature were estimated using least squares. Figure

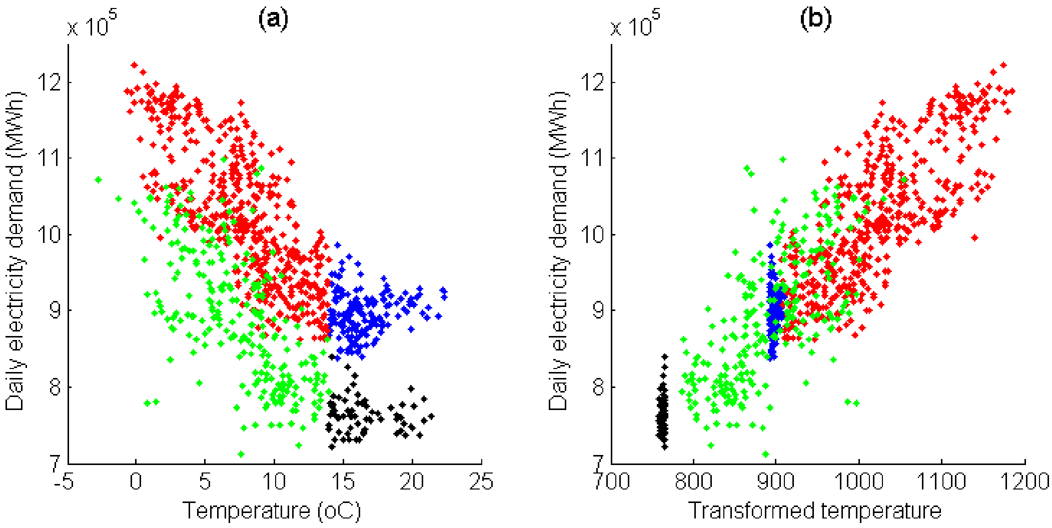


Figure 3.2: Scatter plot of electricity demand versus temperature/transformed temperature. Blue dots are data from working days with $\tau \geq 14^0C$. Red dots are data of working days with $\tau < 14^0C$. Black dots are data from weekends with $\tau \geq 14^0C$. Green dots are data of weekends with $\tau < 14^0C$.

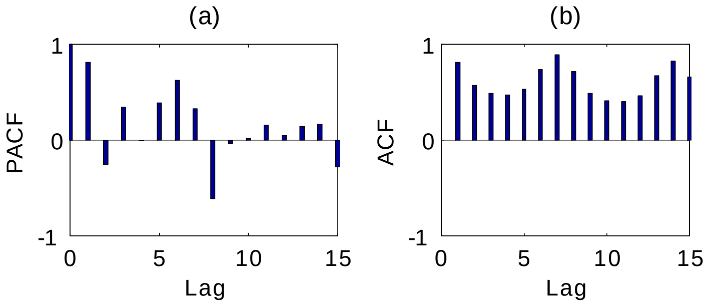


Figure 3.3: PACF and ACF of daily electricity demand time series. (a) PACF. (b) ACF.

3.2(b) confirms that electricity demand is approximately linearly related to the transformed temperature $\widehat{\tau}$.

ACF and PACF

We implemented the software to run our these experiments in Matlab: the code for PACF is based on the code written by Dr. Dan Cornford.

Figure 3.3 shows the ACF and PACF of electricity demand time series. Based on this figure, we chose lags 1, 6, 7, and 8 for linear models. Threshold is chosen by cross-validation.

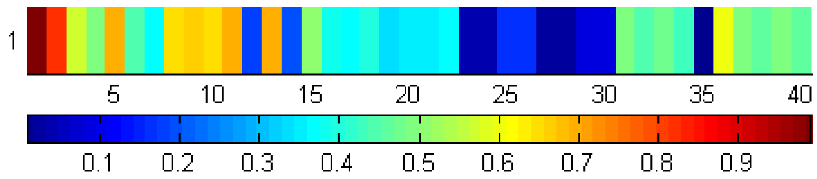


Figure 3.4: Absolute correlation of the electricity demand and exogenous variables.

Correlation matrix

We computed the correlation matrix ρ of electricity demand and exogenous variables. Figure 3.4 shows the absolute value of the correlation matrix $|\rho|$. The indexed attributes in the correlation matrix are listed as follows:

1 Electricity demand at time step t (This is target value in electricity demand dataset).

2 Electricity supply at the time step $t-1$ (denote by s_{t-1}).

3 Electricity supply at the time step $t-2$.

4 Electricity supply at the time step $t-3$.

5 Transformed temperature at the time step $t-1$ (denote by $\widehat{\tau}_{t-1}$).

6 Transformed temperature at the time step $t-2$.

7 Transformed temperature at the time step $t-3$.

8 Average temperature at the time step $t-1$.

9 Average temperature at the time step $t-2$.

10 Average temperature at the time step $t-3$.

11 Gas demand $t-1$ (denote by g_{t-1}).

12 swd at time step t.

13 cwd at time step t (denote by cwd_t).

14 syd at time step t.

15 cyd at time step t.

16 Price of weekday ahead base load electricity product at time step $t-1$.

17 Price of weekday ahead peak load electricity product at time step $t-1$.

18 Price of weekend ahead base load electricity product at time step $t-1$.

19 Price of one-month-ahead forward product, base load at time step $t-1$.

20 Price of one-month-ahead forward product, base load at time step $t-2$.

21 Price of one-month-ahead forward product, peak load at time step $t-1$.

22 Price of one-month-ahead forward product, peak load at time step $t-2$.

23 Price of one-winter-ahead forward product, base load at time step $t-1$.

24 Price of one-winter-ahead forward product, base load at time step $t-2$.

25 Price of one-summer-ahead forward product, base load at time step $t-1$.

26 Price of one-summer-ahead forward product, base load at time step $t-2$.

27 Price of one-winter-ahead forward product, peak load at time step $t-1$.

28 Price of one-winter-ahead forward product, peak load at time step $t-2$.

29 Price of one-summer-ahead forward product, peak load at time step $t-1$.

30 Price of one-summer-ahead forward product, peak load at time step $t-2$.

31 Gas SMP buy at time step $t-1$.

32 Gas SMP buy at time step $t-2$.

33 Gas SMP sell at time step $t-1$.

34 Gas SMP sell at time step $t-2$.

35 Weather: wind speed at time step $t-1$.

36 Weather: sunset time at time step $t-1$.

37 Gas SAP at time step $t-1$.

38 Gas SAP at time step $t-2$.

39 Price of day-ahead gas forward product at time step $t-1$.

40 Price of day-ahead gas forward product at time step $t-2$.

Attributes 2, 5, 11 and 13, which are s_{t-1}, $\widehat{\tau}_{t-1}$, g_{t-1}, and cwd_t respectively, are the most highly correlated and were chosen as inputs for linear forecasting models.

ARD

We used ARD to select relevant inputs for non-linear prediction models (i.e. MLP and RBF) by estimating the corresponding hyperparameters. The target variable was daily electricity demand. Table 3.3 shows a list of these potential input variables and their corresponding hyperparameters for electricity demand forecasting.

These pre-processing procedures were used to rank the relevance of variables/attributes for each linear and non-linear model. Then we used 10-fold cross-validations to decide the numbers of input variables for forecasting models. Tables 3.4 shows the final selection of input variables used for predicting electricity demand.

Variable	Hyperparameter	Variable	Hyperparameter
cwd	7.0	Daily electricity demand at t-8	559.9
swd	14.8	Gas demand at t-1	869.1
Daily electricity demand at t-3	22.9	Wind speed at t-1	1402.3
Price of electricity baseload 1-summer-ahead at t-1	26.4	Price of electricity working day ahead baseload at t-1	3259.7
Daily electricity supply at t-3	32.0	Gas SAP at t-2	13421.0
Daily electricity demand at t-1	34.5	Price of electricity working day ahead peakload at t-1	1832400
Transformed temperature at t-1	35.0	Transformed temperature at t-2	1910500
Price of electricity peakload 1-summer-ahead at t-1	36.4	Sunset time at t-1	7972200
Daily electricity demand at t-9	37.1	Gas day ahead at t-2	12395000
Daily electricity demand at t-7	37.2	Price of electricity peakload 1-winter-ahead at t-2	22358000
Daily electricity demand at t-5	48.7	Price of electricity baseload 1-winter-ahead at t-1	29745000
Daily electricity supply at t-1	49.1	Gas SMP buy at t-1	65024000
Gas SMP buy at t-2	50.8	Daily electricity demand at t-6	76232000
Price of electricity baseload 1-summer-ahead at t-2	85.1	Price of electricity monthly baseload at t-1	130580000
cyd	90.9	Temperature at t-3	206130000
Temperature at t-1	100.0	Price of electricity monthly baseload at t-2	346600000
Gas SMP sell at t-2	117.7	Gas day ahead at t-1	652920000
Price of electricity peakload 1-summer-ahead at t-2	119.9	Price of electricity monthly peakload at t-1	752050000
Daily electricity demand at t-4	124.8	Price of electricity monthly peakload at t-2	1961300000
Temperature at t-2	143.3	Gas SMP sell at t-1	2914700000
Daily electricity supply at t-2	190.6	Gas SAP at t-1	4785700000
Daily electricity demand at t-2	201.2	Price of electricity peakload 1-winter-ahead at t-1	11996000000
Price of electricity weekend ahead at t-1	293.0	Price of electricity baseload 1-winter-ahead at t-2	79748000000
syd	402.5	Transformed temperature at t-3	532880000000

Table 3.3: Hyperparameters associated with the potential input variables for forecasting daily electricity demand.

Dataset	Methodologies	Target	Input variables
Daily electricity demand	MLP,RBF	d_t	$d_{t-1}, d_{t-3}, d_{t-5}, d_{t-7}, d_{t-9},$ $cwd_t, swd_t, \widehat{\tau}_{t-1}, s_{t-1}, s_{t-3}, p^{bs}_{t-1}, p^{ps}_{t-1}$
	LR, LR-GARCH	d_t	$d_{t-1}, d_{t-6}, d_{t-7}, d_{t-8}, \widehat{\tau}_{t-1}, g_{t-1}, s_{t-1}, cwd_t$
where:	d : daily electricity demand s : electricity supply from Great Britain $\widehat{\tau}$: Transformed temperature g : gas demand swd, cwd : two dummy variables presenting day of week p^{bs} : Price of electricity base load one-summer-ahead product p^{ps} : Price of electricity peak load one-summer-ahead product		

Table 3.4: Input variables of prediction models for daily electricity demand.

3.6.2 Pre-processing procedures on the gas forward price dataset

We also applied these pre-processing procedures and 10-fold cross-validation to the gas forward price dataset. Because the gas forward price dataset includes a large number of sub-datasets over a long period of time and is non-stationary, after some time we should look again at the exogenous variables to see if they are still relevant to the gas price and thus help in the prediction of day ahead price. The correlation of gas price and exogenous variable can change over time, for example in the period of June 2006 to May 2007, gas price is highly correlated to the price of one-winter-ahead gas forward product, but the following year the correlation no longer holds. Table 3.5 shows the input variables for predicting gas forward prices.

3.6.3 Forecasting results on the electricity demand dataset

We used the NETLAB toolbox[4] for training the MLP, RBF, LR and ARD. The number of hidden units in the MLP models for forecasting electricity demand was 12. The number of basis functions in the RBF model is 80. These numbers were selected by 10-fold cross-validation (see Section 3.3.2, page 48). We used the MLPs with tanh activation functions.

Table 3.6 shows the errors and improvement ratios of the prediction models. All models

[4] This toolbox is available at http://www1.aston.ac.uk/eas/research/groups/ncrg/resources/netlab/.

Product of test set	**Methodologies**	**Target**	**Input variables**
Jun06 - May07 (first 12 sub-datasets)	MLP, RBF, LR, LR-GARCH	p_t	$p_{t-1}, p_{t-2}, p^w_{t-1}, p^w_{t-2}$
Jun07-May08 (last 12 sub-datasets)	MLP, RBF, LR, LR-GARCH	p_t	$p_{t-1}, p_{t-2}, p^s_{t-1}, p^s_{t-2}$
where:	p : price of monthly gas forward product p^w : price of one-winter-ahead gas forward product p^s : price of one-summer-ahead gas forward product		

Table 3.5: Input variables of prediction models for gas forward price.

Models	**IR(RMSE)**	**RMSE**	**NRMSE**	**MAPE**	**MAE**	**NMAE**
Benchmark	0.00%	39365	0.36550	2.96%	29011	0.32877
LR-GARCH	45.72%	21369	0.19841	1.72%	16538	0.18742
LR	44.49%	21850	0.20252	1.76%	16915	0.19112
MLP	**53.12%**	**18455**	**0.17135**	**1.43%**	**13940**	**0.15798**
RBF	**48.72%**	**20187**	**0.18743**	**1.63%**	**15589**	**0.17666**

Table 3.6: Errors and improvement ratio of NMSE of forecast methods for the electricity demand dataset.

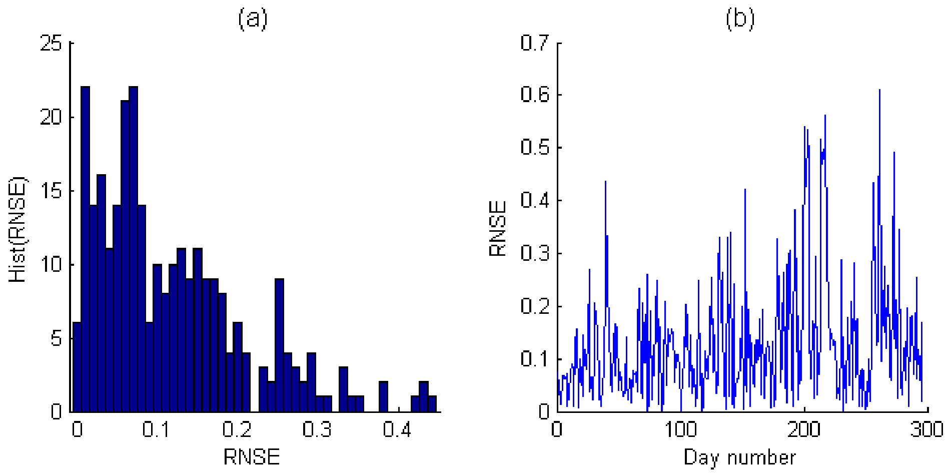


Figure 3.5: Root normalised squared error (RNSE) of the MLP model for forecasting electricity demand. (a). Histogram of RNSE. (b) Values of RNSE over time.

worked very well on this dataset. Non-linear models (i.e. RBF and MLP) provided better prediction than linear models (i.e. LR and LR-GARCH). The MLP model was the best with the NRMSE of 0.17135 which improve 53.12% comparing with the benchmark model. Figure 3.5 plots the histogram and values over time of the root normalised squared error (RNSE) for the prediction of the MLP model, where the RNSE is a time series whose each element at each time step t is given by:

$$e_{RNSE,t}=\sqrt{\frac{(y_t-\widehat{y}_t)^2}{Var(y)}},$$

where y is the real demand, $\widehat{y}$ is the forecast demand, and $Var(y)$ is the variance of y. $P(RNSE < 0.2) = 81\%$ and $P(\text{NRSE}< 0.4) = 95\%$. This means that there were only a few data points which have large RNSE: 95% data points have RNSE falling into the range $[0, 0.4]$.

3.6.4 Forecasting results on the gas forward price dataset

The number of hidden units in the MLP models for forecasting gas price was 8. We also used the MLPs with $tanh$ activation functions. The number of basis functions in the RBF model is 30. These numbers were selected by 10-fold cross-validation.

The gas forward price dataset consists of 24 sub-datasets. The IR_{RMSE}, RMSE, NRMSE, MAPE, MAE, and NMAE were computed for each sub-dataset and for each prediction method. Their averaged values for 24 sub-datasets are shown in Table 3.7.

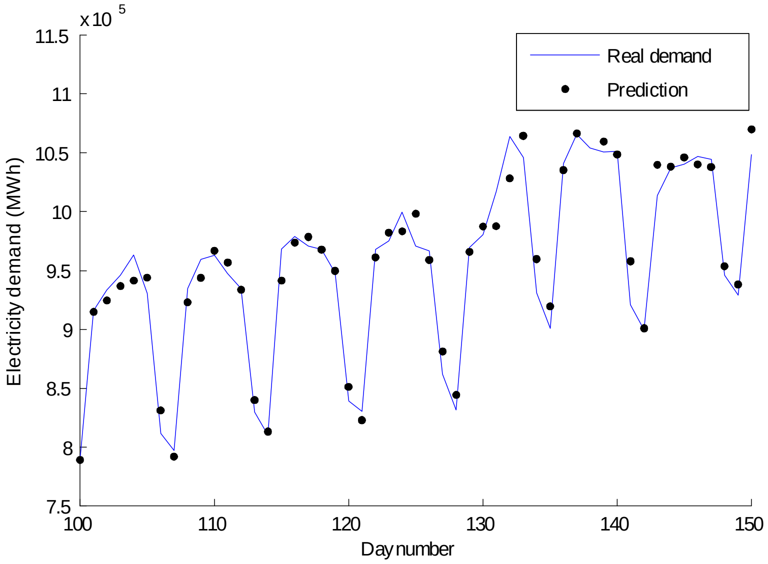


Figure 3.6: The real demand and prediction of the MLP model from 30/09/2006 to 19/11/2006.

Models	IR(RMSE)	RMSE	NRMSE	MAPE	MAE	NMAE
Benchmark	0.00%	1.11862	0.48980	2.31%	0.84562	0.45182
LR	3.17%	1.08295	0.47735	2.26%	0.83577	0.44283
LR-GARCH	**3.77%**	**1.07378**	**0.47310**	**2.26%**	**0.83562**	**0.44281**
MLP	2.97%	1.09047	0.47941	2.27%	0.83586	0.44599
RBF	2.15%	1.09969	0.48346	2.28%	0.84292	0.44976

Table 3.7: Errors and Improvement Ratio of RMSE of forecast methods for the gas forward price dataset.

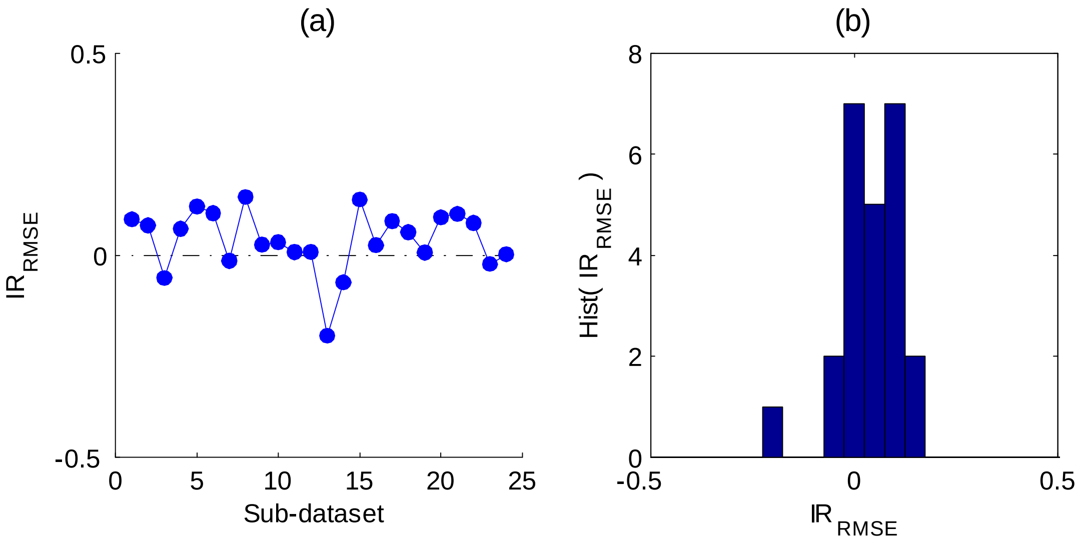


Figure 3.7: IR_{RMSE} of LR-GARCH models for forecasting gas forward price. (a) Values of IR_{RMSE} over sub-dataset. (b) Histogram of IR_{RMSE}.

Table 3.7 shows that all these prediction models generally provided a poor quality of prediction on this dataset. Because the IR$_{RMSE}$ of all forecasting model were near to zero, these models did not improve much compared to the random walk model. This means that they did not beat the simple random walk model.

The volatility of the gas price dataset is higher than that of the electricity demand dataset. There is no periodicity as in the demand dataset. In addition, the price data is non-stationary: for example, the trend of price suddenly changes in the period around 1^{st} February 2007 (see Figure 2.9 on page 38). These characteristics of the price dataset make the prediction task much more difficult.

Figure 3.7 shows the histogram of IR_{RMSE} of 24 sub-datasets using LR-GARCH models. The biggest forecasting error occurred in the sub-dataset 13 with IR_{RMSE} of -19%. This happened because of the irregular characteristic of gas forward price around 1^{st} February 2007. In this period, the market experienced a problem in which the trend of the forward price suddenly changed from downward to stable (see Figure 2.9). The test set starts around February 2007, the training set of course had to be chosen from observations before that critical time. Because of this change, the trends of the training set and test set are too different: the training set tends to decrease while the test set does not. Thus the models whose parameters were inferred from the training sets no longer capture the correct trends of the test set. This is the reason why the prediction models did not work on this sub-dataset.

Models	IR(RMSE)	RMSE	NRMSE	MAPE	MAE	NMAE
RW	0.00%	1.5554	0.52192	2.050%	0.94011	0.39467
E1	**0.51%**	**1.5475**	**0.51927**	**2.040%**	**0.93138**	**0.39101**
E2	-0.17%	1.5580	0.52280	2.080%	0.94694	0.39754
E3	-1.80%	1.5833	0.53129	2.087%	0.95183	0.39959
E4	-4.55%	1.6262	0.54568	2.197%	1.02126	0.42874
E6	**0.18%**	**1.5525**	**0.52097**	**2.052%**	**0.93916**	**0.39427**

Table 3.8: Errors of the financial stochastic models for the electricity forward price dataset.

3.6.5 Forecasting results on the electricity forward price dataset

As mentioned in Section 3.3.5 on page 53, the financial stochastic models are designed for electricity forward prices only, but not for other commodity prices or any general time series because they capture an important property of electricity: it cannot be stored. Therefore, in this section we introduce a new dataset of electricity prices in order to evaluate performance of these financial stochastic models. This dataset contains daily prices of monthly base load electricity forward products from Aug-2005 to May-2007 and is sampled daily from 4^{th} April 2005 to 27^{th} April 2007. Because it is possible to trade electricity from one to four months ahead, there are four months of daily price data (approximately 85-95 data points) for each monthly electricity product. We created 22 sub-datasets: each sub-dataset corresponds to the price time series of a single product, in which the first two thirds of the time series is used as the training set and the remaining is used as the test set.

Note that these financial stochastic models require a certain set of variables as their input vector (including time step, start and end time of the delivery period), but are not a black box as in the machine learning models. Because of this reason, we do not need to perform the input variable selection step.

Table 3.8 shows results of these models on the dataset of the base load electricity monthly forward price. The table shows that all these financial stochastic models did not work on this dataset: IR_{NRMSE} of all financial stochastic models are around 0. We actually tested these fixed models on other electricity forward prices: base load/peak load quarterly/seasonal forward price, but the results are the same or even slightly worse than those shown in these tables.

One of the reasons that these models do not work is that we used a local minimisation algorithm, the scaled conjugate gradient (SCG), to optimise the models parameters. The SCG is a local search only and it is not strong enough for training these financial models. Because the objective functions in training these models are complicated and some of them include *sine* and *cosine* functions in their equations (see Appendix A on page 162), the search landscape is very complicated and there are many local minima. SCG cannot work well for problems with this structure as it can easily get stuck in a local minimum. This problem was shown in our experiments: every time we initiated with a different value of parameters, we got a different solution. This means the parameters found are not optimal globally, but just optimal locally. In future research, the results of these methods can be improved by using global optimisation algorithms for training the models (see Section 8.2 on page 152).

3.7 Summary

This chapter presented an overview of approaches to solving prediction problems. We reviewed a number of standard prediction models, pre-processing procedures, and performance measures. They were applied to two datasets: the first dataset is electricity demand, which is stationary, and the second dataset is gas forward price, which is noisy and non-stationary. For the first dataset, the standard prediction models provide very good predictions, of which the non-linear models are better than the linear models and the best performer is the MLP with IR_{RMSE} of 53.12% compared with the benchmark.

However, for the second dataset, these standard prediction models show some deficiencies in predicting the gas price. The accuracy of these prediction models on this dataset is almost the same as the benchmark. Similarly, the performance of the financial stochastic models on the electricity price dataset are the same as the random walk model. These shortcomings prompted us to investigate different factors of data and models to improve prediction performance. In terms of data, we will use the wavelet transform to decompose data into several components before prediction. In terms of models, we will investigate updating model parameters on the test set and the form of the noise distribution. These techniques do help the standard forecasting models to achieve better performance. They will be presented in Chapters 4, 5, and 6 respectively.

4 Pre-processing with the wavelet transform

The wavelet transform (WT) is one of the techniques for improving forecasting performance that we investigate in this thesis. It is a pre-processing procedure which helps us to decompose the trend and details of data. This chapter studies the question of which types of WT can be used in forecasting applications. We will also discuss different methods for using the WT in prediction and empirically compare their performances.

4.1 Introduction

To improve the accuracy of forecasting, multiresolution decomposition techniques such as the wavelet transform have been used as a pre-processing procedure. The WT can produce a good local representation of the signal in both the time and frequency domains. In this chapter, we present combinations of wavelet transform and the standard prediction models (i.e. LR-GARCH, MLP, LR and RBF). The transformation is applied to the target variable prior to modelling. We compare the prediction performance of the prediction models without WT and the performance of the following two combination methods:

- *Multicomponent-forecast method*: a WT decomposes the target variable y_t into multiple wavelet components, and then each component is forecast with a separate machine learning or time series model.
- *Direct-forecast method*: the components of the WT are used as input variables to a single forecast model to directly predict the target.

Although both methods of combining the WT with prediction models have already published, previous papers only used either the multicomponent-forecast or the direct-forecast. In this thesis, we use both types of the named methods and compare their prediction accuracy. This comparison will provide an answer to the question of which is better for energy datasets. The experimental results on the UK data will show that the multicomponent-forecast method outperforms both the direct-forecast method and the models without the WT. In addition, we will analyse the correlation of residuals of components when using the multicomponent-forecast method. The analysis results will show that the residuals of the WT components are highly correlated. This raises an open question for future study: how to use this special characteristic of these residuals to improve prediction performance.

In Section 4.2, an overview of the wavelet transform is presented. Section 4.3 explains why we choose the redundant Haar wavelet transform (RHWT) in this thesis and how the RHWT decomposes data into components. Section 4.3 defines the detailed forecasting frameworks using the WT. Numerical results and evaluation on data from the UK energy markets are given in Section 4.4.

4.2 Wavelet transform

Mathematical transforms can be used to represent time series data in different domains, such as time or frequency; so they provide us with further information that is not observable in the original data. There are a number of transformations introduced in the literature, among them the Fourier transforms (FT) are the most popular. The Fourier transform converts data from time-based to frequency-based: it shows which frequency components are presented in the data and with what strength. However, the main disadvantage of the FT is that the FT discards time information. The wavelet transforms is one technique which can overcome this shortcoming. Unlike the Fourier transform, the

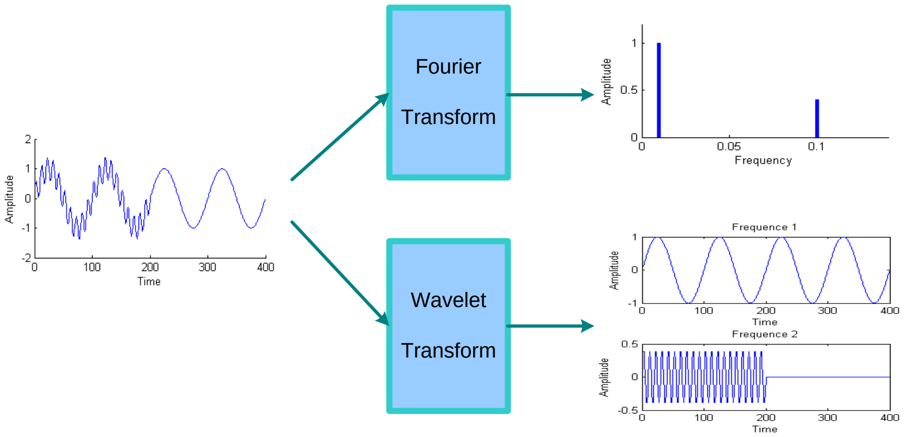


Figure 4.1: The wavelet transform versus the Fourier transform. The wavelet transform can represent data in both time and frequency domains while the Fourier transform can represent data in frequency domain only. Note that the wavelet transform here is not an actual one, but an ideal WT for the illustrative purpose.

wavelet transform represents data in both time and frequency domains: WTs show us not only what frequencies are in the data but also when each frequency occurred (see Figure 4.1). In this figure, the original data contains two frequencies: the lower frequency exists all the time (i.e. time period $[0, 400]$) and the higher frequency occurs only in the period $[0, 200]$. The Fourier transform shows these frequencies and their amplitudes but there is no information about temporal variation in frequency strength. In the wavelet transform, we obtain more information about the data. The data is represented by two components which correspond to two frequencies. Each component shows the contribution of each frequency and how it varies with time.

There are two types of WT: continuous and discrete. In the scope of this thesis, we are concerned with the discrete WT only because we have to deal with time series measured at discrete time points.

The WT analyses the data in multiple frequency bands at multiple resolutions: the lower frequencies have better resolution in frequency and the higher frequencies have better resolution in time. The wavelet decomposition normally has two steps (Figure 4.2(a)):

- *Step 1*: Filtering the original data. The original time series y_t is decomposed into different frequency bands by passing the time series y_t through a halfband high-pass filter and low-pass filter. The WT has two functions: (1) a scaling function $l(t)$, which is associated with the low-pass filter, and (2) a wavelet function $g(t)$, which is associated with the high-pass filters. After this step we obtain an approximate component A and detailed component D which correspond to the low-frequency and high-frequency information of the data.

- *Step 2*: Subsample (or down-sample) the above components by 2. It means that we keep only one point out of two. This step reduces the number of samples in the WT. After this step we have two wavelet coefficients cA and cD whose lengths are equal to half of A and D. The main advantage of the step 2 is to reduce the storage requirement.

Filtering a time series y_t can be implemented by a convolution and can be mathematically represented as follows:

$$\begin{aligned} D_t &= \sum_{\tau=-\infty}^{+\infty} y_\tau g(t-\tau) \\ A_t &= \sum_{\tau=-\infty}^{+\infty} y_\tau l(t-\tau), \end{aligned}$$

where A and D are the outputs of the high-pass and low-pass filters respectively. The high-pass and low-pass filters are a dependent pair, and their relationship is given by

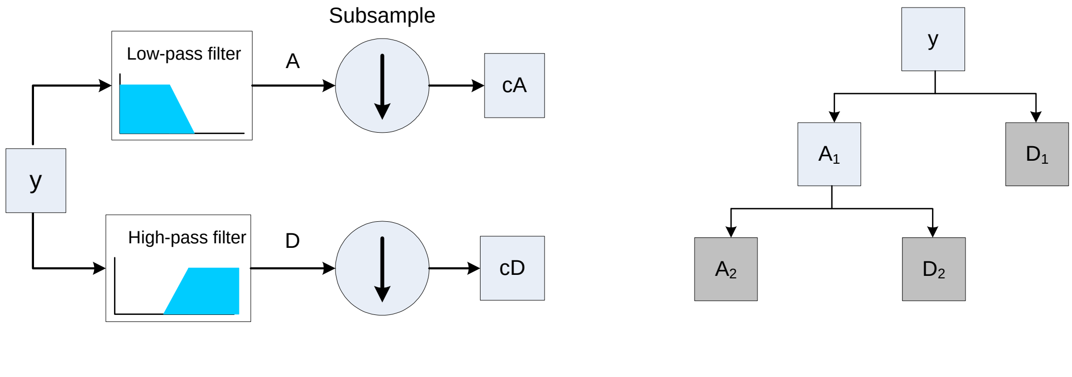
$$g(M-1-t) = (-1)^t \cdot l(t),$$

where M is the filter length.

The decomposition process can be iterated to create a multi-layer decomposition (see Figure 4.2(b)). For example, an n-level wavelet decomposition has an approximation A_n (which is the low-frequency component of the signal) and n details $D_1, D_2, ..., D_n$ (which are high-frequency components). Each component represents the data in a frequency range that is less volatile and easier to forecast than the original time series y. We also can reconstruct the original data from the WT components. However, it is not easy to get a perfect reconstruction.

A number of WT families have been introduced, such as Daubechies, Haar, Meyer, Symlet (see Figure 4.3). Each WT family corresponds to a pair of scaling and wavelet functions and a pair of reconstruction functions. Among various wavelet families, the Daubechies wavelet is the most popular.

WTs were introduced only about three decades ago, but they are very powerful and have been used in a wide range of applications, such as signal denoising, signal/image/finger-



(a) (b)

Figure 4.2: (a) Wavelet decomposition. (b) 2-level wavelet decomposition (based on Misiti et al. (2008)).

print compression, speech/image recognition, etc. In this thesis we used WTs for prediction.

4.3 Redundant Haar wavelet transform

As mentioned before, there is a range of WT families, but we used a simple one in this thesis: the redundant Haar wavelet transform (RHWT). This section shows why we chose the RHWT and how to compute the RHWT.

4.3.1 Why the RHWT?

As mentioned before, there are a number of WT families, such as Daubechies, Haar, Meyer, Symlet. Of these the symmetric WTs like Meyer are not appropriate for prediction. The first reason is that the components of the symmetric WTs take into account not only previous information but also future information (see Figure 4.4), but in forecasting problems, we can only use data obtained earlier in time.

Another reason are the difficulties with distortion at the boundary of the time series when applying the WT. In an asymmetric WT, we use only previous observations to compute components, thus at the beginning of the time series, there are not sufficiently many previous lags for computing WT components. For example, in the Haar WT, we have to use data at time t and $t-1$ to compute WT components at time t, but there is no data at time 0 to compute the WT components at time step 1. Some extensions of the basic WT procedure have been proposed to avoid this problem, such as symmetrical

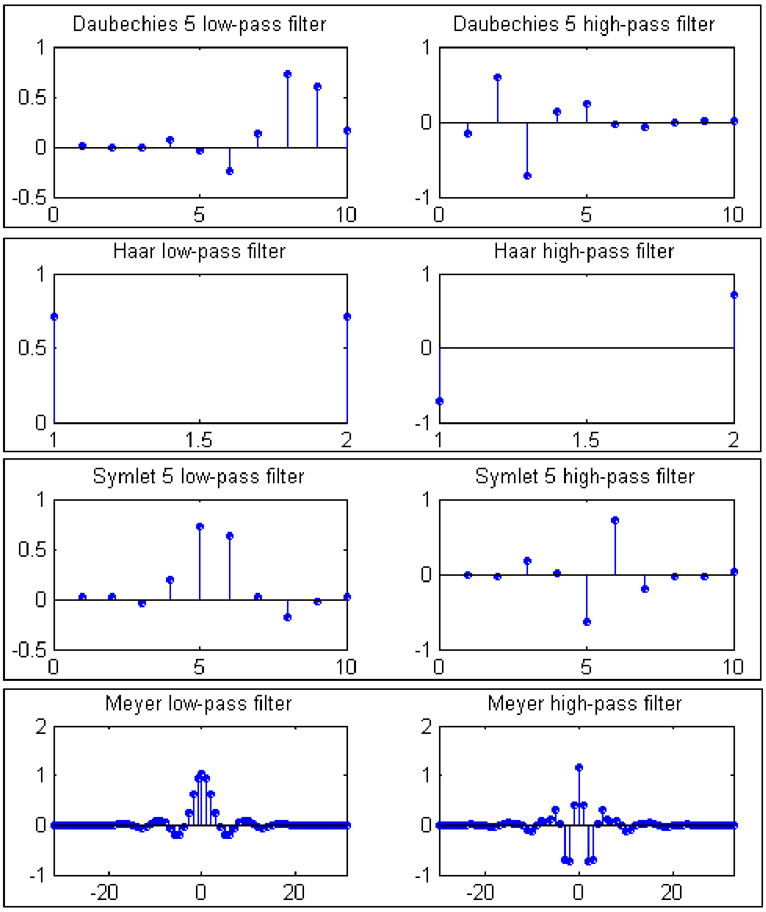


Figure 4.3: Low-pass and high-pass filters of the wavelet families.

extension or zero padding. These procedures are applied to the beginning of a time series, before computing WT components of the first observations. However, this leads to an inconsistency between some observations near the left-boundary and the remaining observations, which we shall call the left-boundary distortion problem. In a symmetric WT, both previous and future observations are used to compute WT components, thus both right-boundary and lelf-boundary distortions occur. This means that asymmetric WTs can avoid the right-boundary distortion while symmetric WTs cannot. The right-boundary distortion makes the features extracted from the last observations, which are the most important for a prediction application, normally worse than the rest of the series.

Two asymmetric WT families used most in the literature for prediction applications are the Haar (Benaouda et al., 2006, Saha et al., 2006, Renaud et al., 2005, Starck and Murtagh, 2001) and Daubechies WT (Yousefi et al., 2005, Conejo et al., 2005, Yao et al., 2000, Xu and Niimura, 2004). Although the Daubechies WT is asymmetric, Benaouda et al. (2006) reported that it is not good for these applications because the Daubechies WT is not consistent in responding to similar events in the observed time series. This means that the identical events across the time series can appear in so many different fashions in the decomposed components. Therefore, we decided to use Haar WT in our thesis, but not Daubechies.

In addition, another issue that we consider when selecting the type of WT is subsampling. A discrete WT normally has two stages: (1) filtering the data and (2) subsampling. Although subsampling reduces the storage requirement, it has the problem of shift variance, i.e. if we delete the first value of the time series, the subsampled coefficients of the WT are different from the heretofore. To overcome this, we can use a redundant or non-subsampled wavelet transform (Starck and Murtagh, 2001). In a redundant WT, only stage (1) is completed. All components of the redundant WT have the same length as the original time series. Therefore, there is a one-to-one correspondence between the original data and components at a given time step. This makes the prediction and modelling procedure more convenient. The RHWT also achieves a perfect reconstruction of the original data from WT components.

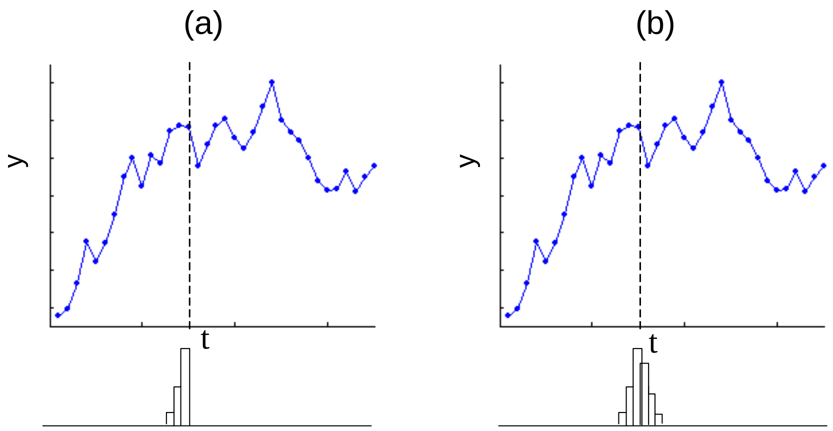


Figure 4.4: Kernel used for computing WT components in (a) asymmetric WT; (b) symmetric WT.

4.3.2 Computing the RHWT

Assuming that there is a time series y_t, $t = 1, 2, \ldots, T$, Figure 4.5 shows how to compute its RHWT components to the n-th decomposition level. At level i, the detail components D_i are retained, while the approximation components A_i are decomposed into a further level of detail D_{i+1} and approximation components A_{i+1}. The original time series can be reconstructed from the wavelet components by the inverse WT procedure. For the RHWT, the inverse WT is simply a summation of the components: $y_t = A_{n,t} + D_{n,t} + \cdots + D_{1,t}$. However, this is not the case for all kinds of WT.

Note that to calculate a component at level $i+1$ at time t ($A_{i+1,t}$ or $D_{i+1,t}$), we need to use the value of time series A_i at time step $t - 2^i$. Therefore, at level $i+1$, it is impossible to exactly compute the component before time step $2^{i+1} - 1$. After applying the RHWT, this thesis will consider only those components after time step $2^n - 1$.

We determined the number of decomposition levels by cross-validation. In both datasets, a 2-level WT (i.e. $n = 2$) was chosen and the results are reported in Section 4.5. An example of decomposing by redundant Haar wavelet transform is shown in Figure 4.6. This data is the price of a monthly forward product in the UK gas market. Comparing to the original data, the approximation component A_2 is much smoother, and the detail components D_2 and D_1 contain periodic elements. Therefore, it is expected that the WT components should be easier to forecast than the original price time series. Figure 4.7 shows a part of the electricity demand time series and its wavelet components.

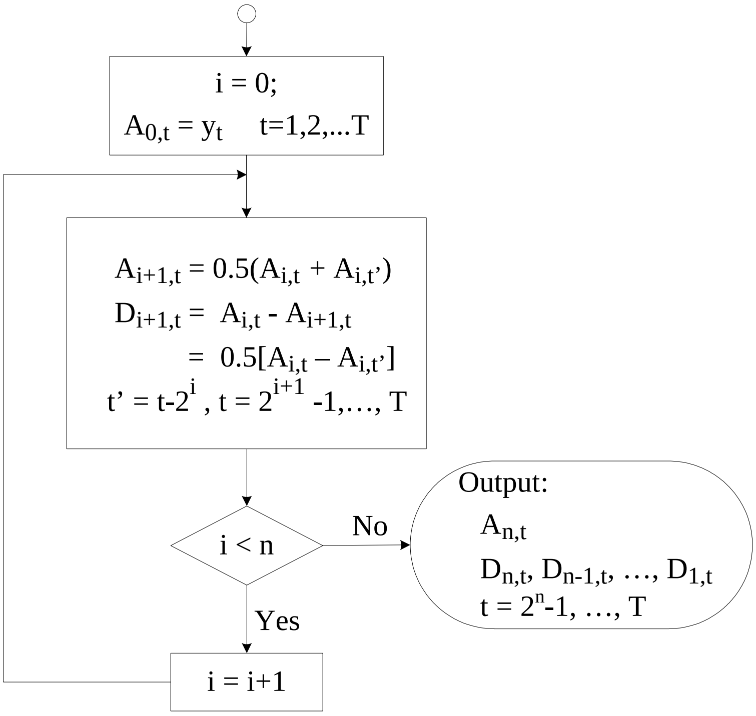


Figure 4.5: Computation of wavelet components of different scales in the RHWT.

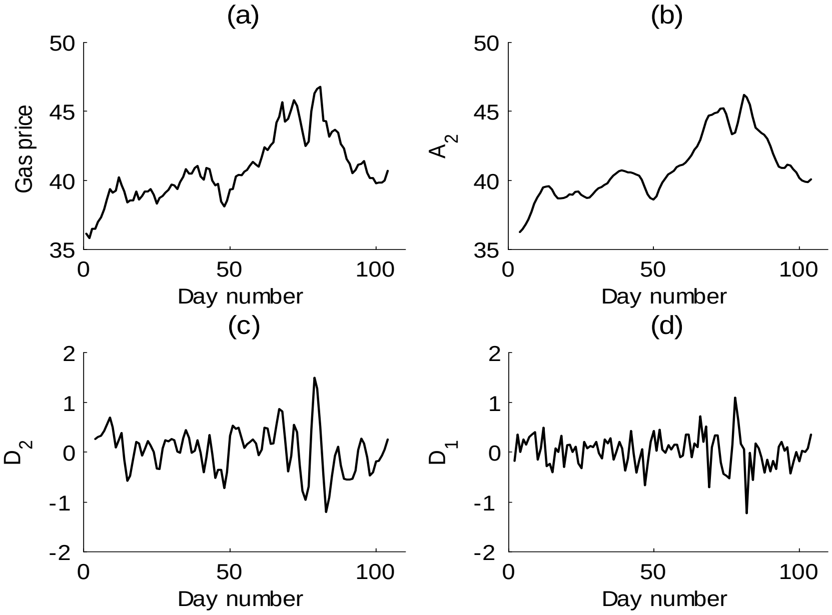


Figure 4.6: A monthly gas forward price and its RHWT components with decomposition level 2. (a) price data, (b) approximation component A_2, (c) detail component D_2, (d) D_1.

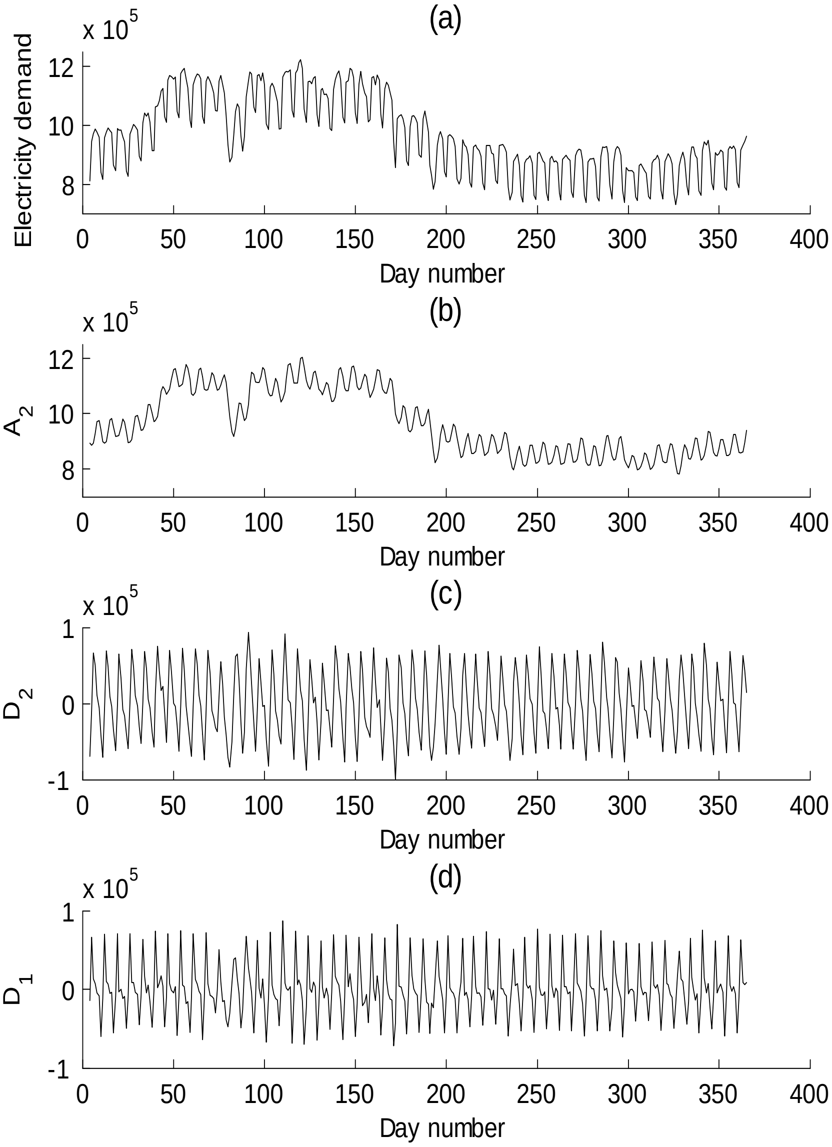


Figure 4.7: The electricity demand and its RHWT components with decomposition level 2. (a) electricity demand, (b) approximation component A_2, (c) detail component D_2, (d) D_1.

4.4 Forecasting frameworks

This section presents two methods of combining the WT with prediction models: multicomponent-forecast method (Conejo et al., 2005, Yousefi et al., 2005, Yao et al., 2000) and direct-forecast method (Benaouda et al., 2006).

4.4.1 Multicomponent-forecast method

The multicomponent-forecast method is shown in Figure 4.8. A dataset is divided into two sub-datasets: (1) a training set to estimate the model parameters and (2) a test set to evaluate performance of these models by calculating an appropriate error measure. The forecasting framework for a time series y_t consists of four steps:

Step 1: Use the RHWT to decompose y of the training set and the test set separately: $A_n, D_n, D_{n-1}, \ldots, D_1$.

Step 2: Create a distinct model for predicting each component. We determine the input vectors (including exogenous variables) for each model by pre-processing procedures (see Section 3.4 on page 56).

Step 3: In the training phase, the training sets are used to estimate parameters of the forecasting models.

Step 4: In the test phase, the developed models are used to predict the future value of the components from the current observable data. The outputs of these models at time t are the forecasts of $A_n, D_n, D_{n-1}, \ldots, D_1$ at time step $t+1$. In this thesis, the models used for forecasting are MLP/RBF/LR/LR-GARCH. The inverse WT is used to compute the forecast value of y_{t+1} from the predictions of the components.

4.4.2 Direct-forecast method

Like the multicomponent-forecast method, the target time series y_t in this method (shown in Figure 4.9) is also decomposed into WT components. These components and exogenous variables are also used as candidates for input variables. However, the main difference between the two methods is that the direct-forecast method uses a single model to predict the time series y_t directly while the multicomponent-forecast method uses several models to forecast wavelet transform components, one model for each WT component.

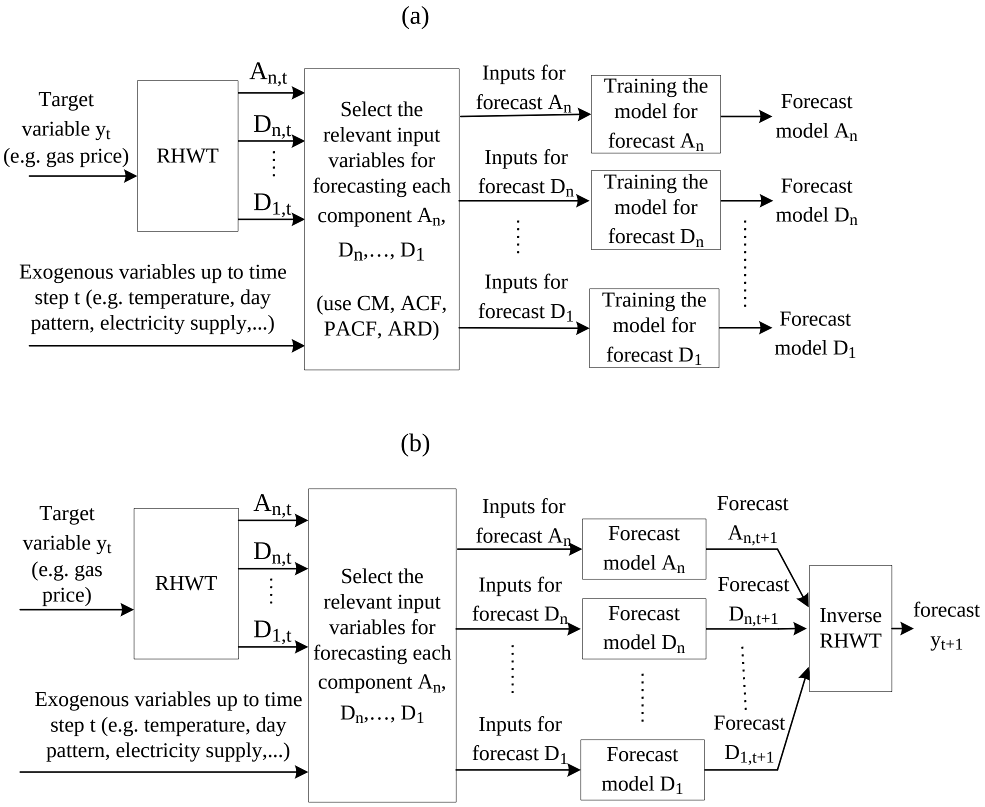

(a)
Target variable y_t (e.g. gas price)
RHWT
$A_{n,t}$
$D_{n,t}$
$D_{1,t}$
Exogenous variables up to time step t (e.g. temperature, day pattern, electricity supply,...)
Select the relevant input variables for forecasting each component A_n, D_n,..., D_1
(use CM, ACF, PACF, ARD)
Inputs for forecast A_n
Inputs for forecast D_n
Inputs for forecast D_1
Training the model for forecast A_n
Training the model for forecast D_n
Training the model for forecast D_1
Forecast model A_n
Forecast model D_n
Forecast model D_1
(b)
Target variable y_t (e.g. gas price)
RHWT
$A_{n,t}$
$D_{n,t}$
$D_{1,t}$
Exogenous variables up to time step t (e.g. temperature, day pattern, electricity supply,...)
Select the relevant input variables for forecasting each component A_n, D_n,..., D_1
Inputs for forecast A_n
Inputs for forecast D_n
Inputs for forecast D_1
Forecast model A_n
Forecast model D_n
Forecast model D_1
Forecast $A_{n,t+1}$
Forecast $D_{n,t+1}$
Forecast $D_{1,t+1}$
Inverse RHWT
forecast y_{t+1}

Figure 4.8: The multicomponent-forecast method. (a) Training phase, (b) Test phase.

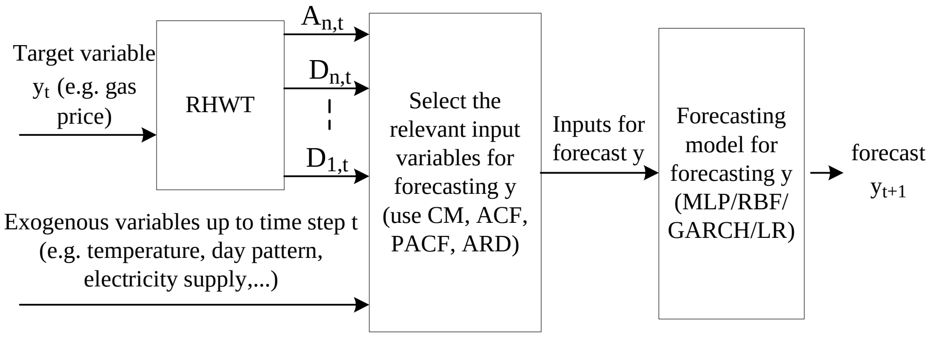

Target variable y_t (e.g. gas price)
RHWT
$A_{n,t}$
$D_{n,t}$
$D_{1,t}$
Exogenous variables up to time step t (e.g. temperature, day pattern, electricity supply,...)
Select the relevant input variables for forecasting y (use CM, ACF, PACF, ARD)
Inputs for forecast y
Forecasting model for forecasting y (MLP/RBF/ GARCH/LR)
forecast y_{t+1}

Figure 4.9: Direct-forecast method.

Methodologies	Target	Input variables
Without RHWT	d_t	$d_{t-1}, d_{t-3}, d_{t-5}, d_{t-7}, d_{t-9}, cwd_t, swd_t, \widehat{\tau}_{t-1}, s_{t-1}, s_{t-3}, p^{bs}_{t-1}, p^{ps}_{t-1}$
Multicomponent-forecast	$A_{2,t}$	$A_{2,t-1}, A_{2,t-2}, A_{2,t-3}, A_{2,t-4}, A_{2,t-5}, A_{2,t-8}, d_{t-2}, p^{pw}_{t-1}, p^{pw}_{t-2}, p^{bs}_{t-1}, p^{bs}_{t-2}, s_{t-1}$
	$D_{2,t}$	$D_{2,t-2}, D_{2,t-3}, D_{2,t-6}, D_{2,t-13}, D_{2,t-14}, D_{2,t-15}, d_{t-1}, A_{2,t-1}, s_{t-1}, cwd, swd, \widehat{\tau}_{t-1}$
	$D_{1,t}$	$D_{1,t-1}, D_{1,t-3}, D_{1,t-4}, D_{1,t-7}, D_{2,t-1}, d_{t-1}, s_{t-1}, cwd_t, swd_t, \widehat{\tau}_{t-1}$
Direct-forecast	d_t	$d_{t-1}, A_{2,t-1}, A_{2,t-2}, A_{2,t-4}, A_{2,t-5}, A_{2,t-7}, cwd_t, swd_t, p^{ps}_{t-2}, p^{pw}_{t-2}, p^{bw}_{t-1}, s_{t-2}$
where	p^{bs} : Price of electricity base load one-summer-ahead p^{ps} : Price of electricity peak load one-summer-ahead p^{bw} : Price of electricity base load one-winter-ahead p^{pw} : Price of electricity peak load one-winter-ahead s: electricity supply d: daily electricity demand swd_t, cwd_t: two dummy variables presenting day of the week. $\widehat{\tau}$: scaled temperature A_2, D_2, D_1: WT components of d.	

Table 4.1: Input variables of MLP and RBF models for daily electricity demand.

4.5 Experiment results

4.5.1 Results on the electricity demand dataset

In addition to potential input variables as specified in Section 3.4, the wavelet components of the target value can be considered as inputs for the forecasting models as well. 2-level WTs were chosen for both electricity demand and gas price dataset (we determined the decomposition level by 10-fold cross-validation). Denote the WT components of the electricity demand by A_2, D_2, D_1. We used pre-processing procedures in Section 3.4 to select input variables. The selected input variables are shown in Tables 4.1 and 4.2.

The number of hidden units in MLP models for forecasting d (in the original MLP model), A_2, D_2, D_1 (in the multicomponent-forecast), and d (in the direct-forecast) were 12, 18, 11, 14, and 11 respectively. The numbers of hidden units in RBF models for forecasting d (in the original MLP model), A_2, D_2, D_1 (in the multicomponent-forecast), and d (in the direct-forecast) were 80, 100, 90, 115, and 95 respectively. These numbers were selected by 10-fold cross-validation (see page 48).

Tables 4.3 and 4.4 contains the IR$_{RMSE}$ and errors of the prediction methods for daily electricity demand forecasting. The tables show that the multicomponent-forecast methods outperform the direct-forecast methods and models without wavelet transform.

Methodologies	Target	Input variables
Without RHWT	d_t	$d_{t-1}, d_{t-6}, d_{t-7}, d_{t-8}, \widehat{\tau}_{t-1}, g_{t-1}, s_{t-1}, cwd_t$
Multicomponent-forecast	$A_{2,t}$	$A_{2,t-1}, A_{2,t-2}, A_{2,t-3}, A_{2,t-4}, A_{2,t-7}, A_{2,t-8}, A_{2,t-9}$, $d_{t-1}, d_{t-7}, s_{t-1}, \widehat{\tau}_{t-1}, g_{t-1}, cwd_t$
	$D_{2,t}$	$D_{2,t-1}, D_{2,t-2}, D_{2,t-4}, D_{2,t-5}, D_{2,t-13}, D_{2,t-14}, D_{2,t-15}$, $D_{1,t-1}, cwd_t$
	$D_{1,t}$	$D_{1,t-2}, D_{1,t-4}, D_{1,t-5}, D_{1,t-7}, swd_t$
Direct-forecast	d_t	$d_{t-1}, d_{t-6}, d_{t-7}, d_{t-8}$, $A_{2,t-1}, s_{t-1}, \widehat{\tau}_{t-1}, g_{t-1}, cwd_t$
where	g: gas demand. s: electricity supply. d: daily electricity demand. $\widehat{\tau}$: transformed temperature. swd_t, cwd_t: two dummy variables presenting day of the week. A_2, D_2, D_1: WT components of d.	

Table 4.2: Input variables of LR and LR-GARCH models for daily electricity demand.

Models	IR(RMSE)	RMSE	NRMSE	MAPE	MAE	NMAE
Benchmark	0.00%	39365	0.36550	2.96%	29011	0.32877
LR-GARCH	45.72%	21369	0.19841	1.72%	16538	0.18742
LR-GARCH+mf	46.05%	21237	0.19807	1.68%	16150	0.18312
LR-GARCH+df	45.77%	21348	0.19821	1.71%	16514	0.18715
LR	44.49%	21850	0.20252	1.76%	16915	0.19112
LR+mf	46.32%	21132	0.19805	1.66%	16093	0.18044
LR+df	45.02%	21643	0.20120	1.73%	16709	0.18898
MLP	53.12%	18455	0.17135	1.43%	13940	0.15798
MLP+mf	**58.15%**	**16474**	**0.15362**	**1.29%**	**12403**	**0.14065**
MLP+df	50.35%	19543	0.18003	1.48%	14665	0.16619
RBF	48.72%	20187	0.18743	1.63%	15589	0.17666
RBF+mf	**55.08%**	**17681**	**0.16335**	**1.38%**	**13194**	**0.14962**
RBF+df	47.64%	20612	0.19138	1.66%	15861	0.17974

Table 4.3: Errors and RMSE improvement ratio of forecasting methods using WT for daily electricity demand dataset. "mf" and "df" refer to multicomponent-forecast and direct-forecast respectively.

RMSE	Without WT	Multicomponent-forecast	Direct-forecast
LR-GARCH	21369	21237	21348
LR	21850	21132	21643
MLP	18455	**16474**	19543
RBF	20187	**17681**	20612

Table 4.4: Comparison of RMSE of forecasting methods with and without WT for daily electricity demand dataset.

This proves the usefulness of the WT in case of multicomponent-forecast. For example, the RMSE of the MLP (RBF) model is 18455 (20187) while that of the MLP (RBF) model combined with multicomponent-forecast method is 16474 (17681). The multicomponent-forecast method combined with MLP is the best with an RMSE of 16474, its RMSE improves 58.15% compared to the RMSE of the benchmark model.

Note that there are significant differences in lags selected for the different wavelet components of the multicomponent-forecast method (see Tables 4.1 and 4.2). This means that each component is highly correlated/relevant to a separate set of input variables. The multicomponent-forecast method can satisfy this restriction but the direct-forecast method cannot. This is why the multicomponent-forecast method achieves better results than the direct-forecast method.

4.5.2 Results on the gas forward price dataset

Denote the WT components of price of monthly gas forward product by A'_2, D'_2, D'_1. The number of hidden units in MLP models for forecasting p (in the original MLP model), A'_2, D'_2, D'_1 (in the multicomponent-forecast method), and p (in the direct-forecast method) are 8, 8, 10, 6, and 8 respectively. The numbers of basis functions in the RBF models for p in the original RBF model, A'_2, D'_2, and D'_1, and p in the direct-forecast were 30, 30, 10, 10, and 15. We used 10-fold cross validation to select the number hidden units and the number of basis functions (see Sections 3.3.2 and 3.3.3). Table 4.5 shows input variables for the gas forward price dataset.

The gas forward price dataset consists of 24 sub-datasets. The IR$_{RMSE}$, RMSE, NRMSE, MAPE, MAE and NMAE were computed for each sub-dataset and for each prediction method. Their averaged values are shown in Tables 4.6 and 4.7.

Methodologies	Target	Input variables
Without RHWT	p_t	$p_{t-1}, p_{t-2}, p^w_{t-1}, p^w_{t-2}$
Multicomponent-forecast	$A'_{2,t}$	$A'_{2,t-1}, A'_{2,t-2}, p_{t-1}, p_{t-2}, p^w_{t-1}, p^w_{t-2}$
	$D'_{2,t}$	$D'_{2,t-1}, D'_{1,t-1}, D'_{1,t-2}$
	$D'_{1,t}$	$D'_{1,t-1}, D'_{1,t-2}$
Direct-forecast	p_t	$p_{t-1}, p_{t-2}, A'_{2,t-1}, A'_{2,t-2}, p^w_{t-1}, p^w_{t-2}$
where	p^w: price of one-winter-ahead gas forward product. p^s: price of one-summer-ahead gas forward product. p: price of monthly gas forward product. A'_2, D'_2, D'_1: WT components of p.	

Table 4.5: Input variables for the first 12 sub-datasets in the gas forward dataset. The input variables for the remaining sub-datasets are similar, but p^w is replaced by p^s .

Models	IR(RMSE)	RMSE	NRMSE	MAPE	MAE	NMAE
Benchmark	0.00%	1.11862	0.48980	2.31%	0.84562	0.45182
LR	3.17%	1.08295	0.47735	2.26%	0.83577	0.44283
LR+df	2.32%	1.09048	0.48342	2.27%	0.84271	0.44852
LR+mf	**9.78%**	**1.00299**	**0.44319**	**2.05%**	**0.77315**	**0.41183**
LR-GARCH	3.77%	1.07378	0.47310	2.26%	0.83562	0.44281
LR-GARCH+df	3.82%	1.06662	0.47309	2.23%	0.82491	0.44091
LR-GARCH+mf	**9.41%**	**1.00614**	**0.44463**	**2.05%**	**0.77320**	**0.41407**
MLP	2.97%	1.09047	0.47941	2.27%	0.83586	0.44599
MLP+df	2.10%	1.09969	0.48405	2.26%	0.84294	0.44982
MLP+mf	**8.85%**	**1.01426**	**0.44477**	**2.08%**	**0.77889**	**0.41563**
RBF	2.15%	1.09969	0.48346	2.28%	0.84292	0.44976
RBF+df	2.92%	1.09048	0.47944	2.26%	0.83595	0.44602
RBF+mf	**8.08%**	**1.02283**	**0.44853**	**2.10%**	**0.78547**	**0.41914**

Table 4.6: Average errors and RMSE improvement ratio of forecasting methods using WT for gas forward price dataset.

NRMSE	Without WT	Multicomponent-forecast	Direct-forecast
LR	0.47735	**0.44319**	0.48342
LR-GARCH	0.47310	**0.44463**	0.47309
MLP	0.47941	**0.44477**	0.48405
RBF	0.48346	**0.44853**	0.47944

Table 4.7: Comparison of NRMSE of forecasting methods with and without WT for gas forward price dataset.

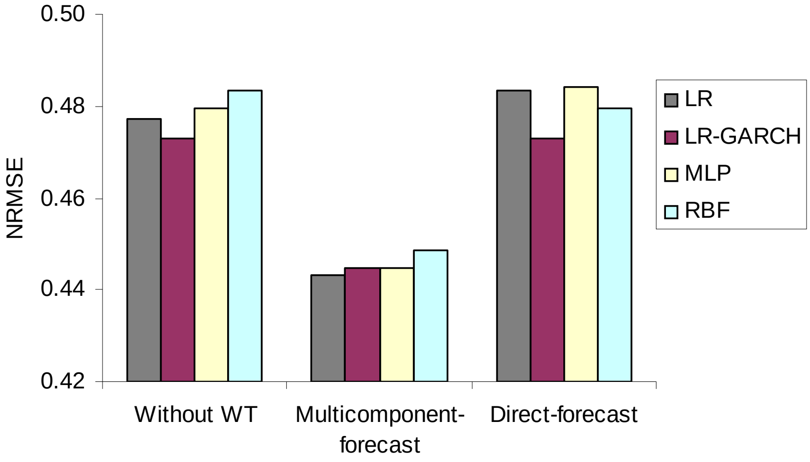


Figure 4.10: NRMSE of the forecasting models with and without WT on the gas forward price dataset.

NRMSE	Benchmark	LR+mf	LR-GARCH +mf	MLP+mf	RBF+mf
Price	0.48980	**0.44319**	**0.44463**	0.44477	0.44853
A_2	0.25169	0.13232	**0.13005**	0.13259	0.13592
D_2	0.87784	**0.41213**	0.41329	0.41337	0.44550
D_1	1.34394	1.00860	1.02119	**0.95603**	1.05186

Table 4.8: NRMSE of individual components and price in the multicomponent-forecast method for gas forward price dataset.

Similar to results on the daily electricity demand dataset, the multicomponent-forecast method outperforms the models without wavelet transform (see Figure 4.10). However results of the direct-forecast method are almost the same as the models without wavelet transform. The LR model with multicomponent-forecast achieved the best results with an NRMSE of 0.44319, which improves 9.78% compared to the NRMSE of the benchmark model.

Table 4.8 shows the average NRMSE of forecasting individual components in LR+mf, MLP+mf, RBF+mf, and LR-GARCH+mf for 24 sub-datasets. The average RMSE improvement ratios of these models are shown in Table 4.9. The errors on each individual component of these models are significantly smaller than the benchmark model. For example, the NMSEs of the components A_2, D_2, and D_1 in the LR-GARCH+mf improved 50.60%, 52.60%, and 23.51% respectively, compared to those in the benchmark model. Because each component presents data in a single range of frequency, it is possible to model the time evolution of the component more accurately than the raw data. However, the sum of all the components (i.e. the price) of these models are not that good: the IR_{RMSE} of the price in LR-GARCH+mf is only 9.41%.

To investigate the relatively small improvement of the overall performance, we analysed the correlation matrix of residuals of components in the method LR-GARCH+mf (see Table 4.10). The component residuals are quite highly correlated, especially for D_1 and D_2. Their correlation coefficients is 0.95684. Figure 4.11 shows the residuals of the test set of sub-dataset 10 using LR-GARCH+mf (the residual of a component is the difference between the predicted value and real value of that component). The shape of the residuals of components D_1 and D_2 are similar. The residuals of these components are normally the same sign, so their sum has a large magnitude. In the benchmark model, the signs

IR(RMSE)	LR+mf	LR-GARCH +mf	MLP+mf	RBF+mf
Price	**9.78%**	**9.41%**	8.85%	8.08%
A_2	49.80%	**50.60%**	48.37%	46.00%
D_2	**52.84%**	52.60%	52.15%	49.25%
D_1	24.49%	23.51%	**28.35%**	21.73%

Table 4.9: RMSE improvement ratio of individual components and price in the multicomponent-forecast method for gas forward price dataset.

Correlation matrix	A_2	D_2	D_1
A_2	1	0.82869	0.85600
D_2	0.82869	1	0.95684
D_1	0.85600	0.95684	1

Table 4.10: The averaged correlation matrix of residuals of components in method LR-GARCH+mf on the gas forward price dataset.

of component residuals are normally different, so they cancel when they are summed up. This is the reason why the IR$_{NMSE}$ of total of components in LR-GARCH+mf are not as large as the IR$_{NMSE}$ of each component.

4.6 Summary

This chapter presented approaches for applying the WT to prediction applications. The WT is used as a pre-processing procedure to decompose raw data into an approximation

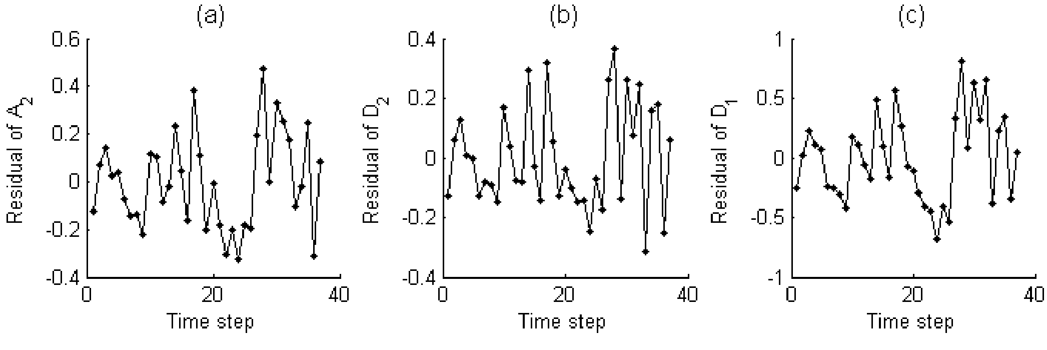


Figure 4.11: Residuals of components A_2 (a), D_2 (b), and D_1 (c) using LR-GARCH+mf for the test set of sub-datatset 10 in the gas price dataset.

component and detail components, of which each component represents the data in a relatively narrow frequency band. These components show the trend and details which are not observable in the raw data. We presented two methods for combining the WT with a range of standard prediction models: the multicomponent-forecast method and the direct-forecast method. The multicomponent-forecast method uses multiple prediction models in which each prediction model captures the development of each the WT component whereas the direct-forecast method uses only a single prediction model. We also empirically compared the prediction accuracy of the two methods.

The results of electricity demand forecasting and gas price forecasting show that the use of the WT improves the prediction performance. The multicomponent-forecast method consistently outperforms the direct-forecast method and models without wavelet transform. The results also show that the residuals of components D_1 and D_2 are highly correlated. This raises an open question for future study: how to use this special characteristic of these residuals to improve prediction performance. This will be discussed in more detail in Section 8.2 on page 151.

5 Adaptive models

In Chapter 3, we presented several standard models for forecasting energy demand and price; and in Chapter 4, we have described an alternative framework to improve prediction performance by using the wavelet transform. In this chapter, we present another approach to make these standard forecast models more effective: adaptive models in which there is online adjustment of the parameters in the test set.

5.1 Introduction

As mentioned in Chapter 3, standard forecasting models work well on the electricity demand dataset which is a stationary time series. However, their performance degrades in predicting non-stationary datasets, such as gas forward price. This is due to some specific characteristics of the data. The characteristics of a non-stationary time series change over time; thus the trend and volatility of training set might be different from these quantities of the corresponding test set. Therefore, the parameters of the prediction model, which are inferred from the training set, become “out of date” after some time. This means that these parameters might no longer capture the correct characteristics of the test set and this might lead to poor prediction performance.

In this chapter we try to reduce the effect of the above issue. We attempt to use observations of the time series as much as possible. The newly added values of price/demand are used for inferring parameters of prediction models. There are a number of approaches to implement this idea. The most trivial way is to update the training set by adding all observations up to the current time and re-train the model every time a new value of price/demand is observed. In this solution, if the test set has N data points, we must re-train the model $N-1$ times. This is very time consuming and computationally expensive. Another way is to use a filter to update the parameters of the forecasting model. Unlike the first approach, the model parameters on the adaptive model are updated by filters but we do not need to retrain the model with an iterative algorithm. Therefore, this is not only much faster than the first approach but also able to capture the impact of the new value of data into model parameters. The detailed framework for adaptive models will be presented in Section 5.2.

In the literature there are a range of papers on hybrid models, a combination of a filter (such as the Kalman filter (KF), or extended Kalman filter (EKF)) and a prediction model, such as radial basis function network, multi-layer perceptron, linear regression, or a financial model. The forecast model is used to forecast the next value of a time series, and the filter updates parameters of these models online as each new value of the time series is observed. Niranjan (1999) used the EKF algorithm to recursively re-estimate parameters of the Black-Scholes model from observations (the Black-Scholes model is a well-known financial model for options pricing). Nabney et al. (1996) showed that an EKF used for online learning parameters of an RBF model give much better tracking of non-stationary data than a fixed RBF model. Some researchers have proposed using an EKF in order to train an MLP. The results of predicting exchange rate (Andreou et al., 2002), estimating wind turbine power generation (Li et al., 1999), and predicting New England electricity prices (Zhang and Luh, 2002) showed that this method is good in the speed of learning and the accuracy of predictions. Parameters of linear models were also estimated using a KF in (Patil et al., 2006).

There are four points in this chapter. Firstly, we present an overall framework for adaptive models. Secondly, the filters used for the adaptive models are described, including

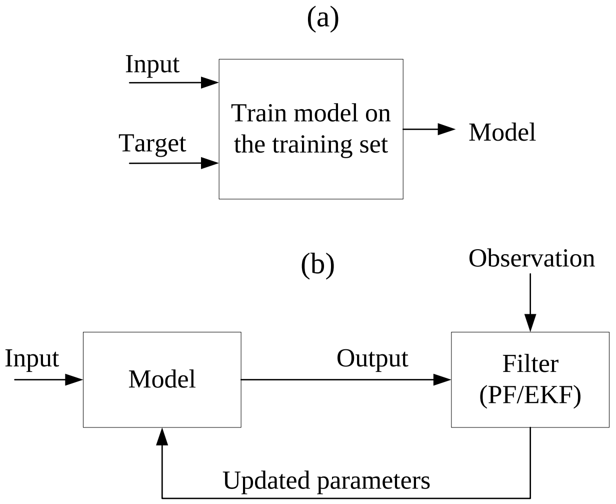


Figure 5.1: The adaptive model framework. (a) Training phase, (b) Test phase.

the KF, EKF, and particle filter (PF). Thirdly, we provide detail of how to combine each type of machine learning, time series, or financial models with the filters to generate adaptive models. Finally, the performance of the presented models are evaluated and compared by testing them on the energy price/demand in the UK market.

Among these adaptive models, the adaptive LR-GARCH and the adaptive financial stochastic models are novel. In addition, we use not only the EKF for adaptive models as earlier authors but also the PF. The PF has some advantages over the EKF: the PF makes no assumptions about the noise distribution, and also it is not necessary to linearise the prediction models as in the EKF.

5.2 Adaptive model framework

The parameters of a fixed prediction model are estimated using the training set only, and the test set is not used to adjust parameters. This constraint may reduce the forecast accuracy, especially in predicting non-stationary data. To overcome this, a filter (extended Kalman filter or particle filter) will be used to update parameters of a model by treating the weights as the states of a state space model (SSM). This can be considered as an estimation problem where the weight values are unknown. A general framework for an adaptive neural network for forecasting is shown in Figure 5.1.

In the training phase, the training set is used to estimate parameters of the model in the usual way (see Chapter 3). In the test phase, two steps are recursively repeated:

Step 1: When a new observation is available, the filter updates parameters of the predictive model (an observation here consists of an input-output pair).

Step 2: Use the predictive model with the latest estimated parameters to predict the next value.

The following sections describe the EKF and PF, and how to use them in adaptive models.

5.3 Filters

5.3.1 State space models

The KF/EKF/PF is based on a state space model (SSM), which is a time series model. The key of this model is that there are two processes happening: the true process and the observation process, and the state space model links these two processes. The true process is assumed to be unobservable and the variable $z_t \in R^k$ representing this process is called the hidden state vector. We assume that the observed time series $y_t \in R^p$ is a function of the hidden state space $z_t \in R^k$. In an adaptive model, z_t are the model parameters or a subset of these parameters. It is also assumed that we do not know the dynamics of the observation, but do know the dynamics of the hidden state (Figure 5.2):

$$\begin{aligned} z_{t+1} &= f_t(z_t) + \epsilon_t, \qquad \epsilon_t \sim \mathcal{D}(0, Q), & (5.1) \\ y_t &= h_t(z_t) + \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{D}(0, R_t), & (5.2) \end{aligned}$$

where f_t and h_t are the state transition function and output function respectively, ε_t and ϵ_t are zero-mean noises, and Q and R_t are the covariances of the noises ϵ_t and ε_t respectively. The hidden state vector obeys the Markov independence property (i.e. the current state depends only on the previous state). z_0 is the system initial condition, modelled as a Gaussian random vector $z_0 \sim \mathcal{D}(\pi_0, P_0)$.

In the adaptive model, the transition function f_t is selected as an identity function because we have no prior belief that the parameters should change with any particular dynamic. Denote the vector of the prediction model parameters which needs to be updated by z, then the evolution equation of these parameters is given by

$$z_t = z_{t-1} + \epsilon_t, \qquad \epsilon_t \sim \mathcal{D}(0, Q). \tag{5.3}$$

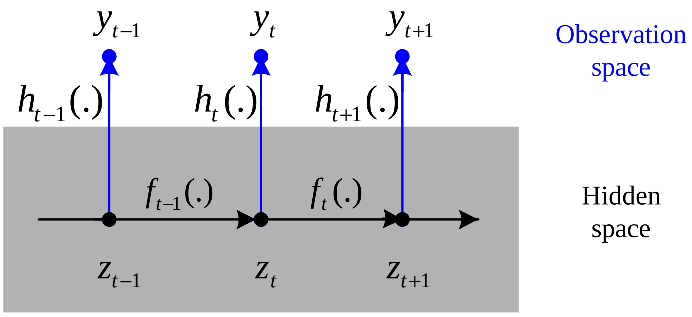


Figure 5.2: State space model.

In Equation (5.3), a random walk allows the parameters to adapt without a bias.

5.3.2 Kalman filter

The state inference problem for a state space model is to track the posterior probabilities of the hidden variables z_t given a sequence of observed variables up to time τ: $p(z_t|\{y\}_1^\tau)$, where $\{y\}_1^\tau = \{y_1, \ldots, y_\tau\}$. There are three cases of the inference problem: (1) filtering if $\tau = t$, (2) smoothing if $\tau > t$, and (3) prediction if $\tau < t$. In this thesis, we focus on filtering only and investigate how to use these filters for adaptive models. In theory, a filtering algorithm can be applied by sequentially iterating the following two steps:

- Predict:

$$p\left(z_t|\,\{y\}_1^{t-1}\right) = \int p\left(z_t|z_{t-1}\right) p\left(z_{t-1}|\,\{y\}_1^{t-1}\right)\, dz_{t-1}. \tag{5.4}$$

- Update:

$$p\left(z_t|\,\{y\}_1^{t}\right) = \frac{p\left(y_t|z_t\right) p\left(z_t|\,\{y\}_1^{t-1}\right)}{\int p\left(y_t|z_t\right) p\left(z_t|\,\{y\}_1^{t-1}\right)\, dz_t}. \tag{5.5}$$

The most difficult task is to compute the integrals in Equations (5.4) and (5.5); in the general case, it is impossible to analytically compute them (i.e. when the SSM is non-linear and the noises has an arbitrary distribution.). In order to make progress, researchers either place some restrictions on the state transition/output functions and noise distributions, so that these integrals become tractable, or some approximation is introduced. Kalman (1960) proposed an algorithm for inferring a special case of SSM in which the functions f_t

and g_t are linear and the noise models are Gaussian distributions. In this case, the state transition function f_t and the output function g_t can be represented by $f_t(z_t) = F_t z_t$ and $h_t(z_t) = H_t z_t$. Therefore, the SSM becomes:

$$\begin{aligned} z_{t+1} &= F_t z_t + \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, Q), && (5.6)\\ y_t &= H_t z_t + \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}(0, R_t), && (5.7)\end{aligned}$$

where F_t and H_t are matrices and $\mathcal{N}(0,\cdot)$ are zero-mean Gaussian noises. Equations (5.6) and (5.7) show that if $p(z_{t-1})$ is Gaussian, then so are $p(z_t)$ and $p(y_t)$. The Kalman filter is a recursive algorithm and each iteration has two phases:

- Prediction:

$$\begin{aligned} z_t^{t-1} &= F_t z_{t-1}^{t-1}, && (5.8)\\ P_t^{t-1} &= F_t P_{t-1}^{t-1} F_t' + Q, && (5.9)\end{aligned}$$

- Update

$$\begin{aligned} K_t &= P_t^{t-1} H_t' \left(H_t P_t^{t-1} H_t' + R_t \right)^{-1} \quad \text{(Kalman gain)} && (5.10)\\ e_t &= y_t - H_t z_t^{t-1}, && (5.11)\\ z_t^t &= z_t^{t-1} + K_t e_t, && (5.12)\\ P_t^t &= \left(I - K_t H_t\right) P_t^{t-1}, && (5.13)\end{aligned}$$

where

$$\begin{aligned} z_t^t &= E[z_t | \{y\}_1^t],\\ z_t^{t-1} &= E[z_t | \{y\}_1^{t-1}],\\ P_t^t &= E[(z_t - z_t^t)(z_t - z_t^t)' | \{y\}_1^t],\\ P_t^{t-1} &= E[(z_t - z_t^{t-1})(z_t - z_t^{t-1})' | \{y\}_1^{t-1}],\end{aligned}$$

z_t^{t-1} is the *a priori* state estimate at time step t given knowledge of the process prior to step t, and z_t^t is an *a posteriori* state estimate at time step t given measurement y_t. The matrices P_t^{t-1} and P_t^t are the *a priori* estimate error covariance and the *a* *posteriori* estimate error covariance respectively.

In the prediction phase, the *a priori* state estimate z_t^{t-1} and the *a priori* estimate error covariance P_t^{t-1} are computed forward from time step $t-1$ to step t. In the update phase, we assume that we have measured the process to obtain y_t. Firstly, the Kalman gain K_t is computed. Then the *a posteriori* state estimate z_t^t and the *a posteriori* estimate error covariance P_t^t are calculated. The initial values are $z_1^0 = \pi_0$ and $P_1^0 = P_0$.

5.3.3 Extended Kalman filter

The EKF is an extension of the Kalman filter. Both of them are recursive algorithms used to compute the probability of the current hidden state space z_t given the sequence of observations up to time t. Their difference is that the Kalman filter is designed for linear state space models (i.e. h_t and f_t are linear functions) only while the EKF can be applied to either linear or non-linear models. In the KF, a Gaussian distribution is propagated through linear functions h and f. Therefore, if $p(z_{t-1})$ is a Gaussian, then so is $p(z_t)$. However, when h or f are non-linear functions, propagating a Gaussian distribution through a non-linear function produces a non-Gaussian output function. Therefore, tracking of evolution of the full probability distribution function is impossible.

One approach is to make an approximation. The EKF does not solve the original problem, but approximates it by locally linearising the non-linear functions $f_{t-1}(z)$ around z_{t-1}^{t-1} in the prediction step and locally linearising the output function $h_t(z)$ around z_t^{t-1} in the update step. The linearisations are made using the first-order Taylor expansion:

$$f_{t-1}(z_{t-1}) = f_{t-1}(z_{t-1}^{t-1}) + \widehat{F}_{t-1}(z_{t-1} - z_{t-1}^{t-1}), \tag{5.14}$$

$$h_t(z_t) = h_t(z_t^{t-1}) + \widehat{H}_t(z_t - z_t^{t-1}). \tag{5.15}$$

where $\widehat{F}_t$ and $\widehat{H}_t$ are the Jacobian matrices of the functions $f(\cdot)$ and $h(\cdot)$ evaluated at z_t^t and z_t^{t-1} respectively: $\widehat{F}_t = \nabla f_t|z_t^t$ and $\widehat{H}_t = \nabla h_t|z_t^{t-1}$. The Jacobian matrix of a function $f : R^n \to R^m$, $y^{(1)}(z^{(1)}, \dots, z^{(n)}), \dots, y^{(m)}(z^{(1)}, \dots, z^{(n)})$ is given by

$$\bigtriangledown f|z = \begin{bmatrix} \frac{\partial y^{(1)}}{\partial z^{(1)}} & \cdots & \frac{\partial y^{(1)}}{\partial z^{(n)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial y^{(m)}}{\partial z^{(1)}} & \cdots & \frac{\partial y^{(m)}}{\partial z^{(n)}} \end{bmatrix}_z .$$

The general SSM in Equations (5.1) and (5.2) becomes a linear SSM. Therefore, the EKF is similar to the Kalman filter. One iteration of the EKF is composed of the following steps (Ribeiro, 2004):

- Prediction:

$$\begin{aligned} z_t^{t-1} &= f_{t-1}(z_{t-1}^{t-1}), \\ P_t^{t-1} &= \widehat{F}_t P_{t-1}^{t-1} \widehat{F}_t' + Q. \end{aligned}$$

- Update:

$$\begin{aligned} K_t &= P_t^{t-1} \widehat{H}_t' [\widehat{H}_t P_t^{t-1} \widehat{H}_t' + R_t]^{-1}, \\ e_t &= y_t - h_t(z_t^{t-1}), \\ z_t^t &= z_t^{t-1} + K_t e_t, \\ P_t^t &= [I - K_t \widehat{H}_t] P_t^{t-1}. \end{aligned}$$

Note that we use the approximation matrices $\widehat{F}_t$ and $\widehat{H}_t$ to update the variances P_t^{t-1} and P_t^t while the original functions f_t and h_t are used to update the means z_t^{t-1} and z_t^t.

5.3.4 Particle filter

The EKF does not solve the original problem, but simplifies it by locally linearising the functions f_t and h_t around previous state estimates (see Equations (5.14) and (5.15)). They may not be good approximations given that some quadratic terms are discarded, along with higher order terms. Especially when the strength of the non-linearity of these functions is great, the linearisations are poor approximations, and the EKF does not work well. The particle filter (PF) is an alternative method to avoid the bad effects of linearisation. The particle filter is more robust than the EKF because it can work well on very non-linear models. In addition, the EKF is limited to Gaussian noise for ϵ_t and ε_t while there is no assumption of noise distributions for the PF.

The PF is a sampling-based method. Firstly, we sample N_p times from an initial distribution $z_{0,i} \sim p(z_0) = \mathcal{D}(\pi_0, P_0)$, and allocate equal weights $w_{0,i} = 1/N_p$. Then, we have N_p samples at time step $t = 0$, called N_p particles. After that, the state mean at t given y_t is estimated by the following steps:

- Evolve particles using the transition function f_t: $z_{t,i} = f_t(z_{t-1,i}) + \eta_i$, where $\eta_i \sim \mathcal{D}(0, Q)$, $i = 1, \ldots N_p$.

- Re-weight particles each time a new observation is available: $w'_{t,i} \propto w_{t-1,i} p(y_t|z_{t,i})$, where $p(y_t|z_{t,i}) = \mathcal{D}(y_t|h_t(z_{t,i}), R_t)$, $i = 1, \ldots, N_p$.

- Normalise the weights: $w_{t,i} = w'_{t,i} / \sum_{i=1}^{N_p} w'_{t,i}$.

- The mean of state z at time step t is the weighted average of particles: $E[z_t| \{y\}_1^t] = \sum_{i=1}^{N_p} w_{t,i} z_{t,i}$.

The observation y_t is reflected in the weights $w_{t,i}$, $i = 1, \ldots, N_p$. If a particle $z_{t,i}$ is far from the true value, $p(y_t|z_{t,i})$ is small, thus so is $w_{t,i}$. This leads to the fact that this particle makes an insignificant contribution to the mean $E[z_t| \{y\}_1^t]$. On the other hand, if a particle is close to the true value, its contribution is large.

In practice, after a large number of time steps, all but a small number of particles may have negligible weight. The problem with this degeneracy is that most of the particles contribute insignificantly to $E[z_t| \{y\}_1^t]$, but they still consume computational effort. To measure this degeneracy, a new parameter, called the effective sample size (E_t), is introduced (Arulampalam et al., 2002). The smaller E_t is, the greater the degeneracy level. It is impossible to exactly evaluate this parameter, but an estimate $\widehat{E}_t$ of E_t can be computed by $\widehat{E}_t = \left\{ \sum_{i=1}^{N_p} (w_{t,i})^2 \right\}^{-1}$.

We can reduce the effect of degeneracy by resampling. A threshold of degeneracy (E_{thres}) is set up. At each time step t, if $\widehat{E}_t < E_{thres}$, resampling is used, as follows:

- Approximate the distribution of particles $z_{t,i}$ by a Gaussian mixture model distribution ($\mathcal{G}$) with N_p centres $z_{t,i}$, $i = 1, \ldots, N_p$ and equal covariances. The details and code for Gaussian mixture model can be found in (Nabney, 2002).

- Sample N_p times from distribution $\mathcal{G}$: $z'_{t,i} \sim \mathcal{G}$, and allocate equal weights $w'_{t,i} = 1/N_p$.

- Assign $z_{t,i} = z'_{t,i}$ and $w_{t,i} = w'_{t,i}$.

Note that no assumptions are made about either functions of the state space model or source of noise. This means that the particle filter can be applied to a general SSM: state transition/output functions can be either linear or non-linear; and noise can be any choice of distributions. Conversely, the extended Kalman filter is limited to SSM whose noise are Gaussian distributions only. Thanks to this characteristic, particle filters can be used for a larger range of forecasting models than extended Kalman filters. This is important because this thesis will discuss not only Gaussian but also Student-t noise models. More detail about these noise models will be presented in Chapter 6.

5.4 Adaptive LR/MLP/RBF models

This section presents how to generate adaptive models from the LR/MLP/RBF which were presented in Section 3.3.1, 3.3.2, and 3.3.3. Combination of the EKF with these machine learning models have been studied in the literature (Lowe and McLachlan, 1995, Nabney et al., 1996, de Freitas et al., 1999, Andreou et al., 2002, Zhang and Luh, 2002, Patil et al., 2006). In adaptive models, we can choose to update all the parameters or only a subset (e.g. only the bias in LR). We did experiments on some scenarios of updating. In the adaptive LR models, we tested on updating the bias b only or updating all parameters $\{\omega, b\}$. In MLP and RBF models, we tested on updating bias ω_0 only or all second layer parameters $\{\omega_j\}_{j=0}^{M}$. The experimental results showed that results on updating the bias only is slightly better than updating more parameters (see Tables E.1 and E.2 in Appendix E, page 178) and hence we restrict our attention to this case.

Updating the bias implies that the models adapt only to changes in the trend (mean), but not the volatility of the time series. Denote these updated parameters θ, and the remaining parameters of a model ϖ. From Equations (3.1), (3.4), (3.5), (3.7), and (3.8), we can summarise the input/output relationship of the models as follows

$$\widehat{y}_t = h(x_t, \theta, \varpi). \tag{5.16}$$

On the training set

We used the same training algorithms in fixed models in Sections 3.3.1, 3.3.2, and 3.3.3 to estimate parameters, denoted θ_0, ϖ_0.

On the test set

We fixed the value $\varpi_t = \varpi_0$, and used a filter to update value of θ_0. For this purpose, a state space model is constructed as follows:

$$\begin{aligned} \theta_t &= \theta_{t-1} + \epsilon_t, & \epsilon_t \sim \mathcal{D}(0, Q), && (5.17) \\ y_t &= h(x_t, \theta_t, \varpi_0) + \varepsilon_t, & \varepsilon_t \sim \mathcal{D}(0, R), && (5.18) \end{aligned}$$

where the bias θ_t in Equation (5.17) is the hidden state vector, the output function $h(x_t, \theta_t, \varpi_0)$ defined by Equations (3.1), (3.4), (3.5), (3.7), and (3.8), and the state transition function is selected as an identity function. Parameters Q, R and P_0 of the SSM can be estimated by using maximum log likelihood (Ghahramani and Hinton, 1996) or just set to relatively small values. The initial state of the SSM is given by: $\pi_0 = \theta_0$.

On the test set, two steps are repeated for the observations in time order:

- *Step 1*: Estimate parameter θ_t of the prediction model: $\theta_t^{t-1} = E\left[\theta_t | \{y\}_1^{t-1}\right]$. In order to obtain this parameter, we use the extended Kalman filter or particle filter to estimate the mean of the hidden state of the above SSM at time step $t-1$ given observations up to time $t-1$: θ_{t-1}^{t-1}. From Equation (5.17), estimation of θ at time step t given observation up to time $t-1$ is $\theta_t^{t-1} = \theta_{t-1}^{t-1}$.

- *Step 2*: Use MLP/RBF/LR model with the latest estimated parameters to predict time series at time step t:

$$y_t = h(x_t, \theta_t^{t-1}, \varpi_0).$$

If we use the extended Kalman filter, when linearising the output function $h(\cdot)$ and state transition functions $f(\cdot)$, the Jacobian matrices $\widehat{F}_t$ and $\widehat{H}_t$ are computed as follows:

$$\begin{aligned} \widehat{F}_t &= \nabla f | \theta_t^t = I \text{ (i.e. identity matrix)}, && (5.19) \\ \widehat{H}_t &= \nabla h | \theta_t^{t-1}. && (5.20) \end{aligned}$$

Because we chose to update the bias of the second layer ω_0 only (in the MLP and RBF) and the bias b_0 in the LR, the Jacobian matrices $\widehat{F}_t$ and $\widehat{H}_t$ are computed as the follows:

$$\begin{aligned}\widehat{F}_t &= \nabla f | b_t^t = 1,\\ \widehat{H}_t &= \nabla h_{lr} | b_t^{t-1} = 1.\end{aligned}$$

Discussions on the adaptive models with bias updating

As mentioned above, we focused on the adaptive models with updating bias (or the constant terms) only because they did slightly better than the adaptive models with updating more parameters. In this case, the MLP, RBF and LR models have time-varying constant terms. Therefore they are somehow similar to regression models with residuals that have an autoregressive moving average model (ARMA). These are sometimes termed ARMAX models, and are reasonably common in the literature.

We can explain the similarity of them in more detail by the following mathematical equations. We presented a forecasting model in the form of

$$\widehat{y} = b + \Phi(\mathbf{x}, \boldsymbol{\omega}), \tag{5.21}$$

where b is the bias (intercept) and $\Phi(\mathbf{x}, \boldsymbol{\omega})$ is the rest of the model. If only the bias term is updated in an adaptive model, b changes over time and the equation for updating this parameter is given by:

$$b_t = b_{t-1} + K_t e_t, \tag{5.22}$$

where K_t is the Kalman gain and e_t is the residual. Equation (5.22) is similar to a regression model with a residual that has an AR model.

However, there are some difference between these two kind of regression models. First, in the bias-adaptive model the Kalman gain K_t changes over time (see Equation (5.10) on page 99) while the corresponding parameter in the ARMA/ARMAX model, which is a parameter in the MA component, is fixed. Second, the ARMA/ARMAX model is more flexible in the sense that it is not limited to orders (1, 1) of the autoregressive and moving average parts as in the bias updating equation. Therefore, the ARMA/ARMAX model can be a good alternative approach for the adaptive models with updating the bias. We will discuss the ARMA/ARMAX in the future work (see Section 8.2 on page 151).

5.5 Adaptive LR-GARCH

This adaptive model is proposed for the first time in this thesis. Similar to adaptive MLP/RBF/LR models, the adaptive LR-GARCH model with updating only bias $\widetilde{\beta}$ is slightly better than the adaptive models with updating more parameters (both $\widetilde{\beta}$ and $\widehat{\beta}$) (see Tables E.1 and E.2 in Appendix E, page 178). In this case, all parameters of the LR-GARCH model are set on the training set, with the exception of the bias $\widetilde{\beta}$ which is adapted online on the test set. Define $\delta = \{\alpha_0, \ldots, \alpha_m, \gamma_1, \ldots, \gamma_r\}$ (see definitions of these notations in Section 3.3.4, page 50).

On the training set

We use maximum likelihood to compute LR-GARCH parameters as in the fixed LR-GARCH model (Section 3.3.4), denote these optimisation parameters $\delta_0 = \{\alpha_{0,0}, \ldots, \alpha_{m,0}, \gamma_{1,0}, \ldots, \gamma_{r,0}$ $\widehat{\beta}_0$, and $\widetilde{\beta}_0$.

On the test set

We fix the value $\delta_t = \delta_0$ and $\widehat{\beta}_t = \widehat{\beta}_0$, and use a filter to update the value of $\widetilde{\beta}_t$. Two steps are recursively repeated.

- *Step 1*: Update parameters of the model using the EKF/PF. The non-linear SSM is given by

$$\widetilde{\beta}_t = \widetilde{\beta}_{t-1} + \epsilon_t, \qquad \epsilon_t \sim \mathcal{D}(0, Q), \tag{5.23}$$

$$y_t = h_t(\widetilde{\beta}_t) + \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{D}(0, R_t), \tag{5.24}$$

where the bias $\widetilde{\beta}$ is the hidden state vector of the SSM, the output function $h_t(\widetilde{\beta}_t) = \widetilde{\beta}_t + \widehat{\beta}_0 x_t$, and parameters Q and the variance P_0 of the initial state z_0 (see page 97) can be estimated by using maximum log likelihood (using the Kalman smoother) (Ghahramani and Hinton, 1996), or just initialised to relatively small values. Other parameters of the SSM are given by

$$\begin{aligned} \pi_0 &= \widetilde{\beta}_0,\ \widehat{F}_t = 1,\ \widehat{H}_t = 1, \\ R_t &= \alpha_{0,0} + \sum_{i=1}^{m} \alpha_{i,0} \varepsilon_{t-i}^2 + \sum_{j=1}^{r} \gamma_{j,0} R_{t-j}, \end{aligned}$$

(see definition of these parameters in Section 3.3.4, page 50).

- *Step 2*: Use the LR-GARCH model with the latest estimated parameters to predict the time series at time step t:

$$y_t = \widetilde{\beta}_t^{t-1} + \widehat{\beta}_0 x_t.$$

5.6 Adaptive financial stochastic models

These kinds of adaptive models are derived from the financial stochastic models presented in Section 3.3.5, page 53. Note that the financial stochastic models are specified for the electricity price dataset only. We do not directly forecast forward prices, but estimate the log-return.

In these adaptive financial stochastic models, all parameters $\widehat{\theta}$ are estimated on the training set, and then are adapted online on the test set.

On the training set

We use maximum likelihood to estimate parameters as in the fixed financial stochastic model: denote these optimised parameters $\widehat{\theta}_0$.

On the test set

To update the model parameters, a state space model is constructed as follows:

$$\widehat{\theta}_t = \widehat{\theta}_{t-1} + \epsilon_t, \qquad \epsilon_t \sim \mathcal{D}(0, Q), \tag{5.25}$$

$$r_t = h_E(\widehat{\theta}_t) + \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{D}(0, R_t), \tag{5.26}$$

where $\widehat{\theta}$ in Equation (5.25) is the hidden state vector, $h_E(\widehat{\theta}_t) = m_t(\widehat{\theta}_t)$ is the output function and $m_t(\widehat{\theta}_t)$ is defined in Section 3.3.5 (on page 54), and parameters Q and P_0 of the SSM can be estimated by using maximum log likelihood (using the Kalman smoother), or just initialised to relatively small values. Other parameters of the SSM are given by

$$\begin{aligned}
\pi_0 &= \widehat{\theta}_0, \\
F_t &= I \text{ (i.e. identity matrix)}, \\
H_t &= \frac{\partial m_t(\widehat{\theta})}{\partial \widehat{\theta}} | \widehat{\theta}_t^{t-1}, \\
R_t &= \nu_t(\widehat{\theta}_t^{t-1}),
\end{aligned}$$

where $m_t(\widehat{\theta})$ and $\nu_t(\widehat{\theta})$ are defined in section 3.3.5. Note that the noise variance of the output function R_t in this model is not fixed, but is estimated over time. This is helpful in updating the financial forecasting model parameters (i.e. hidden states in SSM) as in Equation (5.10), (5.11), and (5.12).

Two steps are repeated through the observations in time order:

- *Step 1*: Estimate parameter $\widehat{\theta}_t$ of the financial model: $\widehat{\theta}_t^{t-1} = E\left[\widehat{\theta}_t | \{r\}_1^{t-1}\right]$. Similar to the other adaptive model, we use the EKF or PF to estimate $\widehat{\theta}_{t-1}^{t-1}$. Then an estimate of the hidden state vector at time step t given observation up to time $t-1$ is $\widehat{\theta}_t^{t-1} = \widehat{\theta}_{t-1}^{t-1}$.
- *Step 2*: Use the financial model with the latest estimated parameters to predict time series at time step t: $r_t = h_E(\widehat{\theta}_t^{t-1})$.

As mentioned in Section 3.3.5 (page 53), there are six different financial stochastic models, E1-E6. We here present an example of the adaptive financial stochastic models: the adaptive E1.

In the adaptive E1, the hidden state vector is $\widehat{\theta} = \{\widehat{a}, \lambda\}$. The state space model for an adaptive E1 is given by Equations (5.25) and (5.26), where

$$\begin{aligned}
h_E(\widehat{\theta}_t) &= \lambda_t e^{\widehat{a}_t} - \frac{1}{2} e^{2\widehat{a}_t}, \\
\pi_0 &= \widehat{\theta}_0, \\
F_t &= I\,, \\
H_t &= [\lambda_t^{t-1} e^{\widehat{a}_t^{t-1}} - e^{2\widehat{a}_t^{t-1}}, e^{\widehat{a}_t^{t-1}}], \\
R_t &= e^{2\widehat{a}_t^{t-1}}.
\end{aligned}$$

5.7 Experimental results

The adaptive machine learning models (i.e. MLP, RBF, LR-GARCH, and LR) are evaluated on two datasets: gas forward price and electricity demand. The adaptive financial stochastic models are specified for electricity prices, therefore these models are tested on electricity forward price only.

5.7.1 Results on the gas forward price dataset

The gas forward price dataset consists of 24 sub-datasets. The IR_{RMSE}, RMSE, NRMSE, MAPE, MAE and NMAE were computed for each sub-dataset and for each fixed or adaptive prediction method. Their averaged values over 24 sub-datasets are shown in Tables 5.1 and 5.2, and Figure 5.3. For the purpose of comparison, these errors were computed for the following models:

- Random walk model, which was used as the benchmark model.
- Fixed models, whose parameters were computed on the training set only.
- Adaptive models with EKF, in which the EKF was used to adjust online parameters of the prediction models on the test set.
- Adaptive models with PF, in which PF was used to update online parameters of the prediction models on the test set.

"LR+EKF" and "LR+PF" referred to adaptive LR models with EKF and PF respectively. Similar notation was used for LR-GARCH, RBF and MLP models.

Tables 5.1 and 5.2 and Figure 5.3 show that the adaptive models are better than the fixed models, which proved the usefulness of using filters. For example, the improvement ratio of RMSE of fixed LR-GARCH model was 3.77% while that of the adaptive LR-GARCH model was 6.16%.

In general, the adaptive models with PF are expected to provide better performance than the adaptive models with EKF because the PF does not require as many assumptions as in EKF. However, the results of these adaptive models were almost the same in these cases. This could be explained by the linearity of the state space models. The state transition functions in Equations (5.17) on page 104, (5.23) on page 106, and (5.25) on page 107 are linear. The outputs of LR-GARCH and LR models are linear in parameters,

Models	IR(RMSE)	RMSE	NRMSE	MAPE	MAE	NMAE
Benchmark	0.00%	1.11862	0.48980	2.31%	0.84562	0.45182
LR	3.17%	1.08295	0.47735	2.26%	0.83577	0.44283
LR+EKF	4.49%	1.06916	0.47248	2.22%	0.83044	0.43795
LR+PF	4.47%	1.06917	0.47249	2.22%	0.83048	0.43797
LR-GARCH	3.77%	1.07378	0.47310	2.26%	0.83562	0.44281
LR-GARCH+EKF	**6.16%**	**1.04144**	**0.46056**	**2.18%**	**0.80492**	**0.43031**
LR-GARCH+PF	**6.15%**	**1.04145**	**0.46063**	**2.19%**	**0.80551**	**0.43050**
MLP	2.97%	1.09047	0.47941	2.27%	0.83586	0.44599
MLP+EKF	4.82%	1.06588	0.47241	2.21%	0.81169	0.43790
MLP+PF	4.82%	1.06586	0.47240	2.21%	0.81153	0.43735
RBF	2.15%	1.09969	0.48346	2.28%	0.84292	0.44976
RBF+EKF	4.86%	1.06347	0.47111	2.20%	0.81152	0.43735
RBF+PF	4.87%	1.06327	0.47111	2.20%	0.81149	0.43392

Table 5.1: Errors and Improvement Ratio of RMSE of forecast methods for the gas forward price dataset.

NRMSE	Fixed	Adaptive with EKF	Adaptive with PF
LR	0.47735	0.47248	0.47249
LR-GARCH	0.47310	**0.46056**	**0.46063**
MLP	0.47941	0.47241	0.47240
RBF	0.48346	0.47111	0.47111

Table 5.2: Improvement ratio of RMSE of the fixed and adaptive methods for the forward gas price dataset.

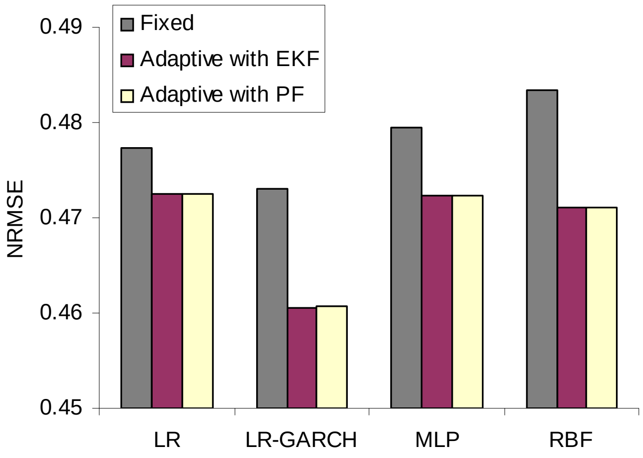


Figure 5.3: NRMSE of the fixed and adaptive forecasting models for the gas forward price dataset.

thus so are their state space models. MLP and RBF models are non-linear in parameters, but we only adjusted on-line the bias of the second layer in MLP and RBF, which have linear relationships with the outputs. Therefore the state space models for these adaptive MLP and RBF models are also linear. The local linearisation of output functions and state transition functions on EKF are perfect and the EKF can provide good updates.

Note that when SSMs are linear, the EKF and the KF provide the same results. Although the simple KF is good enough for these linear SSMs, we actually used the EKF, which is more complicated but has similar efficiency, in our experiments. The reason is that at the beginning we developed adaptive models for two scenarios: (1) updating all parameters and (2) updating a subset only. The SSMs associated with the first scenarios for the MLP are non-linear, so we had to implement the EKF for this case. Because the EKF can be used for both scenarios, we did not need to implement the KF for the second scenario, but just used the EKF instead.

Figure 5.4 shows IR_{RMSE} of the adaptive and fixed MLP models for 24 sub-datasets of the gas forward price. The adaptive models generally achieved better prediction accuracy than the fixed models. There was no difference between the prediction performance of the adaptive model with EKF and the adaptive models with PF: the lines showing their improvement ratios overlap.

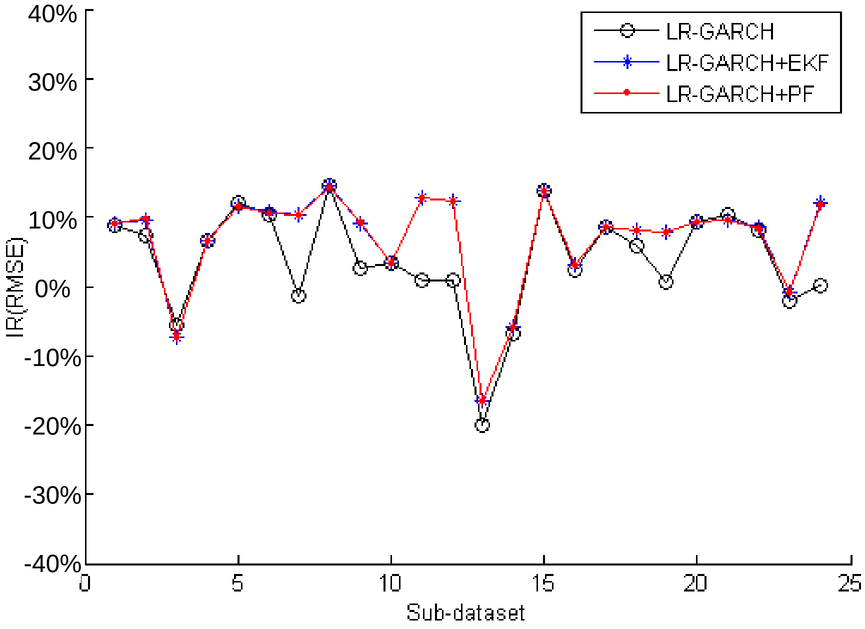


Figure 5.4: IR_{RMSE} of the fixed and adaptive LR-GARCH models for 24 sub-datasets on the gas price forecast.

5.7.2 Results on the electricity demand dataset

Tables 5.3, 5.4 and Figure 5.5 provide the results of the prediction methods for the daily electricity demand. The adaptive models performed better than the fixed models. There was no significant difference between performance of the adaptive models with EKF and PF. The adaptive MLP models achieved the best results with RMSE of 17266, which improved 56.14% comparing to RMSE of the benchmark model. In this dataset, the non-linear models (MLP/RBF) generally provided better prediction accuracy than the linear models (LR-GARCH/LR).

5.7.3 Results on the electricity forward price dataset

This section is dedicated to testing the performance of the adaptive financial stochastic models in Section 5.6. As mentioned in Section 3.3.5 on page 53, because these financial stochastic models are designed for electricity forward prices but not for a general time series, we tested them on the UK electricity forward prices. Note that due to this reason, we cannot apply the multicomponent-forecast of the WT for these models. We cannot apply direct-forecast either because this kind of model requires a certain set of variables as inputs (including time step, start and end time of the delivery period), but are not a

Models	IR(RMSE)	RMSE	NRMSE	MAPE	MAE	NMAE
Benchmark	0.00%	39365	0.36550	2.96%	29011	0.32877
Fixed LR-GARCH	45.72%	21369	0.19841	1.72%	16538	0.18742
LR-GARCH+EKF	45.86%	21312	0.19772	1.69%	16307	0.18432
LR-GARCH+PF	46.06%	21232	0.19698	1.69%	16305	0.18364
Fixed LR	44.49%	21850	0.20252	1.76%	16915	0.19112
LR+EKF	45.80%	21337	0.19803	1.70%	16352	0.18560
LR+PF	45.78%	21343	0.19808	1.70%	16354	0.18562
Fixed MLP	53.12%	18455	0.17135	1.43%	13940	0.15798
MLP+EKF	**56.14%**	**17266**	**0.16031**	**1.36%**	**13186**	**0.14886**
MLP+PF	**55.56%**	**17493**	**0.16213**	**1.36%**	**13186**	**0.14900**
Fixed RBF	48.72%	20187	0.18743	1.63%	15589	0.17666
RBF+EKF	49.72%	19792	0.18284	1.57%	15325	0.17315
RBF+PF	49.68%	19810	0.18301	1.58%	15338	0.17401

Table 5.3: Errors and Improvement ratio of RMSE of forecast methods for the electricity demand dataset.

RMSE	Fixed	Adaptive with EKF	Adaptive with PF
LR-GARCH	21369	21312	21232
LR	21850	21337	21343
MLP	18455	**17266**	**17493**
RBF	20187	19792	19810

Table 5.4: RMSE of fixed and adaptive models for the electricity demand dataset.

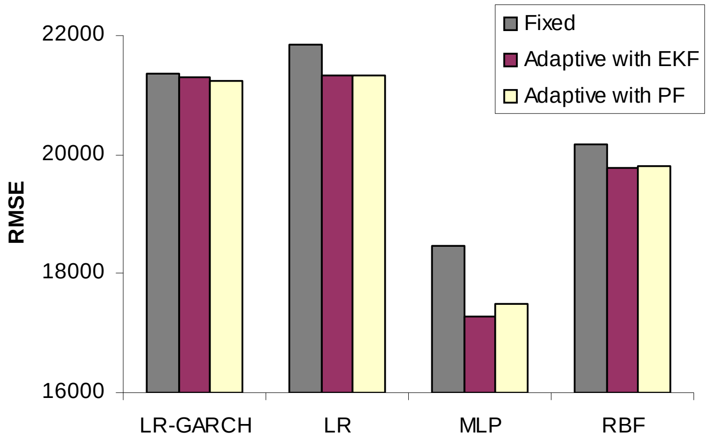


Figure 5.5: RMSE of fixed and adaptive models on the electricity demand dataset.

black box as in the machine learning models.

Tables 5.5 and 5.6 show results of these models on the dataset of the base load electricity monthly forward price. The tables show that the adaptive financial stochastic models did not work on this dataset: their improvement ratios are around zero. This means that their performance is almost the same as the benchmark model. We actually tested these fixed and adaptive models on other electricity forward prices: base load/peak load quarterly/seasonal forward prices, but the results are the same or even a little worse than those shown in these tables.

5.8 Summary

This chapter presented a framework of adaptive models for prediction applications. It also shows how to combine each type of prediction model with filters. In adaptive models, the model parameters are not fixed but updated online every time a new observation is available. Therefore, we can make the most use out of the data. This makes the prediction models more plastic and can provide good results, especially for non-stationary datasets.

It was shown experimentally that the adaptive machine learning and time series models did improve prediction performance. However, this improvement is not as great as the improvement induced by using the WT in Chapter 4. In these prediction models, noise is assumed to be Gaussian, therefore we can use both EKF and PF as filters. The adaptive models with PF and the adaptive models with EKF achieved similar results. In the gas price forecast, the adaptive LR-GARCH models achieves best results with NRMSE of

Models	IR(RMSE)	RMSE	NRMSE	MAPE	MAE	NMAE
RW	0.00%	1.5554	0.52192	2.050%	0.94011	0.39467
E1+EKF	**0.60%**	**1.5460**	**0.51879**	**2.030%**	**0.92750**	**0.38938**
E2+EKF	-0.07%	1.5565	0.52230	2.071%	0.94324	0.39599
E3+EKF	-1.62%	1.5806	0.53038	2.064%	0.94167	0.39533
E4+EKF	-4.35%	1.6230	0.54463	2.180%	1.01427	0.42581
E6+EKF	**0.27%**	**1.5512**	**0.52053**	**2.041%**	**0.93418**	**0.39218**
E1+PF	**0.42%**	**1.5488**	**0.51971**	**2.034%**	**0.92897**	**0.39000**
E2+PF	-0.36%	1.5610	0.52380	2.082%	0.94727	0.39768
E3+PF	-0.49%	1.5630	0.52449	2.070%	0.93986	0.39457
E4+PF	-3.06%	1.6030	0.53789	2.144%	0.99151	0.41625
E6+PF	**0.52%**	**1.5472**	**0.51919**	**2.042%**	**0.94064**	**0.39489**

Table 5.5: Errors of fixed and adaptive financial stochastic methods for the electricity forward price dataset.

IR(RMSE)	Fixed	Adaptive with EKF	Adaptive with PF
E1	**0.51%**	**0.60%**	**0.42%**
E2	-0.17%	-0.07%	-0.36%
E3	-1.80%	-1.62%	-0.49%
E4	-4.55%	-4.35%	-3.06%
E6	**0.18%**	**0.27%**	**0.52%**

Table 5.6: NRMSE improvement ratio of fixed and adaptive finanical stochastic models for the electricity forward price dataset.

0.46056, which improves 6.16% compared to that of the random walk model.

We have developed the adaptive financial stochastic models based on the above framework. Because they were specifically designed for electricity forward prices only, we were applied them on the UK electricity forward price. However, both the adaptive and the fixed financial stochastic models did not perform well on this dataset.

6 Analysing the noise model

This chapter investigates the noise distribution issue which is one of the techniques for improving prediction performance of the standard models. We will discuss the need to use models with Student-t noise for energy price time series and propose a novel framework for inferring parameters of Student-t models.

6.1 Why does the noise model matter?

In forecasting models, we assume that the dependent data is corrupted by noise with a zero-mean probability distribution. In Chapters 3, 4, and 5, the noise was assumed to be drawn from a Gaussian distribution. As mentioned before, this assumption is very popular in the literature either because of arguments derived from the Central Limit Theorem or just to simplify calculations. For example, the log likelihood of a Gaussian noise model is a quadratic function of the output variables. This leads to the fact that in the training process, we can easily estimate the maximum likelihood solution using optimisation algorithms. Software and frameworks for training machine learning models such as RBF,

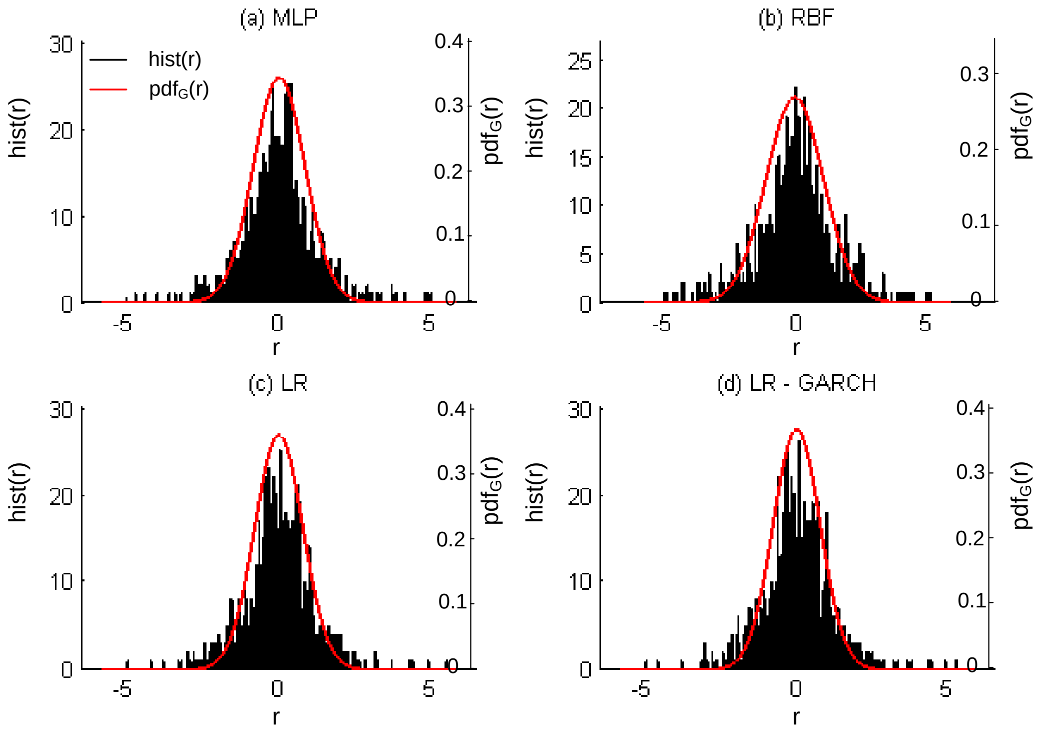


Figure 6.1: Functions $hist(r)$ are the histograms of the residuals r of the gas forward dataset and functions $pdf_G(r)$ are the probability density functions of the Gaussian distributions, which have the same means and variances of the residuals r.

MLP, and LR with Gaussian noise can be found in (Nabney, 2002). Conversely, other noise models are much less tractable. So why use non-Gaussian distributions?

In the previous chapters we used models with Gaussian noise to forecast gas forward prices in the UK energy market. In these experiments, the kurtosis[1], which is a measure of how outlier-prone a distribution is, of the residuals is between 16 and 17: the kurtosis of the Gaussian distribution is 3. Furthermore, $P(\mu - 3\sigma < r < \mu + 3\sigma) \approx 0.982$, where μ and σ are the mean and standard derivation of the residual respectively. The equivalent probability for a Gaussian distribution is 0.997; therefore, the residual distribution has heavy tails. Figure 6.1 shows the histogram of the residuals on gas forward prices and the probability density function of the Gaussian distributions which have the same mean and variance as the residuals. It shows that the residual distributions are more outlier-prone than the Gaussian distribution. It is clear that this data is not modelled well by a Gaussian distribution, as has often been noted for financial data.

As a consequence, a Student-t distribution can be considered as a good alternative to

[1] Kurtosis of a variable y is definined by $k = \sum_{t=1}^{T}(y_t - \mu)^4 / \left[(T-1)\,\sigma^4\right]$, where μ and σ are the mean and the standard deviation of y.

a Gaussian because it is a fat-tailed distribution and hence more robust. Moreover, the Student-t distribution family contains the Gaussian distribution as a special case.

There are several previous studies of inference with Student-t models. Tipping and Lawrence (2005) proposed a framework for training an RBF model with fixed basis functions. This study is a fully Bayesian treatment based on a variational approximation framework. A variational inference scheme was also used for unsupervised learning with mixture models: Bishop and Svensén (2005) presented an algorithm for automatically determining the number of components in a mixture of t-distribution using a Bayesian variational framework. In order to obtain a tractable solution, it was assumed that the latent variables are independent, and thus posterior distributions of latent variables can be factorized. This means that the algorithm does not capture correlations among the latent variables. Archambeau and Verleysen (2007) introduced a new variational Bayesian learning algorithm for Student-t mixture models, in which they removed the assumption of variable independence. Numerical experiments showed that their model had a greater robustness to outliers than Bishop and Svensén's method.

This chapter discusses Student-t noise models and presents a novel methodology to infer parameters of probabilistic models whose output noise is a Student-t distribution. This methodology for "maximum a posterior" (MAP) estimation is an extension of earlier work (Tipping and Lawrence, 2005), which is for models that are linear in parameters. Both approaches are based on a variational approximation. The main advantage of our method is that it is not limited to models whose output is linearly dependent on model parameters. On the other hand, our approach provides only MAP estimates of parameters while Tipping and Lawrence give a fully Bayesian treatment in which predictions are made by integrating out all the parameters apart from those defining the Student-t, which are optimised. Thus, although our algorithm can be applied to models that are linear in parameters, we would not expect it to outperform Tipping and Lawrence's algorithm, so our discussion focuses on the MLP.

6.2 Student-t noise models

We assume that the output data is corrupted by noise with a Student-t distribution. We are not investigating the case where the independent variables $\mathbf{x}_t$ are also noisy.

$$y_t = f(\mathbf{x}_t, \boldsymbol{\omega}) + \varepsilon_t,$$

where ε_t is a Student-t noise process, and $f(\mathbf{x}_t, \boldsymbol{\omega})$ is the output function of a forecast model, which can be an MLP, RBF, or LR. In the case of MLP models, the output is non-linear in parameters. Conversely, the output is linear in parameters when the model is LR or RBF.

The Student-t distribution can be considered as a mixture of an infinite number of zero-mean Gaussians with different variances:

$$\begin{aligned} p(\varepsilon_t|c,d) &= \textstyle\int_0^\infty p(\varepsilon_t|\beta_t)p(\beta_t|c,d)\ d\beta_t, \\ &= \frac{d^c}{\Gamma(c)}\left(\frac{1}{2\pi}\right)^{1/2}\left[d+\frac{\varepsilon_t^2}{2}\right]^{-c-1/2}\Gamma(c+1/2), \end{aligned} \tag{6.1}$$

where $p(\varepsilon_t|\beta_t)$ is a Gaussian distribution and $p(\beta_t|c,d)$ is a Gamma distribution:

$$p(\varepsilon_t|\beta_t) = \mathcal{N}(\varepsilon_t|0,\beta_t^{-1}),$$

$$p(\beta_t|c,d) = Gamma(\beta_t|c,d) = \frac{d^c}{\Gamma(c)}\beta_t^{c-1}\exp(-\beta_t d),$$

$\Gamma(c)$ is the gamma function (Abramowitz and Stegun, 1964).

The mixture weight for a given β_t is specified by the Gamma distribution $p(\beta_t|c,d)$. $\nu = 2c$ is called the "number of degrees-of-freedom" and $\sigma = \sqrt{d/c}$ is the scale parameter of the distribution. The degrees-of-freedom parameter ν can be considered as a robustness tuning parameter (Archambeau and Verleysen, 2007). When ν tends to infinity, this distribution converges to a Gaussian. Therefore, the Student-t noise model still contains the Gaussian as a special case when ν is very large. Figure 6.2 shows the density functions of Gaussian and Student-t distributions with the same mean and variance. The Student-t distribution with $\nu = 100$ is nearly overlapped by the Gaussian distribution.

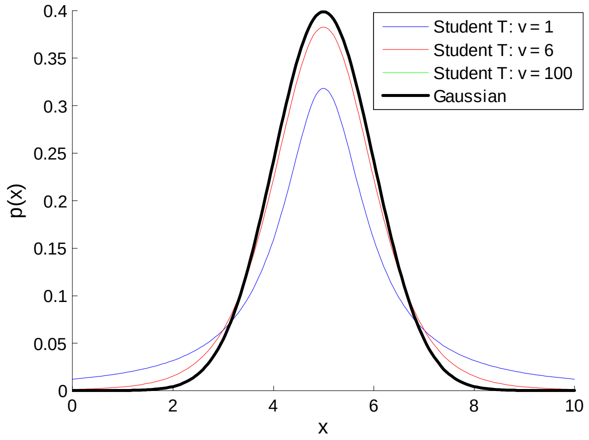


Figure 6.2: PDF of Gaussian and Student-t distributions with the same variance $\sigma^2 = 1$ and mean $\mu = 5$.

6.3 Variational inference method for the RBF and the LR

This section presents a variational inference methodology proposed by (Tipping and Lawrence, 2005) for training the RBF model with a Student-t noise distribution. In their paper, it is assumed that the basis functions $\phi_m(x)$ are fixed, $m = 1, \ldots, M$. The output function $f(x)$ is a sum of the basis functions, weighted by the vector $\boldsymbol{\omega} = (\omega_1, \omega_2, \ldots, \omega_M)$: $f(x, \boldsymbol{\omega}) = \sum_{m=1}^{M} \omega_m \phi_m(x)$.

Variational inference is a method which applies the variational method (Jordan et al., 1999) to inference problems. Variational inference trains a model by finding approximations to an intractable posterior distribution. This is done by restricting the range of functions over which the optimisation is performed. We can see in Tipping and Lawrence's method described below that in order to estimate a posterior, they introduce an approximation of the posterior and impose some restrictions on the approximation to make it analytically tractable (in particular so that certain Bayesian integrals can be performed).

The main objective of Tipping and Lawrence's methodology is to estimate the posterior probability of parameters of the model and noise distribution given a training set $\mathbf{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)\}$:

$$p(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}|\mathbf{D}, a, b, c, d) = \frac{p(\mathbf{D}|\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta})p(\boldsymbol{\beta}|c, d)p(\boldsymbol{\omega}|\boldsymbol{\alpha})p(\boldsymbol{\alpha}|a, b)}{p(\mathbf{D})}, \tag{6.2}$$

where the likelihood term $p(\mathbf{D}|\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ and prior terms $p(\boldsymbol{\beta}|c, d)$, $p(\boldsymbol{\omega}|\alpha)$, and $p(\boldsymbol{\alpha}|a, b)$

are given by:

$$p(\mathbf{D}|\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta}) = \prod_{t=1}^{T} \mathcal{N}\left(y_t|f(\boldsymbol{\omega},\mathbf{x}_t),\beta_t^{-1}\right),$$

$$p(\boldsymbol{\beta}|c,d) = \prod_{t=1}^{T} Gamma(\beta_t|c,d),$$

$$p(\boldsymbol{\omega}|\boldsymbol{\alpha}) = \prod_{m=1}^{M} \mathcal{N}(\omega_m|0,\alpha_m^{-1}),$$

$$p(\boldsymbol{\alpha}|a,b) = \prod_{m=1}^{M} Gamma(\alpha_m|a,b).$$

The hyperparameters a and b are fixed to very small values to obtain relatively flat hyperpriors over each α_m while the hyperparameters c and d are optimised, as described below.

The denominator $p(\mathbf{D})$ is not analytically tractable, thus neither is $p(\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta}|\mathbf{D},a,b,c,d)$. Tipping and Lawrence (2005) proposed an alternative approach, using variational inference, to approximate the posterior of $\boldsymbol{\omega}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$. The variational inference involves the introduction of a distribution $q(\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta})$ which provides an approximation to the true posterior distribution $p(\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta}|\mathbf{D})$. To approximate $p(\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta}|\mathbf{D})$, we consider the following decomposition of $\log p(\mathbf{D})$:

$$\begin{aligned}\log p(\mathbf{D}) &= \int q(\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta}) \log\left\{\frac{p\left(\mathbf{D},\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta}\right)}{q(\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta})}\right\} d\boldsymbol{\omega}\ d\boldsymbol{\alpha}\ d\boldsymbol{\beta} \\ &\quad -\int q(\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta}) \log\left\{\frac{p\left(\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta}|\mathbf{D}\right)}{q(\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta})}\right\} d\boldsymbol{\omega}\ d\boldsymbol{\alpha}\ d\boldsymbol{\beta} \\ &= \mathcal{L}(q(\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta})) + KL(q(\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta})||p(\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta}|\mathbf{D})). \end{aligned} \tag{6.3}$$

The second term in Equation (6.3) is the Kullback-Leibler divergence (KL) between the approximating distribution $q(\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta})$ and the true posterior $p(\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta}|\mathbf{D})$. Because the Kullback-Leibler divergence $KL(q(\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta})||p(\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta}|\mathbf{D})) \geqslant 0$, the first term is a lower bound on $\log p(\mathbf{D})$. The good point of the above decomposition is that through a suitable choice for the form of $q(\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta})$, we can analytically track the lower bound $\mathcal{L}(q(\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta}))$ and the approximation distribution $q(\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta})$, even though it is impossible to do that for $\log p(\mathbf{D})$ and the original posterior $p(\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta}|\mathbf{D})$. On the other hand, as $\log p(\mathbf{D})$ is independent of $q(\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta})$, maximising the lower bound with respect to $q(\boldsymbol{\omega},\boldsymbol{\alpha},\boldsymbol{\beta})$ is equivalent to minimising the KL divergence. This leads to the fact that we can indirectly obtain

$q(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta})$, which is an approximation of the true posterior distribution $p(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}|\mathbf{D})$, by maximising the lower bound $\mathcal{L}(q(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}))$ with respect to $q(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ because this must simultaneously minimise the KL divergence.

Selecting the form of distribution $q(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ is an important issue in the variational inference. If we accept any possible choice for $q(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta})$, the minimum of KL divergence occurs when $q(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = p(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}|\mathbf{D})$. However, this leads nowhere because we have to work with the true posterior which is intractable. We have to select a suitable form of $q(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ that is simple enough to analytically track the lower bound $\mathcal{L}(q(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}))$ and $q(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta})$, but flexible enough to provide a good approximation to the true posterior distribution (Corduneanu and Bishop, 2001). A common choice of the distribution forms is to assume that $\boldsymbol{\omega}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$ are separable: $q(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = q_{\boldsymbol{\omega}}(\boldsymbol{\omega})q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha})q_{\boldsymbol{\beta}}(\boldsymbol{\beta})$. Bishop (2006) showed that with this assumption $\mathcal{L}(q(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}))$ is maximised by

$$q_{\boldsymbol{\omega}}(\boldsymbol{\omega}) \propto \exp \langle \log p\left(\mathbf{D}, \boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}\right)\rangle_{q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha})q_{\boldsymbol{\beta}}(\boldsymbol{\beta})},$$

with symmetric expressions for $q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha})$ and $q_{\boldsymbol{\beta}}(\boldsymbol{\beta})$. In addition, the output of the model is linear in parameters, thus it is possible to analytically compute the update equations for parameters of the distributions of $q_{\boldsymbol{\omega}}(\boldsymbol{\omega})$, $q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha})$, and $q_{\boldsymbol{\beta}}(\boldsymbol{\beta})$ and the lower bound (see their detailed expressions in (Tipping and Lawrence, 2005)). In summary, variational inference is an iterative algorithm, in which each loop consists of two steps:

- Step 1: Update parameters of the distributions $q_{\boldsymbol{\omega}}(\boldsymbol{\omega})$, $q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha})$, and $q_{\boldsymbol{\beta}}(\boldsymbol{\beta})$.
- Step 2: Maximise the lower bound $\mathcal{L}(q(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}))$ with respect to c and d by scaled conjugate gradients (Møller, 1993).

This inference framework is a fully Bayesian treatment in the sense that we can estimate the full distribution of parameters. To make a prediction, we marginalise over the model parameters. The output for a new data point $\mathbf{x}^*$ is given by:

$$y^* = \int f(\mathbf{x}^*, \boldsymbol{\omega})q_{\boldsymbol{\omega}}(\boldsymbol{\omega})\ d\boldsymbol{\omega} = f(\mathbf{x}^*, \langle\boldsymbol{\omega}\rangle_{q_{\boldsymbol{\omega}}(\boldsymbol{\omega})}). \tag{6.4}$$

Although Tipping and Lawrence's paper is written in terms of a generalised linear regression model (such as RBF), their inference framework can be applied to the LR as well because like the RBF model, the output of the LR is also linear in parameters. We used this method to train both RBF and LR with Student-t noise distributions.

6.4 MAP estimation for the MLP

This section presents our method for inferring the MLP with Student-t noise. The aim of our approach is to find maximum a posterior (MAP) estimates of network and noise model parameters. MAP estimation is not a fully Bayesian treatment because it finds the optimal parameters of the models instead of finding full distributions of parameters. This is equivalent to the type-II maximum likelihood method (Berger, 1985). We will describe an algorithm for training a model with a Student-t noise model. This training framework can be used for both "non-linear in parameters" models and "linear in parameters" models.

Given a dataset $\mathbf{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)\}$, our goal is to optimise parameters of a predictive model (i.e. MLP, LR or RBF) using MAP. To simplify the notation, let $\Omega = \{\boldsymbol{\omega}, c, d, \boldsymbol{\alpha}\}$ be the set of parameters/hyperparameters of the model and noise. The posterior density of the parameters given a dataset $\mathbf{D}$ is given by

$$p(\boldsymbol{\Omega}|\mathbf{D}) = \frac{p(\mathbf{D}|\boldsymbol{\Omega})p(\boldsymbol{\Omega})}{p(\mathbf{D})},$$

where $p(\mathbf{D}|\boldsymbol{\Omega})$ is the dataset likelihood, $p(\boldsymbol{\Omega})$ is the prior, and $p(\mathbf{D})$ is the evidence. Because the denominator does not affect the MAP solution, we can ignore this term: $p(\boldsymbol{\Omega}|\mathbf{D}) \propto p(\mathbf{D}|\boldsymbol{\Omega})p(\boldsymbol{\Omega})$. The likelihood and the prior are given by

$$p(\mathbf{D}|\boldsymbol{\Omega}) = p(\mathbf{D}|\boldsymbol{\omega},c,d) = \prod_{t=1}^{T} p(y_t|\mathbf{x}_t, \boldsymbol{\omega},c,d),$$

$$p(y_t|\mathbf{x}_t, \boldsymbol{\Omega}) = \frac{d^c}{\Gamma(c)} \left(\frac{1}{2\pi}\right)^{1/2} \left[d + \frac{(y_t - f(\mathbf{x}_t, \boldsymbol{\omega}))^2}{2}\right]^{-c-1/2} \Gamma(c + 1/2),$$

$$p(\boldsymbol{\Omega}) = p(\boldsymbol{\omega}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})p(c,d).$$

The weight prior $p(\boldsymbol{\omega}|\boldsymbol{\alpha})$ distribution is Gaussian. It is helpful to generalise the hyperparameter $\boldsymbol{\alpha}$ to multiple hyperparameters $\alpha_1, \ldots, \alpha_M$ corresponding to groups of weights $\mathcal{W}_1, \ldots, \mathcal{W}_M$. In theory, we can create groupings of the weights in any way that we want. However, weights in an MLP are normally divided into four groups: first-layer weights, first-layer biases, second-layer weights, and second-layer biases. In addition, the first layer weights can be also divided into several groups: weights fanning out from an input variable are associated to a distinct group. The latter grouping approach relates to automatic relevance determination (ARD) (MacKay, 1994) and is used in our experiments. Denote group dimensions by $\mathbf{W}_1, \ldots, \mathbf{W}_M$ corresponding to the groups $\mathcal{W}_1, \ldots, \mathcal{W}_M$. Thus the dimension of $\boldsymbol{\omega}$ is $\mathbf{W} = \sum_{m=1}^{M} \mathbf{W}_m$.

$$p(\boldsymbol{\omega}|\boldsymbol{\alpha}) = \prod_{m=1}^{M} \mathcal{N}(\mathcal{W}_m|0,\alpha_m^{-1}) = \prod_{m=1}^{M} \left(\frac{\alpha_m}{2\pi}\right)^{\mathbf{W}_m/2} \exp\left[\sum_{m=1}^{M}\left(-\frac{\alpha_m}{2}\sum_{\omega\in\mathcal{W}_m}\omega^2\right)\right]. \quad (6.5)$$

There are many possible choices for the densities $p(\boldsymbol{\alpha})$ and $p(c,d)$, but for simplicity we assume that they are uniform distributions. Therefore, they will be ignored in the subsequent analysis. Hence

$$\begin{aligned}\log p(\boldsymbol{\Omega}|\mathbf{D}) &\propto \log\left[p(\mathbf{D}|\boldsymbol{\Omega})p(\boldsymbol{\Omega})\right] \\ &= Tc\log d + T\log\frac{\Gamma(c+1/2)}{\Gamma(c)} - \frac{\mathbf{W}+T}{2}\log 2\pi \\ &\quad -\left(c+\frac{1}{2}\right)\sum_{t=1}^{T}\log\left[d+\frac{(y_t - f(\mathbf{x}_t,\boldsymbol{\omega}))^2}{2}\right] \\ &\quad +\sum_{m=1}^{M}\left(\frac{\mathbf{W}_m}{2}\log\alpha_m\right) - \sum_{m=1}^{M}\left(\frac{\alpha_m}{2}\sum_{\omega\in\mathcal{W}_m}\omega^2\right), \end{aligned} \quad (6.6)$$

where T is the number of observations on the training set $\mathbf{D}$.

6.4.1 Variational approximation

The Student-t distribution of each observation y_t can be considered as a mixture of an infinite number of zero-mean Gaussians with inverse variance β_t. Let $\boldsymbol{\beta} = \{\beta_1, \beta_2, \ldots, \beta_T\}$; then

$$p(\mathbf{D}|\boldsymbol{\Omega}) = \int_0^{\infty} p(\mathbf{D},\boldsymbol{\beta}|\boldsymbol{\Omega})d\boldsymbol{\beta} = \int_0^{\infty} p(\mathbf{D}|\boldsymbol{\beta},\boldsymbol{\Omega})p(\boldsymbol{\beta}|\boldsymbol{\Omega})d\boldsymbol{\beta}, \quad (6.7)$$

where

$$p(\mathbf{D}|\boldsymbol{\beta},\boldsymbol{\Omega}) = \prod_{t=1}^{T} p(y_t|\beta_t,\boldsymbol{\Omega}) = \prod_{t=1}^{T}\left(\frac{\beta_t}{2\pi}\right)^{1/2}\exp\left\{-\frac{\beta_t}{2}(y_t - f(\mathbf{x}_t,\boldsymbol{\omega}))^2\right\}, \quad (6.8)$$

$$p(\boldsymbol{\beta}|\boldsymbol{\Omega}) = \prod_{t=1}^{T} p(\beta_t|\boldsymbol{\Omega}) = \prod_{t=1}^{T} Gamma(\beta_t|c,d) = \prod_{t=1}^{T}\frac{d^c e^{-d\beta_t}\beta_t^{c-1}}{\Gamma(c)}. \quad (6.9)$$

It is difficult to optimise $p(\mathbf{D}|\mathbf{\Omega})$ directly, but optimising $p(\mathbf{D},\boldsymbol{\beta}|\mathbf{\Omega})$ is significantly easier. We use a variational framework (Bishop, 1999) to approximate the posterior $p(\mathbf{D}|\mathbf{\Omega})p(\mathbf{\Omega})$ as follows. We introduce an approximating distribution $q(\boldsymbol{\beta})$ for $p(\boldsymbol{\beta}|\mathbf{D},\mathbf{\Omega})$: for every choice of $q(\boldsymbol{\beta})$, the following decompositions hold:

$$\begin{aligned}\log\left[p(\mathbf{D}|\mathbf{\Omega})p(\mathbf{\Omega})\right] &= \log\left[p(\mathbf{D},\boldsymbol{\beta}|\mathbf{\Omega})p(\mathbf{\Omega})\right] - \log p(\boldsymbol{\beta}|\mathbf{D},\mathbf{\Omega}),\\ \log\left[p(\mathbf{D}|\mathbf{\Omega})p(\mathbf{\Omega})\right] &= \mathcal{L}(q,\mathbf{\Omega}) + KL(q||p), \qquad (6.10)\end{aligned}$$

where

$$\begin{aligned}\mathcal{L}(q,\mathbf{\Omega}) &= \int_0^\infty q(\boldsymbol{\beta})\log\left\{\frac{p(\mathbf{D},\boldsymbol{\beta}|\mathbf{\Omega})p(\mathbf{\Omega})}{q(\boldsymbol{\beta})}\right\}d\boldsymbol{\beta}\\ &= \int_0^\infty q(\boldsymbol{\beta})\log\left[p(\mathbf{D},\boldsymbol{\beta}|\mathbf{\Omega})p(\mathbf{\Omega})\right]d\boldsymbol{\beta} - \int_0^\infty q(\boldsymbol{\beta})\log q(\boldsymbol{\beta})\ d\boldsymbol{\beta}, \qquad (6.11)\\ KL(q||p) &= -\int_0^\infty q(\boldsymbol{\beta})\log\left\{\frac{p(\boldsymbol{\beta}|\mathbf{D},\mathbf{\Omega})}{q(\boldsymbol{\beta})}\right\}d\boldsymbol{\beta}.\end{aligned}$$

In Equation (6.10), the second component $KL(q||p)$ is the Kullback-Leibler divergence between $q(\boldsymbol{\beta})$ and $p(\boldsymbol{\beta}|\mathbf{D},\mathbf{\Omega})$. It is clear that $KL(q||p) \geq 0$, with equality if and only if $q(\boldsymbol{\beta}) = p(\boldsymbol{\beta}|\mathbf{D},\mathbf{\Omega})$. Therefore, $\mathcal{L}(q,\mathbf{\Omega}) \leq \log\left[p(\mathbf{D}|\mathbf{\Omega})p(\mathbf{\Omega})\right]$, i.e. $\mathcal{L}(q,\mathbf{\Omega})$ is a lower bound on $\log\left[p(\mathbf{D}|\mathbf{\Omega})p(\mathbf{\Omega})\right]$.

6.4.2 EM for optimising the posterior

Based on the decomposition in Equation (6.10), we use an expectation maximisation (EM) algorithm[2] to maximise $p(\mathbf{D}|\mathbf{\Omega})\ p(\mathbf{\Omega})$. The two following steps are repeated:

- **E-step**: fix $\mathbf{\Omega}$ and maximise $\mathcal{L}(q,\mathbf{\Omega})$ with respect to $q(\boldsymbol{\beta})$. The lower bound can be seen as a negative Kullback-Leibler divergence between $q(\boldsymbol{\beta})$ and a distribution which is proportional to $p(\mathbf{D},\boldsymbol{\beta}|\mathbf{\Omega})p(\mathbf{\Omega})$. Thus maximising $\mathcal{L}(q,\mathbf{\Omega})$ is equivalent to minimising this Kullback-Leibler divergence. The lower bound is maximised when

[2] The EM algorithm, proposed by (Dempster et al., 1977), is a technique for finding maximum likelihood or maximum a posteriori (MAP) solutions of parameters in statistical models, where the model depends on unobserved variables (also called latent variables). The EM algorithm has two iterative steps:

- Expectation step (E-step): the distributions of the latent variables are estimated given the observed data and current estimate of the model parameters. Then the expectation of the likelihood/posteriori is evaluated using the current estimate of the latent variables.
- Maximisation step (M-step): the model parameters are computed by maximising the expected likelihood/posteriori found on the E-step. These estimates of the parameters are then used to determine the distribution of the latent variables in the next E-step.

$q(\boldsymbol{\beta}) \propto p(\mathbf{D}, \boldsymbol{\beta}|\boldsymbol{\Omega})p(\boldsymbol{\Omega}) = p(\mathbf{D}|\boldsymbol{\beta}, \boldsymbol{\Omega})p(\boldsymbol{\beta}|\boldsymbol{\Omega})p(\boldsymbol{\Omega})$. We have

$$\begin{aligned} p(\mathbf{D}|\boldsymbol{\beta}, \boldsymbol{\Omega}) &= p(\mathbf{D}|\boldsymbol{\omega}, \boldsymbol{\beta}) = \prod_{t=1}^{T} p\left(y_t|\boldsymbol{\omega}, \beta_t\right), \\ p(\boldsymbol{\beta}|\boldsymbol{\Omega}) &= p(\boldsymbol{\beta}|c, d) = \prod_{t=1}^{T} p(\beta_t|c, d), \\ p(\boldsymbol{\Omega}) &= p(\boldsymbol{\omega}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})p(c, d). \end{aligned}$$

Therefore $\log q(\boldsymbol{\beta}) \propto \log\left[p(\boldsymbol{\omega}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})p(c, d)\right] + \sum_{t=1}^{T} \log\left\{p(y_t|\boldsymbol{\omega}, \beta_t)p(\beta_t|c, d)\right\}$. Because $\log\left[p(\boldsymbol{\omega}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})p(c, d)\right]$ is independent of β, we can discarding this term:

$$\begin{aligned} \log q(\boldsymbol{\beta}) &\propto \sum_{t=1}^{T} \log\left\{p(y_t|\boldsymbol{\omega}, \beta_t)p(\beta_t|c, d)\right\}, \\ \log q(\beta_t) &\propto \log\left\{p(y_t|\boldsymbol{\omega}, \beta_t)p(\beta_t|c, d)\right\} \\ &\propto (c - \frac{1}{2}) \log \beta_t \; - \left[d + \frac{1}{2}(y_t - f(\mathbf{x}_t, \boldsymbol{\omega}))^2\right] \beta_t + const. \end{aligned}$$

The above equation shows that $\log q(\beta_t)$ is a linear combination of $\log \beta_t$ and β_t. Therefore, $q(\boldsymbol{\beta})$ is a product of Gamma distributions with the following parameters:

$$q(\boldsymbol{\beta}) = \prod_{t=1}^{T} Gamma(\beta_t|\widetilde{c}, \widetilde{d}_t), \tag{6.12}$$

$$\widetilde{c} = c + \frac{1}{2},\; \widetilde{d}_t = d + \frac{1}{2}(y_t - f(\mathbf{x}_t, \boldsymbol{\omega}))^2. \tag{6.13}$$

Note that the method in (Tipping and Lawrence, 2005) estimated posterior distributions of parameters $\boldsymbol{\omega}, \boldsymbol{\alpha}$, and $\boldsymbol{\beta}$. In order to obtain a tractable solution for these distributions, they assumed that $\boldsymbol{\omega}, \boldsymbol{\alpha}$, and $\boldsymbol{\beta}$ are a posteriori separable, such that $q(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = q_{\boldsymbol{\omega}}(\boldsymbol{\omega})q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha})q_{\boldsymbol{\beta}}(\boldsymbol{\beta})$. In our work, this assumption changes since we estimate the distribution of $\boldsymbol{\beta}$ only; the other parameters (i.e. $\boldsymbol{\omega}$ and $\boldsymbol{\alpha}$) are optimised in the M-step (which is equivalent to a delta function for each parameter vector posterior distribution).

- **M-step**: fix $q(\boldsymbol{\beta})$ using Equations (6.12) and (6.13), and maximise $\mathcal{L}(q, \boldsymbol{\Omega})$ with respect to $\boldsymbol{\Omega}$. In Equation (6.11), the first component is the expectation of a complete-data log likelihood. The second component is the entropy of $q(\boldsymbol{\beta})$ and does not depend on $\mathbf{\Omega}$. Therefore, we can ignore this component in the subsequent analysis:

$$\mathcal{L}(q, \mathbf{\Omega}) = \int_0^\infty q(\boldsymbol{\beta}) \log \{p(\mathbf{D}, \boldsymbol{\beta}|\mathbf{\Omega})p(\mathbf{\Omega})\} \, d\boldsymbol{\beta} = \langle \log \{p(\mathbf{D}, \boldsymbol{\beta}|\mathbf{\Omega})p(\mathbf{\Omega})\}\rangle_{q(\beta)} . \quad (6.14)$$

We now describe how this optimisation can be done in the following section.

6.4.3 Optimising the lower bound of log posterior

Firstly, we have to compute the lower bound $\mathcal{L}(q, \mathbf{\Omega})$.

$$\begin{aligned} \log \{p(\mathbf{D}, \boldsymbol{\beta}|\mathbf{\Omega})p(\mathbf{\Omega})\} &= \sum_{t=1}^{T} \log \{p(y_t|\boldsymbol{\omega}, \beta_t)p(\beta_t|c, d)\} + \log p(\boldsymbol{\omega}|\boldsymbol{\alpha}) \\ &\quad + \log p(\boldsymbol{\alpha}) + \log p(c, d). \end{aligned} \quad (6.15)$$

$q(\boldsymbol{\beta})$ is defined by Equation (6.12). The densities $p(\boldsymbol{\alpha})$ and $p(c, d)$ are assumed to be uniform distributions[3]. Therefore, they will be ignored in the subsequent analysis.

$$\mathcal{L}(q, \mathbf{\Omega}) = \sum_{t=1}^{T} \langle \log \{p(y_t|\boldsymbol{\omega}, \beta_t)p(\beta_t|c, d)\}\rangle_{q(\boldsymbol{\beta})} + \langle \log p(\boldsymbol{\omega}|\boldsymbol{\alpha})\rangle_{q(\boldsymbol{\beta})} .$$

From Equations (6.5), (6.8), and (6.9), we have

$$\begin{aligned} \langle \log \{p(y_t|\boldsymbol{\omega}, \beta_t)p(\beta_t|c, d)\}\rangle_{q(\boldsymbol{\beta})} &= \left(c - \frac{1}{2}\right) \langle \log \beta_t\rangle_{q(\boldsymbol{\beta})} - \frac{(y_t - f(\mathbf{x}_t, \boldsymbol{\omega}))^2}{2} \langle \beta_t\rangle_{q(\boldsymbol{\beta})} \\ &\quad -d \langle \beta_t\rangle_{q(\boldsymbol{\beta})} + c \log d - \log \Gamma(c) - \frac{1}{2} \log(2\pi), \end{aligned}$$

$$\langle \log p(\boldsymbol{\omega}|\boldsymbol{\alpha})\rangle_{q(\boldsymbol{\beta})} = \sum_{m=1}^{M} \left(\frac{\mathbf{W}_m}{2} \log \alpha_m\right) - \frac{\mathbf{W}}{2} \log (2\pi) - \sum_{m=1}^{M} \left(\frac{\alpha_m}{2} \sum_{\omega \in \mathcal{W}_m} \omega^2\right),$$

where we use the results $\langle \log \beta_t\rangle_{q(\boldsymbol{\beta})} = \langle \log \beta_t\rangle_{p(\beta_t|\widetilde{c}, \widetilde{d}_t)} = \psi(\widetilde{c}) - \log \widetilde{d}_t$ and $\langle \beta_t\rangle_{q(\boldsymbol{\beta})} = \langle \beta_t\rangle_{p(\beta_t|\widetilde{c}, \widetilde{d}_t)} = \widetilde{c}/\widetilde{d}_t$, with $\psi(\cdot)$ the "psi" or "digamma" function, defined as $\psi(x) = \partial/\partial x \left[\log \Gamma(x)\right]$ (Abramowitz and Stegun, 1964). The lower bound is given by (constant

[3] This assumption is reasonable because we have no idea that the parameters $\boldsymbol{\alpha}$, c, and d should have any particular values. This uniform distribution assumption gives equal weight to all possible values. Uniform priors $p(\boldsymbol{\alpha})$ and $p(c, d)$ are called non-informative prior (Berger, 1985).

components are ignored for simplicity):

$$\begin{aligned}\mathcal{L}(q,\mathbf{\Omega}) &= \left(c-\frac{1}{2}\right)\sum_{t=1}^{T}\left(\psi(\widetilde{c})-\log\widetilde{d}_t\right)-\frac{1}{2}\sum_{t=1}^{T}\frac{\widetilde{c}}{\widetilde{d}_t}(y_t-f(\mathbf{x}_t,\boldsymbol{\omega}))^2\\ &\quad -d\sum_{t=1}^{T}\frac{\widetilde{c}}{\widetilde{d}_t}+Tc\log d-T\log\Gamma(c)\\ &\quad +\sum_{m=1}^{M}\left(\frac{\mathbf{W}_m}{2}\log\alpha_m\right)-\sum_{m=1}^{M}\left(\frac{\alpha_m}{2}\sum_{\omega\in\mathcal{W}_m}\omega^2\right). \qquad (6.16)\end{aligned}$$

We partition the parameters into three groups $\{c,d\}$, $\{\boldsymbol{\omega}\}$, and $\{\boldsymbol{\alpha}\}$, and optimise each group in turn with the others held fixed.

Optimise $\{c,d\}$

We can use a nonlinear optimisation algorithm (e.g. scaled conjugate gradient (SCG)) to find an optimal solution for c,d. Derivatives of the lower bound with respect to c, d are given by:

$$\begin{aligned}\frac{\partial\mathcal{L}(q,\mathbf{\Omega})}{\partial c} &= \sum_{t=1}^{T}\left(\psi(\widetilde{c})-\log\widetilde{d}_t\right)+T\log d-T\psi(c),\\ \frac{\partial\mathcal{L}(q,\mathbf{\Omega})}{\partial d} &= -\sum_{t=1}^{T}\frac{\widetilde{c}}{\widetilde{d}_t}+T\frac{c}{d},\end{aligned}$$

with constraints $c,d>0$. These constraints can be enforced by a substitution: $c=\exp(\widehat{c})$, $d=\exp(\widehat{d})$. Derivatives of the lower bound with respect to $\widehat{c}$, $\widehat{d}$ are given by:

$$\begin{aligned}\frac{\partial\mathcal{L}(q,\mathbf{\Omega})}{\partial\widehat{c}} &= \frac{\partial\mathcal{L}(q,\mathbf{\Omega})}{\partial c}\frac{\partial c}{\partial\widehat{c}}=c\left[\sum_{t=1}^{T}\left(\psi(\widetilde{c})-\log\widetilde{d}_t\right)+T\log d-T\psi(c)\right],\\ \frac{\partial\mathcal{L}(q,\mathbf{\Omega})}{\partial\widehat{d}} &= \frac{\partial\mathcal{L}(q,\mathbf{\Omega})}{\partial d}\frac{\partial d}{\partial\widehat{d}}=d\left[-\sum_{t=1}^{T}\frac{\widetilde{c}}{\widetilde{d}_t}+T\frac{c}{d}\right].\end{aligned}$$

Optimise $\boldsymbol{\omega}$

Now we can consider optimisation of $\mathcal{L}(q,\mathbf{\Omega})$ with respect to $\boldsymbol{\omega}$ using a nonlinear optimisation algorithm such as SCG. The relevant partial derivative of the lower bound $\mathcal{L}(q,\mathbf{\Omega})$ is given by:

$$\begin{aligned}\frac{\partial\mathcal{L}(q,\mathbf{\Omega})}{\partial\omega_i} &= -\frac{1}{2}\frac{\partial}{\partial\omega_i}\left[\sum_{t=1}^{T}\frac{\widetilde{c}}{\widetilde{d}_t}(y_t-f(\mathbf{x}_t,\boldsymbol{\omega}))^2\right]-\widehat{\alpha}_i\omega_i\\ &= -\sum_{t=1}^{T}\left\{\frac{\widetilde{c}}{\widetilde{d}_t}\frac{\partial}{\partial\omega_i}\left[\frac{1}{2}(y_t-f(\mathbf{x}_t,\boldsymbol{\omega}))^2\right]\right\}-\widehat{\alpha}_i\omega_i, \qquad (6.17)\end{aligned}$$

where $\widehat{\boldsymbol{\alpha}} = [\widehat{\alpha}_1, \widehat{\alpha}_2, \ldots, \widehat{\alpha}_{\mathbf{W}}]$, $\widehat{\alpha}_i = \alpha_m$ if $i \in \mathcal{W}_m$, $i = 1, \ldots, \mathbf{W}$ and $m = 1, \ldots, M$. The term $\partial/\partial\omega_i \left[(y_t - f(\mathbf{x}_t, \boldsymbol{\omega}))^2/2\right]$ is the derivative of the mean square error function (NMSE) for models with Gaussian noise. Equations for this derivative are presented in (Nabney, 2002).

Optimise $\boldsymbol{\alpha}$

Our objective is to estimate the most probable value of $\boldsymbol{\alpha}$, in other words we maximise $p(\boldsymbol{\alpha}|\mathbf{D})$. The following procedure is derived from the standard evidence procedure (MacKay, 1992). The main difference is that the scalar hyperparameter β in the standard evidence procedure is replaced by a T-dimensional vector $\boldsymbol{\eta}$, to be derived below (T is number of data points in the training set.). This vector is fixed and defined by Equation (6.13) while β in the standard evidence procedure is optimised simultaneously with $\boldsymbol{\alpha}$.

$$p(\boldsymbol{\alpha}|\mathbf{D}) = \frac{p(\mathbf{D}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})}{p(\mathbf{D})}.$$

The distribution $p(\mathbf{D}|\boldsymbol{\alpha})$ is called the evidence for $\boldsymbol{\alpha}$ (MacKay, 1992). Because the denominator does not affect the optimisation solution and $p(\boldsymbol{\alpha})$ is assumed to be uniform, these terms are ignored in the subsequent analysis. This means that we have to maximise the evidence $p(\mathbf{D}|\boldsymbol{\alpha})$ with respect to $\boldsymbol{\alpha}$. Firstly, we have to compute $p(\mathbf{D}|\boldsymbol{\alpha})$.

$$p(\mathbf{D}|\boldsymbol{\alpha}) = \int_{-\infty}^{\infty} p(\mathbf{D}|\boldsymbol{\omega})p(\boldsymbol{\omega}|\boldsymbol{\alpha})\ d\boldsymbol{\omega}. \tag{6.18}$$

In the E-step, $\mathcal{L}(q, \boldsymbol{\Omega})$ is maximised with respect to $q(\boldsymbol{\beta})$, in other word we minimise $KL(q||p)$ with respect to $q(\boldsymbol{\beta})$. In this case $KL(q||p)$ is close to 0, thus $\log\left[p(\mathbf{D}|\boldsymbol{\Omega})p(\boldsymbol{\Omega})\right] \approx \mathcal{L}(q, \boldsymbol{\Omega})$. Therefore, $\log p(\mathbf{D}|\boldsymbol{\Omega})$ can be defined by Equation (6.16) without the last two terms (which are derived from the component $\log p(\boldsymbol{\Omega})$). Ignoring the components which are independent of $\boldsymbol{\omega}$, we obtain:

$$\log p(\mathbf{D}|\boldsymbol{\omega}) = -\frac{1}{2}\sum_{t=1}^{T} \frac{\widetilde{c}}{\widetilde{d}_t}(y_t - f(\mathbf{x}_t, \boldsymbol{\omega}))^2 + const. \tag{6.19}$$

Substitute Equations (6.5) and (6.19) to (6.18), we have:

$$p(\mathbf{D}|\boldsymbol{\alpha}) \propto \prod_{m=1}^{M} \left(\frac{\alpha_m}{2\pi}\right)^{\mathbf{W}_m/2} \int_{-\infty}^{\infty} \exp\left(-S(\boldsymbol{\omega})\right) d\boldsymbol{\omega}, \tag{6.20}$$

where

$$S(\boldsymbol{\omega}) = \boldsymbol{\eta}' E_{\mathbf{D}}(\boldsymbol{\omega}) + \boldsymbol{\alpha}' E_w(\boldsymbol{\omega}), \tag{6.21}$$

where $\boldsymbol{\eta}$, $\boldsymbol{\alpha}$, $E_{\mathbf{D}}(\boldsymbol{\omega})$, and $E_w(\boldsymbol{\omega})$ are column vectors, $\boldsymbol{\eta}'$ and $\boldsymbol{\alpha}'$ are the transposes of $\boldsymbol{\eta}$ and $\boldsymbol{\alpha}$ respectively, and $\boldsymbol{\eta}' E_{\mathbf{D}}(\boldsymbol{\omega})$ are the inner product of $\boldsymbol{\eta}$ and $E_{\mathbf{D}}(\boldsymbol{\omega})$:

$$E_{\mathbf{D}}(\boldsymbol{\omega}) = \left[E^1_{\mathbf{D}}(\boldsymbol{\omega}), \ldots, E^T_{\mathbf{D}}(\boldsymbol{\omega})\right]', \; E^t_{\mathbf{D}}(\boldsymbol{\omega}) = \frac{1}{2}\left(y_t - f(\boldsymbol{\omega}, \mathbf{x}_t)\right)^2,$$

$$\boldsymbol{\eta} = \left[\frac{\widetilde{c}}{\widetilde{d}_1}, \ldots, \frac{\widetilde{c}}{\widetilde{d}_T}\right]',$$

$$E_w(\boldsymbol{\omega}) = \left[E^1_w(\boldsymbol{\omega}), \ldots, E^M_w(\boldsymbol{\omega})\right]', \; E^m_w(\boldsymbol{\omega}) = \frac{1}{2}\sum_{\omega \in \mathcal{W}_m} \omega^2,$$

$$\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_M]'.$$

Note that in the equation of the overall error function $S(\boldsymbol{\omega})$, the scalar hyperparameter β in the standard evidence procedure is replaced by a T-dimensional vector $\boldsymbol{\eta}$, and the hyperparameter $\boldsymbol{\alpha}$ is generalised to multiple hyperparameters $\alpha_1, \ldots, \alpha_M$.

To evaluate the integral in Equation (6.20), we assume that $\boldsymbol{\omega}_{MP}$ is a local minimum of $S(\boldsymbol{\omega})$ (of course, it is the local maximum of lower bound $\mathcal{L}(q, \boldsymbol{\Omega})$ as well). $S(\boldsymbol{\omega})$ is approximated using a second Taylor series expansion:

$$S(\boldsymbol{\omega}) \approx S(\boldsymbol{\omega}_{MP}) + \frac{1}{2}\Delta\boldsymbol{\omega}' \mathbf{A} \Delta\boldsymbol{\omega},$$

where $\Delta\boldsymbol{\omega} = (\boldsymbol{\omega} - \boldsymbol{\omega}_{MP})$. There is no first-order term because $\partial S(\boldsymbol{\omega}_{MP})/\partial\omega_i = 0$ for all weights. The matrix $\mathbf{A}$ is the Hessian of the overall error function:

$$\mathbf{A} = \sum_{t=1}^{T} \boldsymbol{\eta}_t \nabla\nabla E^t_{\mathbf{D}}(\boldsymbol{\omega}_{MP}) + \operatorname{diag}(\widehat{\boldsymbol{\alpha}}),$$

where $\boldsymbol{\eta}_t = \widetilde{c}/\widetilde{d}_t$, $\widehat{\boldsymbol{\alpha}}$ is a $\mathbf{W}$-dimensional vector: $\widehat{\boldsymbol{\alpha}} = [\widehat{\alpha}_1, \widehat{\alpha}_2, \ldots, \widehat{\alpha}_{\mathbf{W}}]$, $\widehat{\alpha}_i = \alpha_m$ if $i \in \mathcal{W}_m$, $i = 1, \ldots, \mathbf{W}$ and $m = 1, \ldots, M$. $\operatorname{diag}(\widehat{\boldsymbol{\alpha}})$ is a diagonal matrix with the elements of $\widehat{\boldsymbol{\alpha}}$ on the main diagonal. We have:

$$\int_{-\infty}^{\infty} \exp\left[S(\boldsymbol{\omega}_{MP}) - S(\boldsymbol{\omega})\right] d\boldsymbol{\omega} = \int_{-\infty}^{\infty} \exp\left[-\frac{1}{2}\Delta\boldsymbol{\omega}' \mathbf{A} \Delta\boldsymbol{\omega}\right] d\boldsymbol{\omega} = \frac{(2\pi)^{\mathbf{W}/2}}{\|\mathbf{A}\|^{1/2}}.$$

Therefore

$$\int_{-\infty}^{\infty} \exp\left[-S(\boldsymbol{\omega})\right] d\boldsymbol{\omega} = \frac{(2\pi)^{\mathbf{W}/2}}{\|\mathbf{A}\|^{1/2}} \exp\left[-S(\boldsymbol{\omega}_{MP})\right]. \tag{6.22}$$

From Equations (6.20), (6.22), and (6.21), we have

$$\log p(\mathbf{D}|\boldsymbol{\alpha}) \propto \sum_{m=1}^{M} \left(\frac{\mathbf{W}_m}{2} \log \alpha_m \right) - \frac{1}{2} \log \|\mathbf{A}\| - \boldsymbol{\eta}' E_{\mathbf{D}}(\boldsymbol{\omega}_{MP}) - \boldsymbol{\alpha}' E_w(\boldsymbol{\omega}_{MP}).$$

Let us return to our main objective, which is to optimise $\log p(\mathbf{D}|\boldsymbol{\alpha})$. The first step is to compute its partial derivative with respect to $\boldsymbol{\alpha}$. The most difficult term is the log of the matrix determinant $\|\mathbf{A}\|$. Let $\lambda_1, \ldots, \lambda_{\mathbf{W}}$ be the eigenvalues of the data Hessian $H = \sum_{t=1}^{T} \boldsymbol{\eta}_t \nabla\nabla E_{\mathbf{D}}^t(\boldsymbol{\omega}_{MP})$. Then $\mathbf{A}$ has eigenvalues $\lambda_1 + \widehat{\alpha}_1, \ldots, \lambda_{\mathbf{W}} + \widehat{\alpha}_{\mathbf{W}}$, and

$$\begin{aligned} \frac{\partial}{\partial \alpha_m} \ln \|\mathbf{A}\| &= \frac{\partial}{\partial \alpha_m} \ln \left(\prod_{i=1}^{\mathbf{W}} (\lambda_i + \widehat{\alpha}_i) \right) = \frac{\partial}{\partial \alpha_m} \sum_{i=1}^{\mathbf{W}} \ln(\lambda_i + \widehat{\alpha}_i) \\ &= \sum_{i \in \mathcal{W}_m} \frac{1}{\lambda_i + \alpha_m} = \sum_{i \in \mathcal{W}_m} \left(A^{-1}\right)_{ii}, \; m = 1, \ldots, M. \end{aligned}$$

The derivative of the log evidence with respect to α_m is:

$$\frac{\partial}{\partial \alpha_m} \log p(\mathbf{D}|\boldsymbol{\alpha}) = -E_w^m(\boldsymbol{\omega}_{MP}) - \frac{1}{2} \sum_{i \in \mathcal{W}_m} \frac{1}{\lambda_i + \alpha_m} + \frac{\mathbf{W}_m}{2\alpha_m}.$$

Equating this to zero and rearranging give an implicit equation for α_m

$$\alpha_m = \frac{\gamma_m}{2E_w^m(\boldsymbol{\omega}_{MP})}, \; m = 1, \ldots, M, \tag{6.23}$$

where

$$\gamma_m = \sum_{i \in \mathcal{W}_m} \frac{\lambda_i}{\lambda_i + \alpha_m}, \tag{6.24}$$

is a measure of the *number of well-determined* parameters; see section 10.4 in (Bishop, 1995).

6.4.4 Summary of training process

1. Chose initial values for $\widehat{c}$, $\widehat{d}$, and $\boldsymbol{\omega}$.

2. Update parameters of distribution $q(\boldsymbol{\beta})$ using Equations (6.12) and (6.13).

	Method in Tipping and Lawrence (2005)	**Our proposed method**
1	Fully Bayesian treatment. The prediction is given by: $y^* = \int f(x^*, \boldsymbol{\omega}) q_{\boldsymbol{\omega}}(\boldsymbol{\omega}) d\boldsymbol{\omega}$.	Not fully Bayesian treatment. The prediction is $y^* = f(x^*, \boldsymbol{\omega}_{MP})$.
2	Can be applied to LR, RBF.	Can be applied to LR, RBF, MLP.
3	It is assumed that $(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ are independent: $q(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = q_{\boldsymbol{\omega}}(\boldsymbol{\omega}) q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}) q_{\boldsymbol{\beta}}(\boldsymbol{\beta})$. In fact, they are dependent: $p(\boldsymbol{\omega}\|\boldsymbol{\alpha}) = \prod_{m=1}^{M} N(\omega_m\|0, \alpha_m^{-1})$.	This assumption changes since in E-step we estimate the distribution of $\boldsymbol{\beta}$ only; the other parameters (i.e. $\boldsymbol{\omega}$ and $\boldsymbol{\alpha}$) are optimised in the M-step (which is equivalent to a delta function for each parameter vector posterior distribution).
4	Less computationally expensive, to be showed in next section.	More computationally expensive.

Table 6.1: Comparing our proposed method and the variational inference method of Tipping and Lawrence for Student-t models.

3. Optimise the lower bound $\mathcal{L}(q, \boldsymbol{\Omega})$ w.r.t $\{\boldsymbol{\omega},\ \widehat{c},\ \widehat{d},\ \boldsymbol{\alpha}\}$: partition these parameters into three groups $\left\{\widehat{c}, \widehat{d}\right\}$, $\{\boldsymbol{\omega}\}$, and $\{\boldsymbol{\alpha}\}$, and optimise each group with the others held fixed:

   (a) Optimise $\widehat{c}$, $\widehat{d}$ using scaled conjugate gradient.

   (b) Optimise $\boldsymbol{\omega}$ using scaled conjugate gradient.

   (c) Optimise $\boldsymbol{\alpha}$ using Equation (6.23).

   (d) Repeat steps (a), (b) and (c) until convergence.

4. Repeat steps 2 and 3 until convergence.

We chose to terminate when either none of the changes at each update to $\boldsymbol{\omega}$, or $\log \alpha_m$ were greater than some threshold, here 10^{-6}, or a maximum number of iterations have been exceeded, depending on whichever occurs first.

Table 6.1 summaries the comparisons of two inference methods for machine learning models: our proposed method and the variational inference (Tipping and Lawrence, 2005).

6.5 Experimental results

We tested the Student-t and Gaussian models on two forecasting tasks. The first is on a synthetic dataset which is similar to that in (Tipping and Lawrence, 2005). The second is a real life application to forecast forward gas prices in the UK market.

6.5.1 Results on a synthetic data

We generated a dataset from the function $\mathrm{sinc}(x) = (\sin(x))/x$ with additive Student-t noise, the target y of the dataset is defined by

$$y = \mathrm{sinc}(x) + \varepsilon, \tag{6.25}$$

where x is the input of the dataset and ε is the additive noise drawn from a zero-mean Student-t distribution with one degree of freedom ($\nu = 1$) and scale parameter $\sigma = 0.02$. The dataset includes a training set (100 points at equally spaced intervals in $[-10, 10]$) and a test set (80 equally spaced noise-free points in $[-10, 10]$).

We compared the prediction performance of four models: Gaussian MLP, Student-t MLP, Gaussian RBF, and Student-t RBF. The Gaussian MLP/RBF models were trained with the algorithms which were presented in Sections 3.3.2 and 3.3.3 on pages 47 and 48. The Student-t MLP was trained by our proposed algorithm and the Student-t RBF was trained with Tipping and Lawrence's algorithm. The Gaussian and Student-t RBF models had 12 and 11 basis functions respectively and their centres were equally spaced in $[-10, 10]$. We used thin plate spline basis functions $\phi_j(r_j) = r_j^2 \log(r_j)$, where $r_j = \|x - \mu_j\|$. Both Gaussian and Student-t MLP models had six hidden units and tanh activation functions. The numbers of basis functions in the RBF models and the numbers of hidden units in the MLP models were selected by 10-fold cross-validation.

Figure 6.3(a) shows the development of the log posterior $\log p(\mathbf{\Omega}|\mathbf{D})$ in Equation (6.6) on page 125 (ignoring the constant terms) during training of the Student-t MLP, indicating that our algorithm converges. Figure 6.3(b) shows the inferred noise distribution of Student-t and Gaussian MLP, compared with the true additive noise. The inferred noise distribution of the Student-t MLP is close to the real noise while that of the Gaussian MLP model is far from the real noise. This implies that the Student-t MLP model is capable of successfully learning noise parameters.

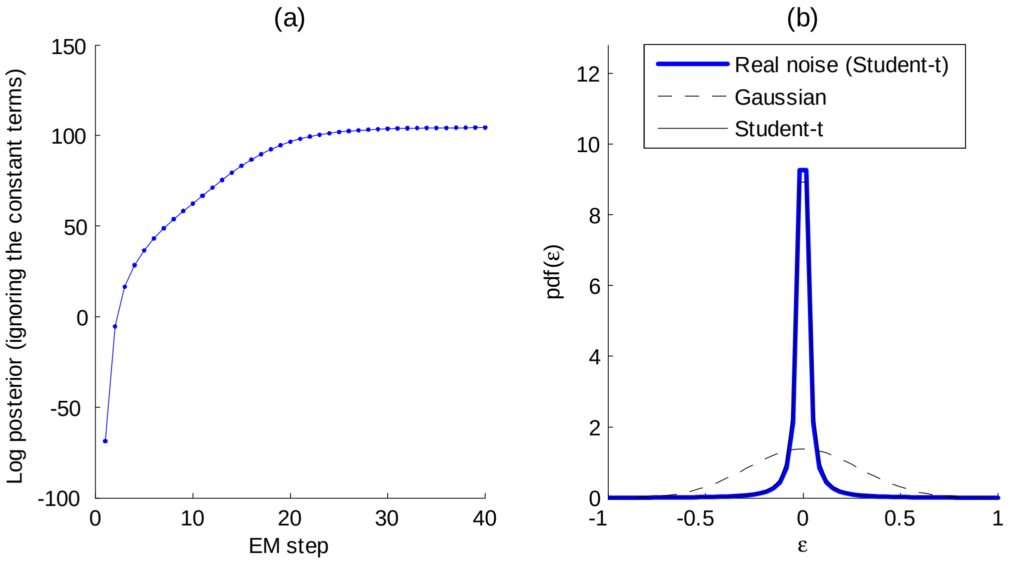


Figure 6.3: Results on synthetic dataset. (a) Log posterior $\log p(\mathbf{\Omega}|\mathbf{D})$ (ignoring the constant terms) in training the Student-t MLP model. (b) The inferred noise distributions in the Student-t MLP and Gaussian MLP models, and the true noise distribution.

Models	**RMSE**	**NRMSE**	**MAPE**	**MAE**	**NMAE**	**Running time (s)**
Gaussian MLP	0.12495	0.34716	129.59%	0.09964	0.35140	1.4305
Student-*t* MLP	**0.01336**	**0.03713**	**18.77%**	**0.01024**	**0.03610**	**597.0530**
Gaussian RBF	0.10909	0.30310	106.28%	0.07051	0.24868	0.1016
Student-*t* RBF	**0.02069**	**0.05748**	**24.48%**	**0.01687**	**0.05948**	**13.9711**

Table 6.2: Errors and running time of Student-t/Gaussian methods for the synthetic dataset.

Figure 6.4 shows prediction results of the four models. In both MLP and RBF cases, models with Student-t noise outperform Gaussian noise. Table 6.2 provides prediction accuracy information, averaged over 20 trials. The table shows that the Student-t noise models are significantly better than Gaussian models. For example, the RMSE of the Student-t MLP model is 0.01336 while the equivalent value for the Gaussian MLP model is 0.12495. This proves the robustness to outliers of Student-t models.

The biggest disadvantage of our presented method is that it is computationally expensive. The average training time for the Student-t MLP model for this case study was 597 (seconds), which is much longer than the others. (We ran experiments with code written in Matlab on a computer with Dual Core 1.66GHz CPU, RAM 1.5GB).

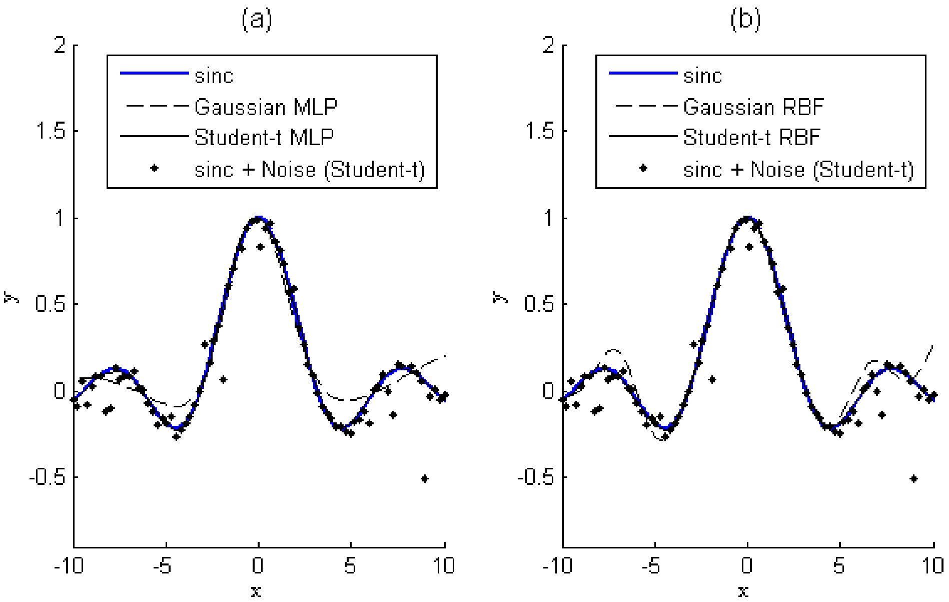


Figure 6.4: Synthetic dataset. (a) Data and predictions of the MLP models. (b) Data and predictions of the RBF models.

6.5.2 Results on the gas forward price dataset

Table 6.3 shows average errors for all 24 sub-datasets of MLP, LR, RBF, and LR-GARCH with Student-t and Gaussian models. The Student-t LR-GARCH is trained by maximum log likelihood (see Appendix F, page 179). The numbers of hidden units for Gaussian and Student-t MLP models are 8. The numbers of basis functions for Gaussian and Student-t RBF models are 30 and 25, respectively. We used a 10-fold cross-validation to select these numbers.

The table shows that the Student-t models generally outperform the Gaussian ones. Especially, the improvement ratio of RMSE of Student-t MLP is 8.72% while the equivalent quantity of Gaussian MLP is only 2.97%. This proves that the Student-t models are more robust to outliers. It is superior to Gaussian models even in this real dataset where the noise is not expected to be an exact Student-t distribution.

We can investigate in more detail the performance of the Student-t MLP model. Figure 6.5 shows the IR_{RMSE} of the Student-t/Gaussian MLP models for the gas contract sub-datasets. The Student-t MLP model generally outperforms the Gaussian MLP model, especially on the sub-datasets where the Gaussian MLP did not work well.

Figure 6.6 shows optimal values of parameters c, d, and ν of the Student-t MLP models for 24 gas forward price sub-datasets. Remember that $\nu = 2c$ is the number of degrees-of-

Models	IR(RMSE)	RMSE	NRMSE	MAPE	MAE	NMAE
Benchmark	0.00%	1.11862	0.48980	2.31%	0.84562	0.45182
Gaussian LR	3.17%	1.08295	0.47735	2.26%	0.83577	0.44283
Student-*t* LR	**7.15%**	**1.03584**	**0.46061**	**2.17%**	**0.80545**	**0.43034**
Gaussian LR-GARCH	3.77%	1.07378	0.47310	2.26%	0.83562	0.44281
Student-*t* LR-GARCH	4.15%	1.07374	0.47250	2.22%	0.83049	0.43942
Gaussian MLP	2.97%	1.09047	0.47941	2.27%	0.83586	0.44599
Student-*t* MLP	**8.72%**	**1.02048**	**0.44851**	**2.11%**	**0.78518**	**0.41913**
Gaussian RBF	2.15%	1.09969	0.48346	2.28%	0.84292	0.44976
Student-*t* RBF	**6.58%**	**1.04141**	**0.46063**	**2.19%**	**0.80550**	**0.43047**

Table 6.3: Average errors of Student-t/Gaussian models for the 24 gas price sub-datasets.

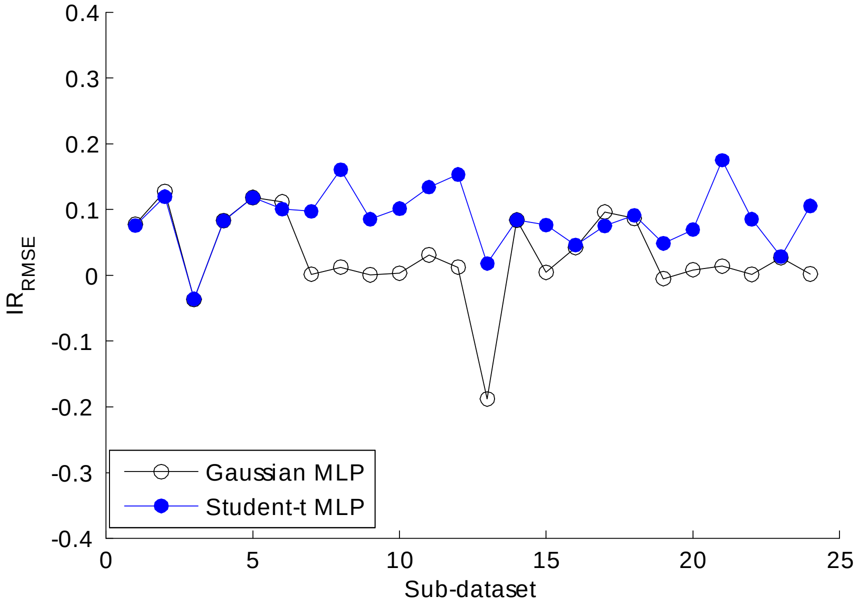


Figure 6.5: IR_{RMSE} of Student-t/Gaussian MLP models for 24 gas price sub-datasets.

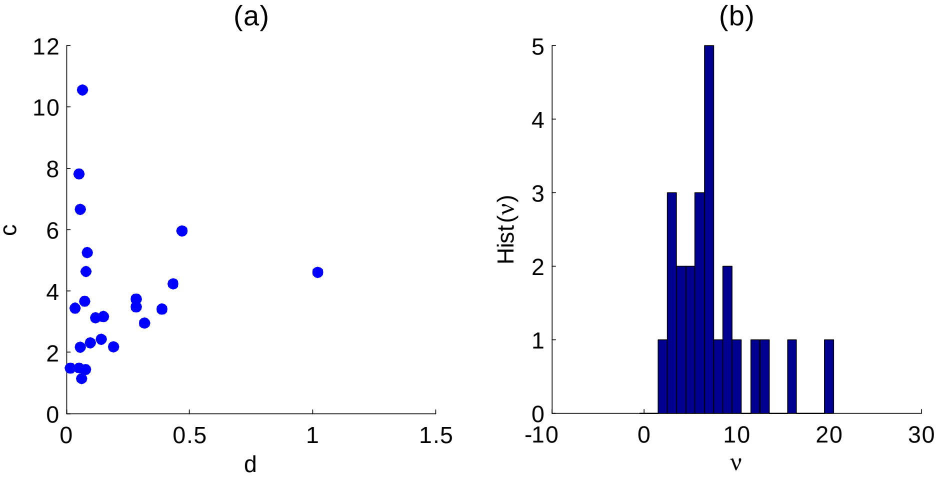


Figure 6.6: Parameters of Student-t MLP models for 24 gas forward price sub-datasets. (a) c and d. (b) Histogram of degree-of-freedom parameter ν.

freedom of the Student-t models. The degrees-of-freedom values on our experiments range from 1 to 21 with an average of 7.6. These values of ν for these sub-datasets are quite small. This indicates that the price time series are heavy tailed. Therefore one more time it is confirmed that it is worthwhile to model them by Student-t noise models.

6.6 Summary

This chapter presented Student-t noise models and a novel methodology for inferring their parameters. It was shown that our proposed method does not require the assumed prior of a linear dependence of the output on model parameters. Removing this assumption allows us to be able to apply our framework to a large range of machine learning models. In particular, we can solve the inference problem of a Student-t MLP model which cannot be solved by the previous methodologies in the literature.

It was shown experimentally that the Student-t models provide better predictions than Gaussian models in both the synthetic data (where additive noise is a Student-t distribution) and the real data of gas forward price in the UK market (where noise has a heavy-tailed distribution but would not normally be expected to be exactly a Student-t distribution). The best models for the gas forward price dataset are Student-t LR and Student-t MLP with IR_{NMSE} of 7.15% and 8.72%, respectively.

The limitation of our presented method is its computational expense. It takes a much longer time to run than Tipping and Lawrence's method. However, in some real life applications, such as day-ahead price/demand energy prediction, this running time is acceptable considering the improved results.

7 Model comparisons

Chapters 3, 4, and 5 presented three modelling improvements and their experimental results showed that these techniques had positive effects on prediction performance when they were individually combined with standard prediction models. In this chapter, we compare all the different improvements and investigate the benefit of combining all of them. Figure 7.1 summarises all the prediction models and possible improvement techniques. Note that since the EKF is limited to Gaussian noises, there are 60 different prediction frameworks by combining all prediction models and improvements.

7.1 Experiment results

7.1.1 Results on the daily electricity demand dataset

Table 7.1 shows results of predicting daily electricity demand. Because there are a large number of different forecasting frameworks (i.e. 60), we do not present results of all frameworks, but select the best. As presented in Chapter 6, the main objective of Student-t models is to capture the fat-tailed distribution of noise, so they are more appropriate

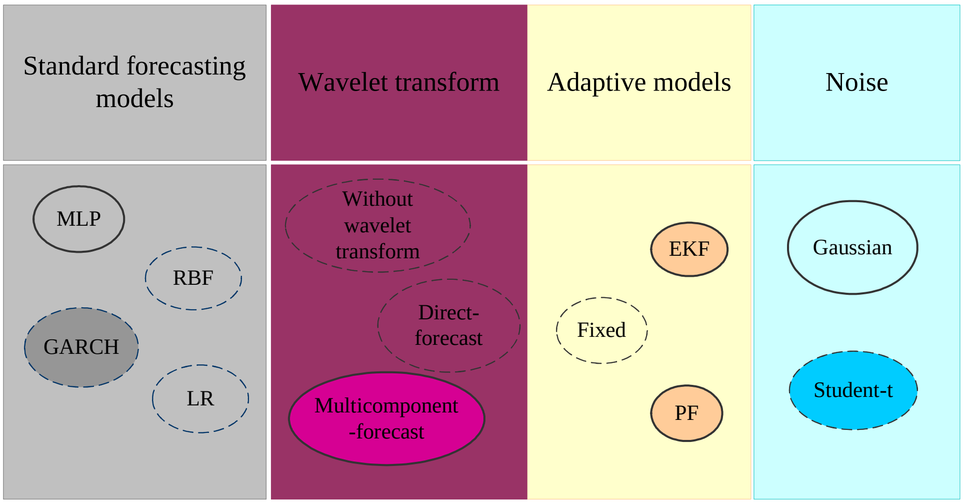


Figure 7.1: Forecasting models and improvements discussed in this thesis. The models/improvement techniques represented in solid-line ovals are the best ones in the electricity demand forecasting. The model/improvement techniques represented by filled ovals (i.e. ovals whose colours are different from the backgrounds) are the best in the gas price forecasting.

for gas forward prices rather than for electricity demand. Therefore, we presented here results of daily electricity demand using Gaussian noise models only. To develop adaptive framework for the multicomponent-forecast, we applied adaptive models for each wavelet transform component, then converted to the original target variable. Results using MLP and RBF are better than LR and LR-GARCH, and so we concentrated on results of these non-linear models.

Table 7.1 shows that wavelet transform and filters (EKF/PF) did improve performance of the prediction models. Wavelet transforms are more effective than adaptive models (see Figure 7.2 (a)). Figure 7.2 (b) shows RMSE of cumulative combination of improvement techniques. The improvement of combining techniques was better than each single improvement technique, but only by a disappointingly small amount.

7.1.2 Results on the gas forward price dataset

Table 7.2 shows average errors of forecasting gas forward prices using the prediction models and improvement techniques. We did not show the results of direct-forecast method because it was confirmed in Chapter 4 that it is not as good as multicomponent-forecast. The best models are Student-t adaptive LR-GARCH with multicomponent-forecast that have NRMSE of 0.43027 and IR$_{RMSE}$ of 12.11%.

Table 7.2 shows that all improvement techniques did enhance performance of the pre-

Models	IR(RMSE)	RMSE	NRMSE	MAPE	MAE	NMAE
Benchmark	0.00%	39365	0.36550	2.96%	29011	0.32877
LR-GARCH	45.72%	21369	0.19841	1.72%	16538	0.18742
LR	44.49%	21850	0.20252	1.76%	16915	0.19112
MLP	53.12%	18455	0.17135	1.43%	13940	0.15798
MLP+mf	**58.15%**	**16474**	**0.15362**	**1.29%**	**12403**	**0.14065**
MLP+df	50.35%	19543	0.18003	1.48%	14665	0.16619
MLP+EKF	56.14%	17266	0.16031	1.36%	13186	0.14886
MLP+EKF+mf	**59.12%**	**16092**	**0.15068**	**1.27%**	**12166**	**0.13725**
MLP+EKF+df	53.68%	18233	0.16929	1.43%	13731	0.15561
MLP+PF	55.56%	17493	0.16213	1.36%	13186	0.14900
MLP+PF+mf	**59.13%**	**16090**	**0.15067**	**1.27%**	**12141**	**0.13697**
MLP+PF+df	53.03%	18489	0.17167	1.44%	13915	0.15769
RBF	48.72%	20187	0.18743	1.63%	15589	0.17666
RBF+mf	55.08%	17681	0.16335	1.38%	13194	0.14962
RBF+df	47.64%	20612	0.19138	1.66%	15861	0.17974
RBF+EKF	49.72%	19792	0.18284	1.57%	15325	0.17315
RBF+EKF+mf	55.41%	17553	0.16221	1.36%	13187	0.14901
RBF+EKF+df	49.71%	19797	0.18300	1.57%	15215	0.17200
RBF+PF	49.68%	19810	0.18301	1.58%	15338	0.17401
RBF+PF+mf	55.42%	17550	0.16218	1.35%	13186	0.14900
RBF+PF+df	49.66%	19816	0.18322	1.57%	15227	0.17214

Table 7.1: Errors of the proposed forecasting models for the daily electricity demand dataset. "mf" and "df" stand for multicomponent-forecast and direct-forecast respectively.

Models	IR(RMSE)	RMSE	NRMSE	MAPE	MAE	NMAE
Benchmark	0.00%	1.11862	0.48980	2.31%	0.84562	0.45182
LR	3.17%	1.08295	0.47735	2.26%	0.83577	0.44283
LR+mf	9.78%	1.00299	0.44319	2.05%	0.77315	0.41183
LR+PF	4.47%	1.06917	0.47249	2.22%	0.83048	0.43797
LR+PF+mf	10.65%	0.99205	0.43843	2.01%	0.76516	0.40702
LRst	7.15%	1.03584	0.46061	2.17%	0.80545	0.43034
LRst+PF+mf	**10.84%**	**0.98826**	**0.43726**	**2.00%**	**0.75876**	**0.40415**
LR-GARCH	3.77%	1.07378	0.47310	2.26%	0.83562	0.44281
LR-GARCH+mf	9.41%	1.00614	0.44463	2.05%	0.77320	0.41407
LR-GARCH+PF	6.15%	1.04145	0.46063	2.19%	0.80551	0.43050
LR-GARCH+PF+mf	**11.42%**	**0.97918**	**0.43384**	**1.99%**	**0.75828**	**0.40388**
LR-GARCHst	4.15%	1.07374	0.47250	2.22%	0.83049	0.43942
LR-GARCHst+PF+mf	**12.11%**	**0.97440**	**0.43027**	**1.98%**	**0.75469**	**0.40268**
MLP	2.97%	1.09047	0.47941	2.27%	0.83586	0.44599
MLP+mf	8.85%	1.01426	0.44477	2.08%	0.77889	0.41563
MLP+PF	4.82%	1.06586	0.47240	2.21%	0.81153	0.43735
MLP+PF+mf	10.56%	0.99240	0.43844	2.02%	0.76516	0.40746
MLPst	8.72%	1.02048	0.44851	2.11%	0.78518	0.41913
MLPst+PF+mf	**10.93%**	**0.98537**	**0.43463**	**2.00%**	**0.75875**	**0.40392**
RBF	2.15%	1.09969	0.48346	2.28%	0.84292	0.44976
RBF+mf	8.08%	1.02283	0.44853	2.10%	0.78547	0.41914
RBF+PF	4.87%	1.06327	0.47111	2.20%	0.81149	0.43392
RBF+PF+mf	9.80%	1.00124	0.44234	2.04%	0.77286	0.41173
RBFst	6.58%	1.04141	0.46063	2.19%	0.80550	0.43047
RBFst+PF+mf	**10.09%**	**0.99692**	**0.44108**	**2.02%**	**0.76517**	**0.40757**

Table 7.2: Average errors of the forecasting models for the gas forward price dataset. "LR" and "LRst" stand for LR models with Gaussian and Student-t noise respectively. Similar notations were used for LR-GARCH, RBF, and MLP.

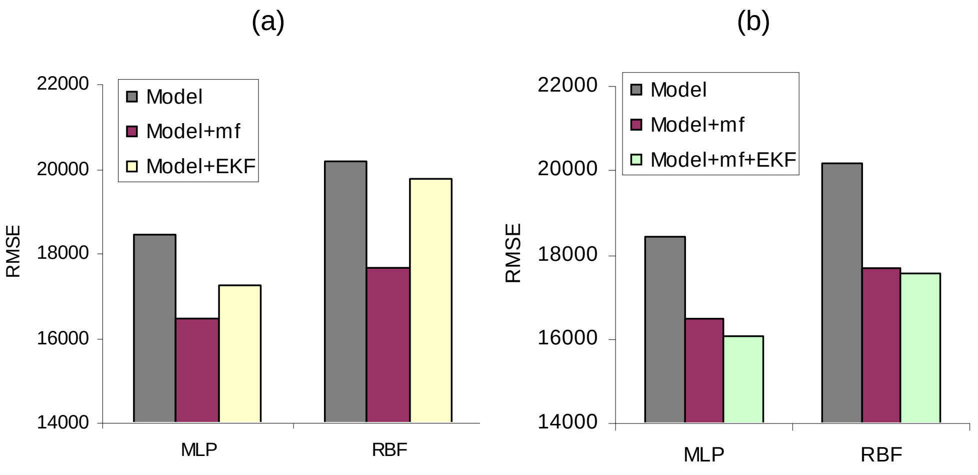


Figure 7.2: RMSE on the daily electricity demand dataset. (a) Forecasting models combined with each improvement technique individually. (b) Forecasting models cumulatively combined with the improvement techniques.

diction models. Among them, wavelet transforms generally give more benefit than adaptive models and Student-t noise (see Figure 7.3(a)). In terms of noise models, the improvement levels of Student-t noise models on the MLP, RBF, and LR were higher than on LR-GARCH: the difference of the IR_{RMSE} of Student-t MLP (RBF, LR) and Gaussian MLP (RBF, LR) is 5.75% (4.42%, 3.98%) while that of Student-t LR-GARCH and Gaussian LR-GARCH is 0.38% only. This might due to the efficiency of training algorithms: we used fully Bayesian and MAP for training Student-t LR, RBF and MLP respectively while a simple maximum likelihood was used for Student-t LR-GARCH model.

Figure 7.3(b) shows the NRMSE of cumulatively using these improvement techniques. Similar to results on daily electricity demand, the improvement of combining multiple improvement techniques was slightly better than each improvement technique individually.

7.2 Summary

This chapter compared the effectiveness of improvement techniques when they are separately combined as well as when they are cumulatively combined with the standard forecasting models. Among the three techniques, wavelet transform, adaptive models, and Student-t models, the first achieves the biggest improvement. In the electricity demand prediction, the adaptive MLP model with multicomponent-forecast is the best with IR_{RMSE} of 59.12%. The adaptive Student-t LR-GARCH models with multicomponent forecasting gets the best results in gas price forecasting. Their RMSE improves 12.11%

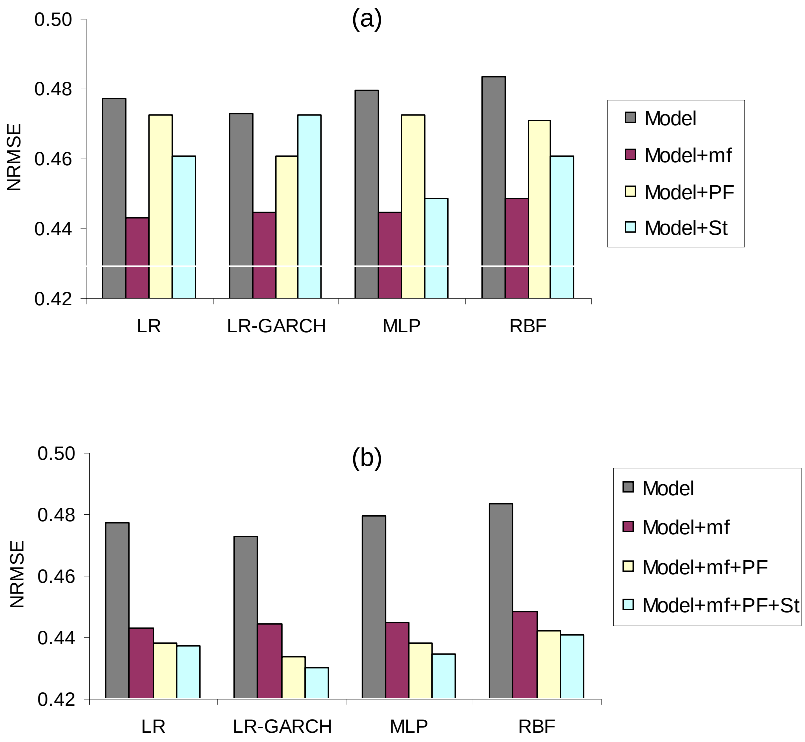


Figure 7.3: NRMSE on the gas forward price dataset. (a) Forecasting models combined with each improvement technique individually. (b) Forecasting models cumulatively combined with the improvement techniques.

compared to the RMSE of the benchmark model.

8 Conclusions and future work

8.1 Conclusions

This thesis focuses on developing three techniques to improve the performance of standard forecasting models, application to energy demand and prices prediction. These improvements are based on three aspects: (1) pre-processing data with wavelet transform, (2) re-estimating parameters with filters, and (3) broadening the range of noise models with the Student-t distribution. We also presented data analysis procedures for selecting input variables and measures for evaluating prediction models. An overview of our findings and contributions is described below.

In Chapter 3, a general two-step procedure for prediction has been described: variable selection and standard prediction models. We have investigated a range of machine learning and time series prediction models which are popular in the literature: MLP, RBF, LR, and LR-GARCH. We also studied some financial stochastic models proposed in (Benth and Koekebakker, 2008). The data was divided into two sets: a training set and a test set. The model development was based on the assumption that the evolution rule driving the data does not change; we trained the models on the training set and then used the trained models to forecast on the test set.

Chapter 3 also studied how to select input variables for prediction models. Besides historical data of the target variable (e.g. electricity demand or gas forward price), a number of exogenous variables (e.g. temperature, wind speed, day pattern, electricity supply and electricity price etc.), are also considered as potential input variables. Some pre-processing procedures (presented in Section 3.4) have been used to select the relevant input variables from these potential inputs for each forecasting model. This step is very important because it removes irrelevant inputs. These procedures not only reduce the computation time for running the forecasting models by reducing the dimension of input vectors but also improve the performance of the models by selecting only relevant inputs for training the models.

Chapter 3 has presented preliminary analysis and results of some standard forecasting models. Then the later chapters of this thesis will present different approaches to improve the performance achieved in this chapter. First, in Chapter 4, the wavelet transform was used as a pre-processing procedure. We have shown why the redundant Haar wavelet transform was chosen for prediction applications. Although combining the WT with a time series or a neural network model is not new, previous papers only used either multicomponent-forecast (a WT decomposes the target value into wavelet components, and then each component is forecast with a separate model) or direct-forecast (the components of the WT are used as input variables to a single forecast model to directly predict the target). In this thesis, we have applied both types of forecast structure and compared their prediction accuracy, which provides an answer to the question of which is better for energy datasets. The experimental results on the UK energy data showed that multicomponent-forecasts provided better results than models without RHWT and direct-forecasts.

Second, in Chapter 5, model parameters are either estimated just once or continuously updated in the testing period. We evaluated the performance of the standard forecast methods (i.e. MLP/RBF/LR/LR-GARCH) with two variations: fixed models and adaptive models. In the fixed models, parameters are fixed after training on a training set. The adaptive forecast model is a hybrid of filters (extended Kalman filter or particle filter) and the standard forecast methods, where parameters are estimated on a training set and then adapted continuously on the test set using these filters. In the adaptive models, we attempt to use observations of the time series as much as possible. Every time a new value of price/demand is observed, it is used for inferring parameters of prediction models. Among these adaptive models, the adaptive LR-GARCH model is proposed for the first time in the thesis. Moreover, we use not only the extended Kalman filter for adaptive models as earlier researches but also the particle filter. The benefits of using the particle filter are that it makes no a priori assumption of Gaussian noise and also that it is not necessary to linearise the prediction model. This means that the applicability of the PF is broader than the EKF; the PF can be applied to both Gaussian and Student-t noise models whereas the EKF is limited to Gaussian noise models only. The use of filters for adaptive models has been proved to improve the performance of prediction techniques. Adaptive models with EKF and adaptive models with PF have similar performance. In this chapter, we also presented how to combine the filters with financial models taken from (Benth and Koekebakker, 2008). Because these financial models are specific to electricity forward price only, we tested them on the electricity forward price in the UK market. However, the results showed that both fixed and adaptive financial models did not perform well. Future work for improving predicting performance of these models will be presented in the next section.

Third, in Chapter 6, we turn our attention to noise distributions of the dependent variables in the forecasting models, using either Gaussian distributions or Student-t distributions. Use of the Student-t distribution is motivated by the fact that residuals of gas price forecasts follow a fat-tailed distribution. The thesis presents a novel methodology to infer the parameters of Student-t noise models. This methodology is an extension of earlier work (Tipping and Lawrence, 2005), in which models are assumed to be linear in parameters (e.g. the RBF with fixed centres, the LR). Our proposed approach is based on a variational approximation, an evidence procedure, and an EM algorithm. The main advantage of our methodology is that it is not limited to models whose output is linearly dependent on model parameters. Therefore, our proposed training techniques broaden the range of models that can be used with a Student-t noise model. This methodology has been used to train Student-t MLP models and compared with Tipping and Lawrence's methodology for Student-t RBF/LR models and maximum likelihood for Student-t LR-GARCH models. The experimental results showed that Student-t models provided better results than Gaussian models on both a synthetic dataset (where the real noise is Student-t) and the gas forward prices in UK energy market (where noise is fat-tailed, but not exactly Student-t).

By combining these three techniques with standard prediction models, we obtain 60 different prediction frameworks. They were applied to two large datasets of real data from the UK energy markets: daily electricity demand (i.e. stationary data) and forward gas price (non-stationary). In the electricity demand forecasting task, the MLP and RBF were generally better than the LR and LR-GARCH, whereas the LR and LR-GARCH were better than the MLP and RBF at gas price prediction. The results on these datasets showed that these improvement techniques have useful effects. The forecast accuracy was significantly improved by using the WT and adaptive models. Student-t noise models outperformed Gaussian noise models in case of forecasting gas forward price, which is known to be a fat-tailed noise time series. The Student-t LR/RBF/MLP models are much better than the Gaussian LR/RBF/MLP models while Student-t LR-GARCH models only perform very slightly better than Gaussian LR-GARCH models. The reason is that we trained Student-t LR/RBF/MLP by variational inference which is a Bayesian treatment while we simply trained Student-t LR-GARCH by maximum likelihood.

We have evaluated performance when improvements were separately used in forecasting as well as when they were combined together. Of the three improvements, WT preprocessing has the greatest effect. When cumulatively using these improvement techniques, the prediction accuracy was better than each single technique, but not significantly. The best models on the electricity demand are the adaptive MLP with the multicomponent-forecast; its RMSE was 16092 which improved 59.12% comparing to the benchmark. In gas price forecast, the adaptive Student-t LR-GARCH with the multicomponent-forecast is the best with average RMSE of 0.97440, which improve 12.11% comparing to the benchmark.

8.2 Future work

There are several related research topics which could be pursued in the future to improve and extend the methods described in this work.

ARMA and ARMAX models

As discussed in Section 5.4 on page 105, the ARMA and ARMAX are good alternatives of the bias-adaptive regression models. We carried out initial tests on the ARMA model. The theory of the model and initial results of this method on electricity demand dataset are presented in Appendix B on page 169. As shown in the appendix, the ARMA model provides good performance on the electricity demand dataset. We could improve the performance of the ARMA and ARMAX models by combining them with some techniques described in this thesis, such as using the wavelet transform for pre-processing. Taylor (2003) and Taylor et al. (2010) proposed double and triple seasonal ARMA models which model the intraday, intraweek and intrayear seasonalities of half-hourly electricity demand. We would like to apply these methods to capture intraweek and intrayear cycles of our daily electricity demand. The application of the ARMA and ARMAX model to forecasting gas forward prices will also be our future work.

Improving performance of the prediction using WTs

In chapter 4, we presented the use of WTs as a pre-processing procedure. It is observed that the predictions of each individual component in the multicomponent-forecast are very good (e.g. IR$_{RMSE}$ of the components A_2, D_2, and D_1 in the LR-GARCH+mf were 50.60%, 52.60%, and 23.51%, respectively). However, the final results (i.e. the prices) of these models are not much better than the error of the benchmark (IR$_{RMSE}$ of 9.41% only). We can further study how to improve the final results. For example, instead of simply summing up all the forecast components as our current method, we can use a more complicated linear or non-linear model to derive predictions results from these forecast components.

There is a third method of using WT transform for forecasting: the WT components of the target variable are used as multiple outputs in an MLP, RBF, or LR model. This method might be an approach to overcome the issue that there exist some correlations between the residuals for the different components from the multicomponent-forecast method (see Table 4.10 on page 92 and Figure 4.11 on page 92).

In addition, because A_n is the approximation component which shows the trend of the time series, we can use forecasts of A_n for multi-step ahead prediction.

Using temperature forecast for forecasting electricity demand

Electricity demand is known to be strongly dependent on temperature. However, our experiments did not use temperature forecasts (because this data was not available during this study) but used historical real temperatures instead, i.e. at time t in order to forecast demand $\widehat{d}_{t+1}$, we used the historical real temperature τ_t. In the future, the electricity demand forecast can be improved by using a forecast of temperature at $t+1$ made at t, which should be more correlated to the demand. Moreover, currently we used average temperatures over regions in Great Britain to forecast total electricity demand. This may not be a good way to average temperature because the distribution of electricity demand over Britain highly depends on population. Therefore, instead of using average temperature over regions, it might be better if we use a weighted average temperature, in which the weights are the relative population of regions as in previous work (Taylor and Buizza, 2003).

Analysing unusual events to improve prediction performance

The preliminary analyses of the real data of electricity/gas prices in the UK showed that there are unusual events which deeply affect the evolution of prices on the day of events and the following days. For example, an extremely cold period in winter or a power station shut-down can cause a big spike in prices on the affected days and for some time afterwards. Because existing forecasting methods do not take these events into account, they might perform poorly in forecasting electricity/gas prices on the event days. We would like to analyse the impact of events on prices, study how to model those events, and find out how to use them as inputs of forecasting models.

Improving financial stochastic models by global optimisation algorithms

Another thing that may need to be studied further is the choice of the optimisation algorithm. Currently we use a local optimisation algorithm (i.e. scaled conjugate gradient) to estimate parameters in training prediction models because of its fast convergence speed and the fact that the code for this algorithm is available in the NETLAB toolbox. However, the test results on the financial stochastic models are not good. As explained in Section 5.7.3 (page 112), this might be because the local optimisation algorithm is not strong enough in this situation. The parameters found by using scaled conjugate gradient algorithm are not optimal globally, but just optimal locally. Therefore, the results on these models can be improved by using more powerful global optimisation algorithms, for example CA-ILS (Nguyen and Yao, 2008). These algorithms might help us to find a better set of optimal parameters of financial models. The main disadvantage of global optimisation algorithms is their slow convergence speed.

Predicting the variance

We have focused on predicting the mean of price/demand only as a single value for each data point. In the future, we would like to predict the variance for each prediction point. The variance allow us to know not only mean value of prediction but also the uncertainty attaching to each prediction. In LR-GARCH models where noise changes over time, computing the noise variance is straightforward because it is a function of historical noise and historical noise variance. In the other models where we now assume that their noise variances are fixed, one way to estimate the variances is to assume that the variances are not fixed but are functions of inputs, and then optimise the parameters of these functions. This methodology has been used by (Bishop and Qazaz, 1996) for Gaussian RBF models with fixed basis functions; they adopted an hierarchical Bayesian treatment to find the parameters of this variance function.

Combining results of prediction models

As mentioned before, by combining standard prediction models and various methodological improvements, we have obtained 60 different prediction frameworks. The thesis has empirically compared their performance. So if we have to give a single prediction, we can select the result of the best framework from this set based on their ranking. Alternatively, we can combine results of multiple frameworks in some way, instead of just using a single framework in isolation. Our testing on a small dataset in my MSc dissertation (Nguyen, 2007), used as the first year report of my PhD, showed that some improved performance can be obtained by combining multiple frameworks in various ways. We tested on hourly electricity demand values in Great Britain from 8^{th} January 2004 to 1^{st} January 2007 and predicted six-hour-ahead demand using MLP, RBF, LR and a weighted committee of these models. The committee had MAPE of 1.60% while that for the MLP, which was the best single model, was 1.74% only. Moreover, combining frameworks can also help to avoid overfitting. Some combining methodologies can be considered such as committees, bagging (Breiman, 1996), weighted committees, or Bayesian model average.

Representing the weekly and annual seasonality for electricity demand forecasting

The hourly electricity demand shows periodicity of a week and a year. In Section 3.6.1 on page 61, we have presented the periodic variables (i.e. days of the week and days of the year) by dummy variables which are the first harmonics of the trigonometric: the day of the week has been represented by $swd = \sin(2\pi i/7)$ and $cwd = \cos(2\pi i/7)$, where $i = 1$ to 7 correspond to Monday to Sunday respectively. We should not use the higher orders of the harmonics, for example $\sin(4\pi i/7)$ or $\sin(6\pi i/7)$ because $\sin(4\pi i/7)$ and $\sin(6\pi i/7)$ have seasonalities of a half or a third of a week, which does not exist in the electricity demand time series. However, we can consider to extend the dummy variables by using $\sin(\pi i/7)$ and $\cos(\pi i/7)$ which capture two week seasonality of the time series.

Skewed-t distributions for noise models

As noted in Chapter 6, the Gaussian distribution is not a good noise model for gas forward price. We have computed the kurtosis of the residuals of Gaussian noise models and plot histogram of these residuals. This evidence shows that the residual distribution has heavy tails. In Chapter 6, we discussed and presented solutions for Student-t noise models, which can capture the fat-tailed properties of the financial time series.

In addition, we have computed the skewness[1] of these residuals: they were between 0.19 and 0.48. Skewness is a measure of the asymmetry of the data around its mean. Because the skewness in these experiments is positive, the residuals are spread out more to the right. Therefore, we should extend the thesis on studying skewed-t distribution for noise models in future.

Extending the LR-GARCH model

In the thesis, we have used an extended version of the GARCH model, i.e. LR-GARCH. We added a linear regression term to the mean component of the GARCH model. The input vector for the linear regression component includes not only lags of target time series but also exogenous variables. Similar to this, it is a good idea to put some exogenous variables

[1] The skewness of a time series is defined as $\gamma = E\left[(x-\mu)^3\right]/\sigma^3$, where μ is the mean of x, σ is the standard deviation of x, and $E\left[t\right]$ represents the expectation of the quantity t.

to the GARCH component. In this case, we have to select a proper form of GARCH function and make some constraints on the parameters of the GARCH components (similar to Equations (3.12) and (3.13) on page 50) to ensure that the noise variance is always positive and not too large.

In addition, we can extend the adaptive LR-GARCH model by updating the parameters of the pure GARCH component. Similar to the extension mentioned on the above paragraph, we have to take into account two constraints of parameters in the GARCH component (Equations (3.12) and (3.13)) during updating the parameters of the GARCH component. It is not trivial to develop an algorithm to satisfy such constraints during the updating process.

Bibliography

R. E. Abdel-Aal and A. Z. Al-Garni. Forecasting monthly electric energy consumption in eastern Saudi Arabia using univariate time-series analysis. *Energy*, 22(11):1059–1069, 1997.

M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* Dover, New York, 1964. ISBN 0-486-61272-4.

D. Akay and M. Atak. Grey prediction with rolling mechanism for electricity demand forecasting of Turkey. *Energy*, 32:1670–1675, 2007.

N. Amjady and F. Keynia. Short-term load forecasting of power systems by combination of wavelet transform and neuro-evolutionary algorithm. *Energy*, 34(1):46–57, 2009.

A. S. Andreou, E. F. Georgopoulos, and S. D. Likothanassis. Exchange-rates forecasting: a hybrid algorithm based on genetically optimized adaptive neural networks. *Computational Economics*, 20:191–210, 2002.

C. Archambeau and M. Verleysen. Robust Bayesian clustering. *Neural Networks*, 20(1): 129–138, 2007.

S. M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

D. Benaouda, F. Murtagh, J. L. Starck, and O. Renaud. Wavelet-based nonlinear multiscale decomposition model for electricity load forecasting. *Neurocomputing*, 70:139–154, 2006.

F. E. Benth and S. Koekebakker. Stochastic modeling of financial electricity contracts. *Energy Economics*, 30:1116–1157, 2008.

J. O. Berger. *Statistical Decision Theory and Bayesian Analysis.* Springer-Verlag, 1985.

M. Bessec and J. Fouquau. The non-linear link between electricity consumption and temperature in Europe: A threshold panel approach. *Energy Economics*, 30(5):2705–2721, 2008.

C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

C. M. Bishop. Variational principal components. In *Proceedings Ninth International Conference on Artificial Neural Networks, ICANN'09*, volume 1, pages 509–514. IEE, 1999.

C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer Science Business Media, LLC, 2006.

C. M. Bishop and C. S. Qazaz. Bayesian inference of noise levels in regression. In *Proceedings 1996 International Conference on Articial Neural Networks, ICANN96, Springer*, pages 59–64, 1996.

C. M. Bishop and M. Svensén. Robust Bayesian mixture modelling. *Neurocomputing*, 64: 235–252, 2005.

T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31:307–327, 1986.

B. L. Bowerman and R.T. O'Connell. *Time series forecasting: Unified concepts and computer implementation.* Duxbury Press: Boston, 2 edition, 1987.

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

D. S. Broomhead and D. Lowe. Multi-variable functional interpolation and adaptive networks. *Complex System*, 2:321–355, 1988.

J. R. Cancelo, A. Espasa, and R. Grafe. Forecasting the electricity load from one day to one week ahead for the Spanish system operator. *International Journal of Forecasting*, 24:588–602, 2008.

A. J. Conejo, M. A. Plazas, R. Espinola, and A. B. Molina. Day-ahead electricity price forecasting using the wavelet transform and ARIMA models. *IEEE Transactions on Power Systems*, 20(2):1035–1042, 2005.

J. Contreras, R. Espinola, F.J Nogales, and A.J. Conejo. ARIMA models to predict next-day electricity prices. *IEEE transactions on Power Systems*, 18(3):1014–1020, 2003.

A. Corduneanu and C.M. Bishop. Variational Bayesian model selection for mixture distributions. In *Proceedings Eighth International Conference on Artificial Intelligence and Statistics*, 2001.

M. Crouch. Electricity interconnector policy. Technical report, Ofgem, 2010.

A. P. A. da Silva, V. H. Ferreira, and R. M. G. Velasquez. Input space to neural network based load forecasters. *International Journal of Forecasting*, 4:616–629, 2008.

P. Davies. Energy policy in the UK: how and why has the mix of fuel for electricity generation changed since 1970. Lecture notes. Aston University, 2009.

N. de Freitas, M. Niranjan, A. Gee, and J. de Freitas. Nonlinear state space estimation with neural networks and the EM algorithm. Technical report, Engineering Department, Cambridge University, 1999.

L. M. de Menezes, D. W. Bunn, and J.W Taylor. Review of guidelines for the use of combined forecasts. *European Journal of Operational Research*, 120:190–204, 2000.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 38(1):1–38, 1977.

V. Dordonnat, S. J. Koopman, M. Ooms, A. Dessertaine, and J. Collet. An hourly periodic state space model for modelling French national electricity load. *International Journal of Forecasting*, 24:566–587, 2008.

I. Drezga and S. Rahman. Input variable selection for ANN-based short-term load forecasting. *IEEE Transactions on Power Systems*, 13:1238–1244, 1998.

R. F. Engle, C. W. J. Granger, J. Rice, and A. Weiss. Semiparametric estimates of the relation between weather and electricity sales. *Journal of the American Statistical Association*, 81(394):310–320, 1986.

V. H. Ferreira and A. P. A. da Silva. Toward estimating autonomous neural network-based electric load forecasters. *IEEE Transactions on Power Systems*, 22:1554–1562., 2007.

R. Fletcher. *Practical Methods of Optimization.* New York: John Wiley, 1987.

F. Gao, X. H. Guan, X. R. Cao, and A. Papalexopoulos. Forecasting power market clearing price and quantity using neural network method. In *Proceeding of Power Engineering Summer Meet*, volume 4, pages 2183–2188, Seattle, 2000.

R. C. Garcia, J. Contreras, M. V. Akkeren, and J. B. C. Garcia. A GARCH forecasting model to predict day-ahead electricity prices. *IEEE Transactions on Power Systems*, 2 (20):867–874, 2005.

Z. Ghahramani and G. E. Hinton. Parameter estimation for linear dynamical system. Technical Report CRG-TR-96-2, Department of Computer Science, University of Toronto, Canada, 1996.

J. J. Guo and P. B. Luh. Improving market clearing price prediction by using a committee machine of neural networks. *IEEE Transactions on Power Systems*, 19(4):1867–1876, 2004.

N. Hazarika and D. Lowe. Iterative time series prediction and analysis by embedding and multiple time scale decomposition networks. *SPIE Proceedings*, 3077:94–104, 1997.

A. Henley and J. Peirson. Non-linearities in electricity demand and temperature: parametric versus non-parametric methods. *Oxford Bulletin of Economics and Statistics*, 59 (1):149–162, 1997.

H. S. Hippert, A. C. Pedreira, and R. C. Souza. Neural networks for short-term load forecasting. *IEEE Transactionson Power Systems*, 16:44–55, 2001.

J. A. Hoeting, D. Madigan, A. E. Raftery, and A. T. Volinsky. Bayesian model averaging: A tutorial. *Statistical Science*, 14:382–417, 1999.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

R. Jursa and K. Rohrig. Short-term wind power forecasting using evolutionary algorithms for the automated specification of artificial intelligence models. *International Journal of Forecasting*, 24:694–709, 2008.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, 82(Series D):35–45, 1960.

G. Li, C. C. Liu, J. Lawarree, M. Gallanti, and A. Venturini. State-of-the-art of electricity price forecasting. *2005 CIGRE/IEEE PES International Symposium*, pages 110–119, 2005.

S. Li, D. C. Wunsch, E. O'Hair, and M. G. Giesselmann. Wind turbine power estimation by neural network with Kalman filter training on SIMD parallel machine. In *International Joint Conference on Neural Networks*, volume 5, pages 3430–3434, 1999.

D. Lowe. On the use of nonlocal and nonpositive definite basis functions in radial basis function networks. In *Fourth International Conference on Artificial Neural Networks*, 1995.

D. Lowe and A. McLachlan. Modelling of nonstationary processes using radial basis function network. In *Fourth IEE International Conference on Artificial Neural Networks*, pages 300–305, 1995.

D. Lowe and A. R. Webb. Time series prediction by adaptive networks: A dynamic systems perspective. *IEE Proceedings - F*, 138(1):17–24, 1991.

D. J. C. MacKay. Bayesian method for backprop network. In E. Domany, J. L. van Hemmen, and K. Schulten, editors, *Models of Neural Networks, III*, pages 211–254. Springer, 1994.

D. J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.

P. Mandal, T. Senjyu, K. Uezato, and T. Funabashi. Several-hours-ahead electricity price and load forecasting using neural networks. *IEEE Power Engineering Society General Meeting*, 3:2146–2153, 2005.

M. Misiti, Y. Misiti, G. Oppenheim, and J. M. Poggi. *Wavelet toolbox 4: User's guide.* The MathWorks, Inc, 2008.

D. C. Montgomery, C. L. Jennings, and M. Kulahci. *Introduction to Time Series Analysis and Forecasting.* John Wiley & Sons, 2008.

J. Moral-Carcedo and J. Vict'ens-Otero. Modelling the non-linear response of Spanish electricity demand to temperature variations. *Energy Economics*, 27:477–494, 2005.

M. F. Møller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6(4):525–533, 1993.

I. T. Nabney. *NETLAB: Algorithms for Pattern Recognition.* Great Britain: Springer, 2002.

I. T. Nabney, A. McLachlan, and D. Lowe. Practical method of tracking of non-stationary time series applied to real world data. In *SPIE conference on the Applications and Science of Artificial Neural Networks II*, pages 152–163, 1996.

T. H. Nguyen. Energy demand and price prediction. Master's thesis, Aston University, 2007.

T. T. Nguyen and X. Yao. An experimental study of hybridizing cultural algorithms and local search. *International Journal of Neural Systems*, 18(1):1–17, 2008.

M. Niranjan. On data driven options prices using neural networks. In *Forecasting Financial Markets: Advances for Exchange Rate, Interest Rate and Asset Management*, pages 1–13, London, 1999.

F. J. Nogales, J. Contreras, A. J. Conejo, and R. Espínola. Forecasting next-day electricity prices by time series models. *IEEE Transactions on Power Systems*, 17(2):342–348, 2002.

A. Panagiotelis and M. Smith. Bayesian density forecasting of intraday electricity prices using multivariate skew t distributions. *International Journal of Forecasting*, 24:710–727, 2008.

S. A. Patil, R. Irwin, S. Srinivasan, S. Prasad, G. Lazarou, and J. Picone. Sequential state-space filters for speech enhancement. In *Proceedings of the IEEE Southeast Conference 2006*, pages 240–243, 2006.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 2nd edition, 1992. ISBN 0-521-43108-5.

Olivier Renaud, Jean-Luc Starck, and Fionn Murtagh. Wavelet-based combined signal filtering and prediction. *IEEE Transactions SMC, Part B*, 35:1241–1251, 2005.

M. I. Ribeiro. Kalman and extended Kalman filter: Concept, derivation and properties. Technical report, Institute for Systems and Robotics, Instituto Superior Tecnico, Portugal, Portugal, 2004.

RTE. Generation adequacy report on the eletricity demand supply balance in France. Technical report, RTE - Gestionnaire du Réseau de Transport d'Électricité, 2005.

A. K. Saha, S. Chowdhury, S. P. Chowdhury, Y. H. Song, and G. A. Taylor. Application of wavelets in power system load forecasting. In *Power Engineering Society General Meeting,IEEE*, 2006.

L. J. Soares and M. C. Medeiros. Modeling and forecasting short-term electricity load: A comparison of methods with an application to Brazilian data. *International Journal of Forecasting*, 24:630–644, 2008.

J. L. Starck and F. Murtagh. MR/finance multiresolution analysis of time series. Technical report, 2001. URL `http://thames.cs.rhul.ac.uk/~multires/doc/mrfin.pdf`.

M. Stevenson. Filtering and forecasting spot electricity price in the increasingly deregulated Australian electricity market. In *The International Institute of Forecaster Conference*, Atlanta, 2001.

J. W. Taylor. Short-term electricity demand forecasting using double seasonal exponential smoothing. *Journal of Operational Research Society*, 54:799–805, 2003.

J. W. Taylor. An evaluation of methods for very short term electricity demand forecasting using minute-by-minute British data. *International Journal of Forecasting*, 24:645–658, 2008.

J. W. Taylor and R. Buizza. Using weather ensemble predictions in electricity demand forecasting. *International Journal of Forecasting*, 19:57–70, 2003.

J. W. Taylor and D. W Bunn. Investigating improvements in the accuaracy of prediction intervals for combinations of forecasts: A simulation study. *International Journal of Forecasting*, 15:325–339, 1999.

J. W. Taylor, L. M. M. de Menezes, and P. E. McSharry. A comparison of univariate methods for forecasting electricity demand up to a day ahead. *International Journal of Forecasting*, 1-16:22, 2006.

J. W. Taylor, P. E. McSharry, and R. Buizza. Wind power density forecasting using wind ensemble predictions and time series models. *IEEE Transactions on Energy Conversion*, 2010.

M. E. Tipping and N. D. Lawrence. Variational inference for Student-t models: Robust Bayesian interpolation and generalised component analysis. *Neurocomputing*, 69:123–141, 2005.

E. Valor, V. Meneu, and V. Caselles. Daily air temperature and electricity load in Spain. *Journal of Applied Meteorology*, 40:1413–1421, 2001.

M. Wiltsher, C. Lock, and J. Collings. Electricity generation: facts and figures. Technical report, Ofgem, 2006.

H. T. Xu and T. Niimura. Short-term electricity price modeling and forecasting using wavelets and multivariable time series. In *Power Systems Conference and Exposition*, volume 1, pages 208–212, 2004.

Y. Y. Yan. Climate and residential electricity consumption in Hong Kong. *Energy*, 23(1): 17–20, 1998.

S. J. Yao, Y. H. Song, L. Z. Zhang, and X. Y. Cheng. Wavelet transform and neural networks for short-term electrical load forecasting. *Energy Conversion & Management*, 41:1975–1988, 2000.

S. Yousefi, I. Weinreichb, and D. Reinarzb. Wavelet-based prediction of oil prices. *Chaos, Solitons and Fractals*, 25:265–275, 2005.

L. Zhang and P. B. Luh. Power market clearing price prediction and confidence interval estimation with fast neural network learning. In *Proc. IEEE 2002 Power Engineering Society Winter Meeting*, volume 1, pages 268–273, 2002.

L. Zhang, P. B. Luh, and K. Kasiviswanathan. Energy clearing price prediction and confidence interval estimation with cascaded neural networks. *IEEE Transaction on Power System*, 1(18):99–105, 2003.

H. Zheng, L. Xie, and L. Z. Zhang. Electricity price forecasting based on GARCH model in deregulated market. In *Power Engineering Conference*, pages 410–416, USA, 2005.

M. Zhou, Z. Yan, Y. Ni, G. Li, and Y Nie. Electricity price forecasting with confidence-interval estimation through an extended ARIMA approach. In *IEEE Proceedings of Generation,Transmission and Distribution*, volume 153, pages 233–238, 2006.

A Financial models

In this appendix, we present details of the equations for m_t, v_t , $\partial m_t/\partial\theta$, and $\partial v_t/\partial\theta$ for the financial stochastic models in Section 3.3.5 on page 53. The stochastic models of log return $r_t(\mathcal{T}_1, \mathcal{T}_2)$ are proposed as a normally distributed random variable $\mathcal{N}(m_t, \nu_t)$ with mean and variance:

$$\begin{aligned} m_t(\mathcal{T}_1, \mathcal{T}_2) &= \int_t^{t+1} \left(\lambda\Upsilon(s, \mathcal{T}_1, \mathcal{T}_2) - \frac{1}{2}\Upsilon^2(s, \mathcal{T}_1, \mathcal{T}_2)\right) dsM, \\ \nu_t(\mathcal{T}_1, \mathcal{T}_2) &= \int_t^{t+1} \Upsilon^2\left(s, \mathcal{T}_1, \mathcal{T}_2\right)\, ds. \end{aligned}$$

Let

$$M = \int_t^{t+1} \Upsilon(s, \mathcal{T}_1, \mathcal{T}_2)\, ds,\, N = \int_t^{t+1} \Upsilon^2(s, \mathcal{T}_1, \mathcal{T}_2)\, ds,$$

then

$$\begin{aligned} m_t &= \lambda M - \frac{1}{2}N, \\ \nu_t &= N, \\ \frac{\partial m_t}{\partial\theta} &= \lambda\frac{\partial M}{\partial\theta} + M\frac{\partial\lambda}{\partial\theta} - \frac{1}{2}\frac{\partial N}{\partial\theta}, \\ \frac{\partial \nu_t}{\partial\theta} &= \frac{\partial N}{\partial\theta}. \end{aligned}$$

The following sections define equations for M, N, $\partial M/\partial\theta$, and $\partial N/\partial\theta$ for these financial stochastic models.

A.1 Model E1

$$
\begin{array}{rcl}
\theta &=& \{a, \lambda\}, \\
M &=& a, \quad N = a^2, \\
\dfrac{\partial M}{\partial a} &=& 1, \ \dfrac{\partial M}{\partial \lambda} = 0, \\
\dfrac{\partial N}{\partial a} &=& 2a, \ \dfrac{\partial N}{\partial \lambda} = 0.
\end{array}
$$

A.2 Model E2

$$
\begin{array}{rcl}
\theta &=& \{a, b, \lambda\}, \\
M &=& \dfrac{a}{(\mathcal{T}_2 - \mathcal{T}_1)} \dfrac{\left[e^b - 1\right] \left(e^{-b(\mathcal{T}_1 - t)} - e^{-b(\mathcal{T}_2 - t)}\right)}{b^2}, \\
N &=& \dfrac{a^2}{2(\mathcal{T}_2 - \mathcal{T}_1)^2} \dfrac{(e^{2b} - 1) \left[e^{-b(\mathcal{T}_1 - t)} - e^{-b(\mathcal{T}_2 - t)}\right]^2}{b^3}.
\end{array}
$$

Partial derivatives of M

$$
\begin{array}{rcl}
\dfrac{\partial M}{\partial a} &=& \dfrac{1}{(\mathcal{T}_2 - \mathcal{T}_1)} \dfrac{A_1}{A_2}, \\
\dfrac{\partial M}{\partial b} &=& \dfrac{a}{(\mathcal{T}_2 - \mathcal{T}_1)} \dfrac{A_1' A_2 - A_2' A_1}{A_2^2}, \\
\dfrac{\partial M}{\partial \lambda} &=& 0, \\
A_1 &=& \left[e^b - 1\right] (e^{-b(\mathcal{T}_1 - t)} - e^{-b(\mathcal{T}_2 - t)}), \\
A_2 &=& b^2, \\
A_1' &=& e^b (e^{-b(\mathcal{T}_1 - t)} - e^{-b(\mathcal{T}_2 - t)}) \\
 && + (e^b - 1) \left[-(\mathcal{T}_1 - t) e^{-b(\mathcal{T}_1 - t)} + (\mathcal{T}_2 - t) e^{-b(\mathcal{T}_2 - t)}\right], \\
A_2' &=& 2b.
\end{array}
$$

Partial derivatives of N

$$
\begin{array}{rcl}
\dfrac{\partial N}{\partial a} &=& \dfrac{a}{(\mathcal{T}_2 - \mathcal{T}_1)^2} \dfrac{B_1}{B_2}, \\
\dfrac{\partial N}{\partial b} &=& \dfrac{a^2}{2(\mathcal{T}_2 - \mathcal{T}_1)^2} \dfrac{B_1' B_2 - B_2' B_1}{B_2^2}, \\
\dfrac{\partial N}{\partial \lambda} &=& 0, \\
B_1 &=& (e^{2b} - 1) \left[e^{-b(\mathcal{T}_1 - t)} - e^{-b(\mathcal{T}_2 - t)}\right]^2, \\
B_2 &=& b^3, \\
B_1' &=& 2e^{2b} \left[e^{-b(\mathcal{T}_1 - t)} - e^{-b(\mathcal{T}_2 - t)}\right]^2, \\
 && + 2(e^{2b} - 1) \left[e^{-b(\mathcal{T}_1 - t)} - e^{-b(\mathcal{T}_2 - t)}\right] \left[-(\mathcal{T}_1 - t) e^{-b(\mathcal{T}_1 - t)} + (\mathcal{T}_2 - t) e^{-b(\mathcal{T}_2 - t)}\right], \\
B_2' &=& 3b^2.
\end{array}
$$

A.3 Model E3

$$
\begin{aligned}
\theta &= \{a, b, d, q, \lambda\}, \\
M &= \widehat{I}K,\ K = C_1 + C_2 D_2 + C_3 D_3, \\
N &= \widehat{I}^2 J,\ J = A_1 B_1 + A_2 B_2 + A_3 B_3 + A_4 B_4 + A_5 B_5, \\
\widehat{I} &= \frac{e^{b(t-\mathcal{T}_1)} - e^{b(t-\mathcal{T}_2)}}{b(\mathcal{T}_2 - \mathcal{T}_1)}, \\
C_1 &= \frac{a(e^b - 1)}{b}, \\
C_2 &= \frac{bd - \Omega q}{b^2 + \Omega^2}, \qquad D_2 = e^b \sin\Omega\,(t+1) - \sin\Omega t, \\
C_3 &= -\frac{bq + \Omega d}{b^2 + \Omega^2}, \quad D_3 = e^b \cos\Omega\,(t+1) - \cos\Omega t, \\
A_1 &= a^2 + \frac{d^2 + q^2}{2},\ B_1 = \frac{\left(e^{2b} - 1\right)}{2b}, \\
A_2 &= \frac{q^2 b - d^2 b + 2dq\Omega}{4\,(b^2 + \Omega^2)},\ B_2 = e^{2b}\cos 2\Omega(t+1) - \cos 2\Omega t, \\
A_3 &= \frac{q^2\Omega - d^2\Omega - 2dqb}{4\,(b^2 + \Omega^2)},\ B_3 = e^{2b}\sin 2\Omega(t+1) - \sin 2\Omega t, \\
A_4 &= \frac{-2ad\Omega - 4aqb}{4b^2 + \Omega^2},\quad B_4 = e^{2b}\cos\Omega(t+1) - \cos\Omega t.
\end{aligned}
$$

Partial derivatives of M

$$
\begin{aligned}
\frac{\partial M}{\partial a} &= \widehat{I}\frac{(e^b - 1)}{b}, \\
\frac{\partial M}{\partial b} &= K\frac{\partial \widehat{I}}{\partial b} + \widehat{I}\frac{\partial K}{\partial b}, \\
\frac{\partial M}{\partial d} &= \widehat{I}\left[\frac{b}{b^2 + \Omega^2}D_2 - \frac{\Omega}{b^2 + \Omega^2}D_3\right], \\
\frac{\partial M}{\partial q} &= \widehat{I}\left[\frac{-\Omega}{b^2 + \Omega^2}D_2 - \frac{b}{b^2 + \Omega^2}D_3\right], \\
\frac{\partial M}{\partial \lambda} &= 0, \\
\frac{\partial \widehat{I}}{\partial b} &= \frac{(t - \mathcal{T}_1)\,e^{b(t-\mathcal{T}_1)} - (t - \mathcal{T}_2)\,e^{b(t-\mathcal{T}_2)}}{b(\mathcal{T}_2 - \mathcal{T}_1)} - \frac{e^{b(t-\mathcal{T}_1)} - e^{b(t-\mathcal{T}_2)}}{b^2(\mathcal{T}_2 - \mathcal{T}_1)}, \\
\frac{\partial K}{\partial b} &= (C_1)' + (C_2)'\,D_2 + C_2\,(D_2)' + (C_3)'\,D_3 + C_3\,(D_3)', \\
\frac{\partial C_1}{\partial b} &= a\frac{be^b - e^b + 1}{b^2}, \\
\frac{\partial C_2}{\partial b} &= \frac{-db^2 + d\Omega^2 + 2qb\Omega}{(b^2 + \Omega^2)^2}, \qquad \frac{\partial D_2}{\partial b} = e^b \sin\Omega\,(t+1), \\
\frac{\partial C_3}{\partial b} &= \frac{-q\Omega^2 + qb^2 + 2db\Omega}{(b^2 + \Omega^2)^2}, \qquad \frac{\partial D_3}{\partial b} = e^b \cos\Omega\,(t+1).
\end{aligned}
$$

Partial derivatives of N

$$
\begin{aligned}
\frac{\partial N}{\partial a} &= \widehat{I}^2 \frac{\partial J}{\partial a}, \\
\frac{\partial N}{\partial b} &= \frac{\partial \widehat{I}^2}{\partial b} J + \widehat{I}^2 J', \\
\frac{\partial N}{\partial d} &= \widehat{I}^2 \frac{\partial J}{\partial d}, \\
\frac{\partial N}{\partial q} &= \widehat{I}^2 \frac{\partial J}{\partial q}, \\
\frac{\partial N}{\partial \lambda} &= 0, \\
\frac{\partial J}{\partial a} &= 2aB_1 - \frac{2d\Omega + 4qb}{4b^2 + \Omega^2} B_4 + \frac{4db - 2q\Omega}{4b^2 + \Omega^2} B_5, \\
\frac{\partial \widehat{I}^2}{\partial b} &= 2\widehat{I} \frac{\partial \widehat{I}}{\partial b}, \frac{\partial \widehat{I}}{\partial b} = \frac{(t - \mathcal{T}_1)\, e^{b(t-\mathcal{T}_1)} - (t - \mathcal{T}_2)\, e^{b(t-\mathcal{T}_2)}}{b(\mathcal{T}_2 - \mathcal{T}_1)} - \frac{e^{b(t-\mathcal{T}_1)} - e^{b(t-\mathcal{T}_2)}}{b^2(\mathcal{T}_2 - \mathcal{T}_1)}, \\
\frac{\partial J}{\partial b} &= A_1 B_1' + A_1' B_1 + A_2 B_2' + A_2' B_2 + \\
&\quad A_3 B_3' + A_3' B_3 + A_4 B_4' + A_4' B_4 + A_5 B_5' + A_5' B_5, \\
\frac{\partial A_1}{\partial b} &= 0, \\
\frac{\partial B_1}{\partial b} &= \frac{e^{2b}}{b} - \frac{\left(e^{2b} - 1\right)}{2b^2}, \\
\frac{\partial A_2}{\partial b} &= \frac{(q^2 - d^2)(\Omega^2 - b^2) - 4dqb\Omega}{4(b^2 + \Omega^2)^2}, \quad \frac{\partial B_2}{\partial b} = 2e^{2b} \cos 2\Omega(t + 1), \\
\frac{\partial A_3}{\partial b} &= \frac{dqb^2 - dq\Omega^2 - b\Omega q^2 + b\Omega d^2}{2(b^2 + \Omega^2)^2}, \quad \frac{\partial B_3}{\partial b} = 2e^{2b} \sin 2\Omega(t + 1), \\
\frac{\partial A_4}{\partial b} &= (-4a) \frac{q\Omega^2 - 4db\Omega - 4qb^2}{(4b^2 + \Omega^2)^2}, \quad \frac{\partial B_4}{\partial b} = 2e^{2b} \cos \Omega(t + 1), \\
\frac{\partial A_5}{\partial b} &= 4a \frac{d\Omega^2 + 4qb\Omega - 4db^2}{(4b^2 + \Omega^2)^2}, \quad \frac{\partial B_5}{\partial b} = 2e^{2b} \sin \Omega(t + 1), \\
\frac{\partial J}{\partial d} &= dB_1 + \frac{-2db + 2q\Omega}{4\left(b^2 + \Omega^2\right)} B_2 + \frac{-2d\Omega - 2qb}{4\left(b^2 + \Omega^2\right)} B_3 \\
&\quad + \frac{-2a\Omega}{4b^2 + \Omega^2} B_4 + \frac{4ab}{4b^2 + \Omega^2} B_5, \\
\frac{\partial J}{\partial q} &= qB_1 + \frac{qb + d\Omega}{2\left(b^2 + \Omega^2\right)} B_2 + \frac{q\Omega - db}{2\left(b^2 + \Omega^2\right)} B_3 \\
&\quad + \frac{-4ab}{4b^2 + \Omega^2} B_4 + \frac{-2a\Omega}{4b^2 + \Omega^2} B_5.
\end{aligned}
$$

A.4 Model E4

$$\begin{aligned}
\theta &= \{a, b, c, \lambda\}, \\
M &= a\left[(1-c)\widehat{I}A_1 + c\right], \\
N &= a^2 J, \\
J &= (1-c)^2\left(\widehat{I}\right)^2 A_2 + 2c\,(1-c)\,\widehat{I}A_1 + c^2, \\
A_1 &= \frac{(e^b-1)}{b}, \\
A_2 &= \frac{(e^{2b}-1)}{2b}.
\end{aligned}$$

Partial derivatives of M

$$\begin{aligned}
\frac{\partial M}{\partial a} &= (1-c)\,\widehat{I}A_1 + c, \\
\frac{\partial M}{\partial b} &= a\,(1-c)\left[A_1\frac{\partial\widehat{I}}{\partial b} + \widehat{I}\,\frac{\partial A_1}{\partial b}\right], \\
\frac{\partial M}{\partial c} &= a\left[-\widehat{I}A_1 + 1\right], \\
\frac{\partial M}{\partial \lambda} &= 0, \\
\frac{\partial A_1}{\partial b} &= \frac{be^b - e^b + 1}{b^2}, \\
\frac{\partial \widehat{I}}{\partial b} &= \frac{(t-\mathcal{T}_1)\,e^{b(t-\mathcal{T}_1)} - (t-\mathcal{T}_2)\,e^{b(t-\mathcal{T}_2)}}{b(\mathcal{T}_2-\mathcal{T}_1)} - \frac{e^{b(t-\mathcal{T}_1)} - e^{b(t-\mathcal{T}_2)}}{b^2(\mathcal{T}_2-\mathcal{T}_1)}.
\end{aligned}$$

Partial derivatives of N

$$\begin{aligned}
\frac{\partial N}{\partial a} &= 2aJ, \\
\frac{\partial N}{\partial b} &= a^2\{(1-c)^2\left[2\widehat{I}\widehat{I}'A_2 + \left(\widehat{I}\right)^2 A_2'\right] \\
&\quad +2c\,(1-c)\left[\widehat{I}'A_1 + \widehat{I}A_1'\right]\}, \\
\frac{\partial N}{\partial c} &= a^2\left[2\,(c-1)\left(\widehat{I}\right)^2 A_2 + 2\,(1-2c)\,\widehat{I}A_1 + 2c\right], \\
\frac{\partial N}{\partial \lambda} &= 0, \\
\frac{\partial A_1}{\partial b} &= \frac{be^b - e^b + 1}{b^2}, \\
\frac{\partial A_2}{\partial b} &= \frac{2be^{2b} - e^{2b} + 1}{2b^2}, \\
\frac{\partial \widehat{I}}{\partial b} &= \frac{(t-\mathcal{T}_1)\,e^{b(t-\mathcal{T}_1)} - (t-\mathcal{T}_2)\,e^{b(t-\mathcal{T}_2)}}{b(\mathcal{T}_2-\mathcal{T}_1)} - \frac{e^{b(t-\mathcal{T}_1)} - e^{b(t-\mathcal{T}_2)}}{b^2(\mathcal{T}_2-\mathcal{T}_1)}.
\end{aligned}$$

A.5 Model E6

$$
\begin{aligned}
\theta &= \{a,b,c,d,q,\lambda\},\\
M &= \frac{A_1}{A_2}+c-\frac{d}{\Omega}\left[\cos\Omega(t+1)-\cos\Omega t\right]-\frac{q}{\Omega}\left[\sin\Omega(t+1)-\sin\Omega t\right],\\
N &= a^2\widehat{I}^2\widehat{J}_1+2a\widehat{I}\widehat{J}_2+J_3,\\
A_1 &= a(e^b-1)(e^{b(t-\mathcal{T}_1)}-e^{b(t-\mathcal{T}_2)}),\\
A_2 &= b^2(\mathcal{T}_2-\mathcal{T}_1),\\
\widehat{J}_1 &= \frac{(e^{2b}-1)}{2b},\\
\widehat{J}_2 &= \frac{c(e^b-1)}{b}+\frac{db-q\Omega}{b^2+\Omega^2}\left[e^b\sin\Omega(t+1)-\sin\Omega t\right]\\
&\quad -\frac{d\Omega+bq}{b^2+\Omega^2}\left[e^b\cos\Omega(t+1)-\cos\Omega t\right],\\
J_3 &= (c^2+\frac{d^2+q^2}{2})\\
&\quad -\frac{d^2-q^2}{4\Omega}\left[\sin 2\Omega(t+1)-\sin 2\Omega t\right]+\frac{dq}{2\Omega}\left[\cos 2\Omega(t+1)-\cos 2\Omega t\right]\\
&\quad -\frac{2cq}{\Omega}\left[\sin\Omega(t+1)-\sin\Omega t\right]-\frac{2cd}{\Omega}\left[\cos\Omega(t+1)-\cos\Omega t\right].
\end{aligned}
$$

Partial derivatives of M

$$
\begin{aligned}
\frac{\partial M}{\partial a} &= \frac{(e^b-1)}{b}\widehat{I},\\
\frac{\partial M}{\partial b} &= \frac{A_1'}{A_2}-\frac{A_1A_2'}{A_2^2},\\
\frac{\partial M}{\partial c} &= 1,\\
\frac{\partial M}{\partial d} &= \frac{-1}{\Omega}\left[\cos\Omega(t+1)-\cos\Omega t\right],\\
\frac{\partial M}{\partial q} &= \frac{-1}{\Omega}\left[\sin\Omega(t+1)-\sin\Omega t\right],\\
\frac{\partial M}{\partial \lambda} &= 0,\\
A_1' &= a(e^b-1)\left[(t-\mathcal{T}_1)e^{b(t-\mathcal{T}_1)}-(t-\mathcal{T}_2)e^{b(t-\mathcal{T}_2)}\right]\\
&\quad +ae^b(e^{b(t-\mathcal{T}_1)}-e^{b(t-\mathcal{T}_2)}),\\
A_2' &= 2b(\mathcal{T}_2-\mathcal{T}_1).
\end{aligned}
$$

Partial derivatives of N

$$\begin{aligned}
\frac{\partial N}{\partial a} &= 2a\widehat{I}^2\widehat{J}_1 + 2\widehat{I}\widehat{J}_2, \\
\frac{\partial N}{\partial b} &= a^2\left[2\widehat{I}\widehat{I}'\widehat{J}_1 + \widehat{I}^2(\widehat{J}_1)'\right] + 2a\left[(\widehat{I})'\widehat{J}_2 + \widehat{I}(\widehat{J}_2)'\right], \\
\frac{\partial N}{\partial c} &= 2a\widehat{I}\frac{\partial \widehat{J}_2}{\partial c} + \frac{\partial J_3}{\partial c}, \\
\frac{\partial N}{\partial d} &= 2a\widehat{I}\frac{\partial \widehat{J}_2}{\partial d} + \frac{\partial J_3}{\partial d}, \\
\frac{\partial N}{\partial dq} &= 2a\widehat{I}\frac{\partial \widehat{J}_2}{\partial q} + \frac{\partial J_3}{\partial q}, \\
\frac{\partial N}{\partial \lambda} &= 0, \\
\frac{\partial \widehat{I}}{\partial b} &= \frac{(t-\mathcal{T}_1)\,e^{b(t-\mathcal{T}_1)} - (t-\mathcal{T}_2)\,e^{b(t-\mathcal{T}_2)}}{b(\mathcal{T}_2-\mathcal{T}_1)} - \frac{e^{b(t-\mathcal{T}_1)} - e^{b(t-\mathcal{T}_2)}}{b^2(\mathcal{T}_2-\mathcal{T}_1)}, \\
\frac{\partial \widehat{J}_1}{\partial b} &= \frac{e^{2b}}{b} - \frac{\left(e^{2b}-1\right)}{2b^2}, \\
\frac{\partial \widehat{J}_2}{\partial b} &= c\frac{be^b - e^b + 1}{b^2} + A_3 e^b \sin\Omega(t+1) + A_3'\left[e^b\sin\Omega(t+1) - \sin\Omega t\right] \\
&\quad -A_4 e^b\cos\Omega(t+1) - A_4'\left[e^b\cos\Omega(t+1) - \cos\Omega t\right], \\
A_3 &= \frac{db - q\Omega}{b^2+\Omega^2}, \quad A_4 = \frac{d\Omega + bq}{b^2+\Omega^2}, \\
A_3' &= \frac{d\Omega^2 - db^2 + 2qb\Omega}{\left(b^2+\Omega^2\right)^2}, \\
A_4' &= \frac{q\Omega^2 - qb^2 - 2db\Omega}{\left(b^2+\Omega^2\right)^2}, \\
\frac{\partial \widehat{J}_2}{\partial c} &= \frac{\left(e^b-1\right)}{b}, \\
\frac{\partial J_3}{\partial c} &= 2c - \frac{2q}{\Omega}\left[\sin\Omega(t+1) - \sin\Omega t\right] - \frac{2d}{\Omega}\left[\cos\Omega(t+1) - \cos\Omega t\right], \\
\frac{\partial \widehat{J}_2}{\partial d} &= \frac{b}{b^2+\Omega^2}\left[e^b\sin\Omega(t+1) - \sin\Omega t\right] \\
&\quad -\frac{\Omega}{b^2+\Omega^2}\left[e^b\cos\Omega(t+1) - \cos\Omega t\right], \\
\frac{\partial J_3}{\partial d} &= d - \frac{d}{2\Omega}\left[\sin 2\Omega(t+1) - \sin 2\Omega t\right] + \frac{q}{2\Omega}\left[\cos 2\Omega(t+1) - \cos 2\Omega t\right] \\
&\quad -\frac{2c}{\Omega}\left[\cos\Omega(t+1) - \cos\Omega t\right], \\
\frac{\partial \widehat{J}_2}{\partial q} &= \frac{-\Omega}{b^2+\Omega^2}\left[e^b\sin\Omega(t+1) - \sin\Omega t\right] \\
&\quad -\frac{b}{b^2+\Omega^2}\left[e^b\cos\Omega(t+1) - \cos\Omega t\right], \\
\frac{\partial J_3}{\partial q} &= q + \frac{q}{2\Omega}\left[\sin 2\Omega(t+1) - \sin 2\Omega t\right] + \frac{d}{2\Omega}\left[\cos 2\Omega(t+1) - \cos 2\Omega t\right] \\
&\quad -\frac{2c}{\Omega}\left[\sin\Omega(t+1) - \sin\Omega t\right].
\end{aligned}$$

B Initial results on the ARMA model

This appendix presents an initial experiment on the univariate ARMA model for forecasting electricity demand.

The ARMA model is defined by:

$$\begin{aligned} y_t &= A_q(L)y_t + B_p(L)\varepsilon_t \\ &= (a_1 y_{t-1} + \cdots + a_q y_{t-q}) + (\varepsilon_t + b_1\varepsilon_{t-1} + \cdots + b_p\varepsilon_{t-p}), \end{aligned}$$

where ε_t is assumed to be a Gaussian noise, and L is the lag operator. The first term $A_q(L)y_t$ is called auto-regressive component and the second term $B_p(L)\varepsilon_t$ is a moving average of Gaussian noise. We used these models to predict demand 1-day ahead.

As discussed in Section 2.5.1 on page 36, the electricity demand significantly drops on special days. Because there is no input selection step for this model, we performed the same smoothing methodology as in (Taylor, 2008): before fitting the model and predicting, we smoothed the data by replacing data on special days by the electricity demand on the same day of the closest previous week, which is not a special day. When evaluating model performance, we exclude the predictions of the special days.

The model was fitted using the System Identification toolbox from Matlab. Order p and q of the model were selected by ACF and PACF of the time series (see Figure 3.3 on page 63). The model for predicting one-day ahead electricity demand had the following form:

$$\begin{aligned} A_8(L) &= -0.8517L^{-1} + 0.0004942L^{-2} + 0.0001586L^{-3} - 0.0005116L^{-4} \\ &\quad -0.0001L^{-5} - 0.001127L^{-6} - L^{-7} + 0.8526L^{-8}, \end{aligned}$$

$$\begin{aligned} C_7(L) &= 1 + 0.1415L^{-1} + 0.09212L^{-2} + 0.02651L^{-3} + 0.02764L^{-4} \\ &\quad +0.1161L^{-5} + 0.1818L^{-6} - 0.8366L^{-7}. \end{aligned}$$

Models	IR(RMSE)	RMSE	NRMSE	MAPE	MAE	NMAE
Benchmark	0.00%	39365	0.36550	2.96%	29011	0.32877
ARMA	**51.78%**	**18983**	**0.17897**	**1.47%**	**13938**	**0.15794**
LR-GARCH	45.72%	21369	0.19841	1.72%	16538	0.18742
LR	44.49%	21850	0.20252	1.76%	16915	0.19112
MLP	**53.12%**	**18455**	**0.17135**	**1.43%**	**13940**	**0.15798**
RBF	**48.72%**	**20187**	**0.18743**	**1.63%**	**15589**	**0.17666**

Table B.1: Errors and RMSE improvement ratio of the ARMA model and the other standard forecasting models on the electricity demand dataset.

Table B.1 shows the results of the ARMA model comparing to other standard forecasting models. It indicates that the ARMA model is a promising model. In the future we would like to combine this model with the improvement techniques and apply them to the problems considered in this thesis. Related discussions about this future work is presented in Section 8.2 on page 151.

C ACF of the squared standardised residuals of the LR and LR-GARCH models

This appendix includes a figure of the auto-correlation function of the squared standardised residuals of the LR and LR-GARCH models on the gas price dataset. The figure shows an evidence for the motivation of using GARCH component, which was discussed in Section 3.3.4 on page 51.

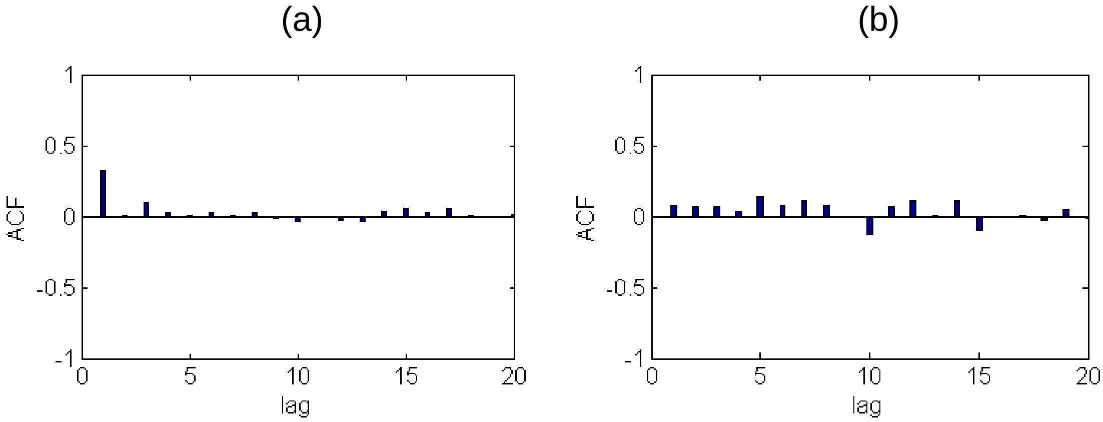


Figure C.1: ACF of the squared standardised residuals of the LR and LR-GARCH models for the gas price dataset. (a) LR. (b) LR-GARCH

D Results on pre-processing procedures

This appendix presents experimental results of the input selection step for models combined with RHWT in Chapter 4.

D.1 Correlation matrix

We computed the correlation matrix ρ of electricity demand and its WT components with exogenous variables. Figure D.1 shows the absolute value $|\rho|$ of the correlation matrix. The indexed attributes in the correlation matrix are listed as follows:

1 Electricity demand at time step t (This is target value in direct-forecast and the forecast models without WT).

2 A at time step t (This is target value in multicomponent-forecast: component A).

3 D_2 at time step t (This is target value in multicomponent-forecast: component D_2).

4 D_1 at time step t (This is target value in multicomponent-forecast: component D_1).

5 Electricity demand at time step $t-1$.

6 A at time step $t-1$.

7 D_2 at time step $t-1$.

8 D_1 at time step $t-1$.

9 Electricity demand at time step $t-2$.

10 A at time step $t-2$.

11 D_2 at time step $t-2$.

12 D_1 at time step $t-2$.

13 Electricity supply at the time step $t-1$.

14 Electricity supply at the time step $t-2$.

15 Electricity supply at the time step $t-3$.

16 Transformed temperature at the time step $t-1$.

17 Transformed temperature at the time step $t-2$.

18 Transformed temperature at the time step $t-3$.

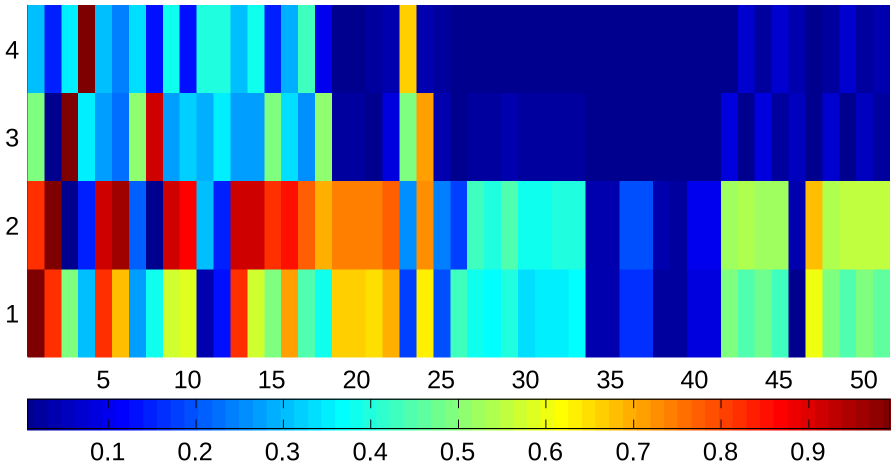


Figure D.1: Absolute value of correlation matrix of electricity demand and its WT components with exogenous variables. Both horizontal and vertical axes represent targets (i.e. the first four variables) and potential input variables (the remaining variables). Because we concern the correlation coefficients between each target and each potential input only (but not between a input and another input), the vertical axe includes the four target variables only. Removing other parts of the correlation matrix make the figure clearer.

19 Average temperature at the time step $t-1$.

20 Average temperature at the time step $t-2$.

21 Average temperature at the time step $t-3$.

22 Gas demand $t-1$.

23 swd at time step t.

24 cwd at time step t.

25 syd at time step t.

26 cyd at time step t.

27 Price of weekday ahead base load electricity product at time step $t-1$.

28 Price of weekday ahead peak load electricity product at time step $t-1$.

29 Price of weekend ahead base load electricity product at time step $t-1$.

30 Price of one-month-ahead forward product, base load at time step $t-1$.

31 Price of one-month-ahead forward product, base load at time step $t-2$.

32 Price of one-month-ahead forward product, peak load at time step $t-1$.

33 Price of one-month-ahead forward product, peak load at time step $t-2$.

34 Price of one-winter-ahead forward product, base load at time step $t-1$.

35 Price of one-winter-ahead forward product, base load at time step $t-2$.

36 Price of one-summer-ahead forward product, base load at time step $t-1$.

37 Price of one-summer-ahead forward product, base load at time step $t-2$.

38 Price of one-winter-ahead forward product, peak load at time step $t-1$.

39 Price of one-winter-ahead forward product, peak load at time step $t-2$.

40 Price of one-summer-ahead forward product, peak load at time step $t-1$.

41 Price of one-summer-ahead forward product, peak load at time step $t-2$.

42 Gas SMP buy at time step $t-1$.

43 Gas SMP buy at time step $t-2$.

44 Gas SMP sell at time step $t-1$.

45 Gas SMP sell at time step $t-2$.

46 Weather: wind speed at time step $t-1$.

47 Weather: sunset time at time step $t-1$.
48 Gas SAP at time step $t-1$.
49 Gas SAP at time step $t-2$.
50 Price of day-ahead gas forward product at time step $t-1$.
51 Price of day-ahead gas forward product at time step $t-2$.

D.2 ACF and PACF of the electricity demand and WT components

We computed the ACF and PACF of the electricity demand and its WT components (see Figure D.2). They were used to select input variables for the linear models for forecasting the electricity demand.

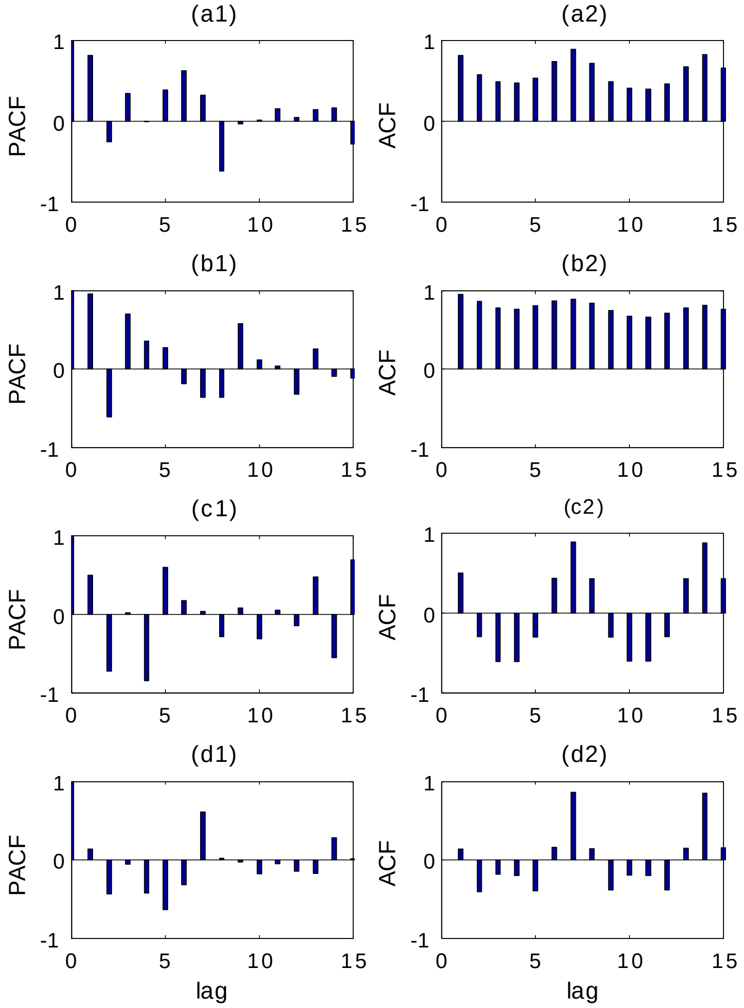


Figure D.2: ACF and PACF of daily electricity demand and WT components. (a1) and (a2) are the PACF and ACF of the electricity demand, respectively. (b1) and (b2) are the PACF and ACF of component A_2, respectively. (c1) and (c2) are the PACF and ACF of component D_2, respectively. (d1) and (d2) are the PACF and ACF of component D_1, respectively.

E Results on different scenarios of the adaptive models

This appendix presents results on the different scenarios of the adaptive models which were mentioned in Sections 5.4 (pages 103) and 5.5 (page 106).

Table E.1 shows results on different scenarios of the adaptive models on the electricity demand dataset. The second column presents RMSE on the case where only bias of the models were updated. The third column presents the results on the case where we tried to update more parameters of the models: in the LR model, we tested on updating all parameters; in MLP and RBF models, we tested on updating all second layer parameters; and in the LR-GARCH model, we updated both $\widetilde{\beta}$ and $\widehat{\beta}$. The experimental results showed that results on updating the bias is slightly better than updating more parameters. Similar results on the gas forward price dataset are shown in Table E.2.

RMSE	Updating the bias only	Updating more parameters
LR-GARCH+EKF	21312	21667
LR-GARCH+PF	21232	21571
LR+EKF	21337	21485
LR+PF	21343	21493
MLP+EKF	**17266**	**17268**
MLP+PF	**17493**	**17493**
RBF+EKF	19792	19811
RBF+PF	19810	19830

Table E.1: RMSE of the different scenarios of the adaptive models on the electricity demand forecasting.

RMSE	Updating the bias only	Updating more parameters
LR+EKF	1.06916	1.07094
LR+PF	1.06917	1.07138
LR-GARCH+EKF	**1.04144**	**1.04191**
LR-GARCH+PF	**1.04145**	**1.04231**
MLP+EKF	1.06588	1.06613
MLP+PF	1.06586	1.06615
RBF+EKF	1.06347	1.06359
RBF+PF	1.06327	1.06368

Table E.2: Average RMSE of different scenarios of the adaptive models on the gas forward price forecasting.

F Training the Student-t LR-GARCH

This appendix presents a maximum-likelihood methodology for training the Student-t LR-GARCH model. It is used in Section 6.5.2, page 136. The structure of a LR-GARCH model is presented in Section 3.3.4, page 50. We assume that the data is corrupted by a Student-t noise distribution ε:

$$p(\varepsilon|\lambda,\nu) = \frac{\Gamma(\nu/2+1/2)}{\Gamma(\nu/2)}\left(\frac{1}{\pi\nu}.\frac{1}{\sigma^2}\right)^{1/2}\left[1+\frac{1}{\sigma^2}\frac{\varepsilon^2}{\nu}\right]^{-\nu/2-1/2},$$

where ν is the number of degrees-of-freedom and σ is the scale parameter of the distribution. Note that this equation is another expression of Equation (6.1), page 120. We use this form of Student-t in order to make it convenient to compute the likelihood. In the LR-GARCH model, σ^2 is denoted by n_t and it changes over time.

Given training data $D = \{(x_1, y_1), (x_2, y_2), ..., (x_T, y_T)\}$, the negative log likelihood is given by:

$$\begin{aligned}\mathcal{L} &= -\log(p(D/\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\nu)) \\ &= -T\log\left\{\frac{\Gamma\left(\nu+1\right)}{\Gamma(\nu/2)}\left[\frac{1}{\pi\left(\nu-2\right)}\right]^{1/2}\right\}+\frac{1}{2}\sum_{t=1}^{T}\log(n_t)+\frac{1}{2}\left(\nu+1\right)\sum_{t=1}^{T}\log\left[1+\frac{\varepsilon_t^2}{(\nu-2)n_t}\right].\end{aligned}$$

We used scaled conjugate gradient to optimise $\mathcal{L}$ with respect to $\theta = \{\beta,\ \alpha,\ \gamma,\ \nu\}$. This local optimal algorithm requires partial derivatives of $\mathcal{L}$. Because it is difficult to analytically compute these partial derivatives, we used the following finite difference approximation:

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}} \approx \frac{\mathcal{L}(\boldsymbol{\theta}+\Delta\boldsymbol{\theta})-\mathcal{L}(\boldsymbol{\theta}-\Delta\boldsymbol{\theta})}{2\Delta\boldsymbol{\theta}}.$$

The code for this is derived from (Press et al., 1992). One of the disadvantages of this approach is that it is computationally expensive.