Natural gas price forecasting using recurrent neural networks

Christian Koopmann

School of Business and Economics

Humboldt Universität zu Berlin

A thesis submitted for the degree of

MSc. Statistics

Berlin, 2017

Statement of Originality

I hereby confirm that I have written the accompanying thesis by myself, without contributions from any sources other than those cited in the text, the references, and the acknowledgements.

This applies explicitly to all graphics, drawings, maps and images included in the thesis.

Berlin, November 23, 2017

Christian Koopmann

Acknowledgements

This thesis has been completed in cooperation with *BASF SE* in the *Advanced Data Analytics* team of the procurement department. I am especially thankful to my supervisor Dr. Ferdinand Kiermaier as well as my colleagues both in the Data Analytics team as well as the energy trading department, for their valuable input and feedback.

Special thanks go to my thesis supervisor Prof. Dr. Stefan Lessmann for his support and guidance throughout the completion of this thesis. Especially his inputs regarding the specification of the topic as well as the interpretation of the results were of particular value.

Abstract

In this thesis Long Short Term Memory networks are used to model prices of monthly futures for natural gas on the TTF trading hub. The models are used to predict both daily closing prices as well as a binary variable indicating price optimality among the remaining trading days until delivery.
Based on the results from the first prediction task the results regarding LSTM performance are mixed, with the method outperforming alternative neural network structures but providing no predictive advantage over current market prices.
Regarding the binary predictions the LSTM model shows a stronger performance outperforming all reference models including simpler recurrent structures as well as feed forward neural networks and logistic regression. Based on these predictions a trading strategy is derived that is able to generate significant savings over the average market price for buyers of natural gas.

Contents

I Introduction **1**

1 Introduction **2**

1.1 Motivation 2

1.2 Prediction problems 3

1.3 Research questions 3

1.4 Structure 4

II The natural gas market **5**

2 The natural gas market **6**

2.1 History 6

2.2 Types of natural gas trading 8

2.3 Natural gas derivatives 10

III Methodology **11**

3 Recurrent neural networks **12**

3.1 General concept 12

3.2 Simple RNN architecture 13

3.3 Vanishing gradient problem 13

3.4 LSTM architecture 16

IV Literature review **19**

4 Literature review **20**
4.1 Electricity 21
4.2 Oil 25
4.3 Natural Gas 28
4.4 Summary 31

V Empirical study **32**

5 Data **33**
5.1 Target Variable 33
5.1.1 Price Level Prediction 33
5.1.2 Binary Prediction 35
5.2 Input Variables 37
5.2.1 Energy Commodity Prices 37
5.2.2 Gas Market Fundamentals 38
5.2.3 Exchange Rates 39
5.2.4 Data Availability 39

6 Experimental Design **41**
6.1 Preprocessing 42
6.1.1 Treatment of missing data 43
6.1.2 Feature scaling 43
6.2 Reference Models 44
6.3 Parameter Tuning 45
6.3.1 Fixed Hyper-Parameters 45
6.3.2 Tuning Parameters 47
6.3.3 Tuning Process 47
6.4 Variable Selection 48
6.5 Model Evaluation 48

7 Results **49**
7.1 Price Level Prediction 49
7.1.1 Univariate Parameter Tuning 49
7.1.2 Variable Selection 50
7.1.3 Multivariate Parameter Tuning 51
7.1.4 Evaluation 51
7.2 Binary Prediction 54
7.2.1 Univariate Parameter Tuning 54
7.2.2 Variable Selection 55
7.2.3 Multivariate Parameter Tuning 55
7.2.4 Evaluation 56
7.2.5 Trading Strategy 58

VI Conclusions **62**

8 Conclusions **63**
8.1 Research Questions 63
8.1.1 Performance of recurrent neural networks 63
8.1.2 Relative performance of LSTM 64
8.1.3 Business value 65
8.2 Other observations 65
8.3 Outlook 66

VII Appendix **74**
8.4 Figures 75
8.5 Tables 85

List of Figures

2.1 Selected European Gas Hubs. Primary Focus: TTF, Secondary Focus: NBP, GP 8
2.2 Types of Natural Gas Trading 9

3.1 Simplified RNN Architecture 14
3.2 Hidden Layer of Elman Network 14
3.3 Hidden Layer of LSTM Network 18

5.1 TTF Front Month vs. Individual Months 34
5.2 Auto Correlations TTF Front Month Closing Prices 35
5.3 Partial Auto Correlations TTF Front Month Closing Prices 35
5.4 Binary target variable and price level for December 2016 36

7.1 Month wise MSE for multivariate models 53
7.2 Month wise MSE for univariate models 53
7.3 Predictions LSTM vs. Lagged Value 54
7.4 Month wise BCE for multivariate models 57
7.5 Month wise BCE for univariate models 57
7.6 Average prices for multivariate models 60
7.7 Average prices for univariate models 60

8.1 TTF Front Month vs. Individual Months 75
8.2 TTF Front Month Closing Prices 76
8.3 TTF Day Ahead 76
8.4 NBP Front Month 77
8.5 Brent Oil Front Month 77

8.6 Phelix Base Load Front Month 78
8.7 Phelix Peak Load Front Month 78
8.8 Dutch LDZ Consumption 79
8.9 Dutch Non-LDZ Consumption 79
8.10 Dutch Production 80
8.11 UK Continental Shelf Production 80
8.12 UK Continental Shelf Production 81
8.13 Dutch Storage 81
8.14 UK Storage 82
8.15 BBL Pipeline Flow 82
8.16 Interconnector Pipeline Flow 83
8.17 USD - EUR Exchange Rate 83
8.18 GBP - EUR Exchange Rate 84

List of Tables

2.1 Future vs. Spot Markets 10

4.1 Electricity Price Forecasting Literature 25
4.2 Oil Price Forecasting Literature 28
4.3 Gas Price Forecasting Literature 30

5.1 Data Availability of all Input Variables 40

7.1 Selected parameter combinations in univariate tuning step of price level prediction 50
7.2 Selected variable combinations in variable selection step of price level prediction 51
7.3 Selected parameter combinations in multivariate tuning step of price level prediction 51
7.4 Average MSE across months for each model in evaluation step 52
7.5 Selected parameter combinations in univariate tuning step of binary prediction 55
7.6 Selected variable combinations in variable selection step of binary prediction 55
7.7 Selected parameter combinations in multivariate tuning step of binary prediction 56
7.8 Average BCE across months for each model in evaluation step 56
7.9 Average price per MWh across months for simple trading strategy based on each model 61

8.1 Annual share of positive observation for binary target variable 85

8.2 Savings in EUR/MWh of univariate LSTM model relative to equal distribution benchmark for differenttrading periods 86

Part I

Introduction

Chapter 1

Introduction

1.1 Motivation

In the recently published World Energy Outlook 2017 the International Energy Agency predicts natural gas to be the fastest growing fuel, accounting for a quarter of all growth in energy consumption until 2040 and overtaking coal to become the second most important fuel (IEA (2017)). This follows their prediction from 2011 of a ”Golden age of Natural Gas” (IEA (2011)) with the growing trade in Liquified Natural Gas and the shale gas revolution of North America being the main growth drivers on the supply side. On the demand side most growth will come from power generation and industrial processing, which will lead to a growing impact of natural gas procurement on the bottom line of many corporations. At the same time trading natural gas is made more complex by the continuing replacement of long term oil indexed contracts by exchange-traded hub priced contracts. Economic success on the natural gas market will therefore be increasingly dependent on successfully modelling and forecasting the price of these contracts. Despite this growing importance natural gas prices have so far received less attention by the forecasting literature than other commodities or financial markets. This thesis tries to fill that gap by applying state of the art machine learning methods to the prediction of future prices on the European natural gas market. Despite their recent success in other time series prediction problems Long Short Term Memory Networks have not yet been widely applied in this area and therefore constitute a natural candidate for this task.

1.2 Prediction problems

Since this thesis was written in cooperation with an industrial partner in the chemical industry it pursues both a methodological as well as an application oriented objective. On the methodological side the aim is to research the power of Long Short Term Networks in the prediction of financial time series relative to other approaches. Regarding the application the purpose is to develop a tool to serve natural gas gas traders in the chemical industry in the support of daily purchasing decisions. To serve both of these objectives equally well, this thesis analyses two prediction problems:

Price Level Prediction: This forecasting problem consists of predicting the closing price of the natural gas future on the next trading day, based on all data available up to the current day.For both the energy commodity as well as a wide selection of other financial markets, this or a very similar problem has been researched for a long time. Therefore methods as well as results from this prediction problem could be easily extended and compared to other areas.

Binary Prediction: Natural gas traders in the chemical industry usually try to purchase gas to satisfy a physical demand in the business at an optimal price. This means that they are often limited to only act as a buying party on the market and need to buy a fixed amount of gas within a certain time frame. Under these restrictions the question these market participants have to ask themselves is: ”Is today’s price minimal, among all remaining trading days until the demand has to be met?”. The binary variable corresponding to the answer to this questions is the target in this second prediction problem.

In the interest of consistency the approach and methods used for both prediction problems will be identical wherever reasonable.

1.3 Research questions

Based on these prediction problems the thesis will attempt to answer the following research questions:

- How well are recurrent neural networks in general able to solve the given prediction problems ?
- Do Long Short Term Memory models offer an advantage over simpler architectures in these prediction problems?
- What is the business value that these models can provide to buyers of natural gas futures ?

1.4 Structure

This thesis will be structured into six parts, the first of which is this introduction. After the introduction an overview over the history and general structure of the European gas markets will be given. Afterwards the reader is provided with a brief explanation of recurrent neural networks in general and the Long Short Term Memory architecture in particular. Here the reader is assumed to have a basic understanding of feed forward neural networks and back propagation. Based on these two chapters the literature regarding price forecasting in energy commodity markets will be reviewed. In part five, the empirical study of above prediction problems, which forms the core of this thesis, will be illustrated. In the course of this explanation an overview over the Data used for the study will be given, as well as an in-depth explanation of the experimental design and an extensive evaluation of the results. In part six the findings from this evaluation are summarised trying to answer above questions and explore possible future directions of research extending this work.

Part II

The natural gas market

Chapter 2

The natural gas market

In this section the reader will be given a brief overview of the history and current situation in the European natural gas market. Special emphasis will lie on the different ways natural gas is traded and how the current system evolved.

2.1 History

The first commercially used gas was in fact produced from coal and used as a lighting fuel beginning in the late 18th century in Britain, afterwards spreading to the United States and Continental Europe during the 19th century. The first well extracting natural gas directly from the ground was dug 1821 in Fredonia, New York (Heather (2015)). At the beginning this gas could only be consumed at locations close to its source. The expansion of the application of natural gas into fields such as heating, chemical processing and power generation was connected with the construction of effective long distance pipelines. In Europe the advent of large scale natural gas imports from the former Soviet Union and North Africa in the mid 20th century required especially large infrastructure investments. To recover this investment long term contracts with durations of 25 to 30 years were negotiated by the usually state owned entities that controlled the natural gas trade. These entities usually integrated many different areas of the value chain from exploration over pipeline transport to the sale of natural gas. During this time there were relatively few participants in the European natural gas market that often occupied a monopoly position in their respective national markets. For the sale of gas from the then newly developed Groningen field the Dutch producers developed an oil based pricing system, where the price for long term natural gas contracts was tied to the market price for crude oil. This pricing system spread to many other natural gas contracts and became the predominant pricing system in the European market. An alternative pricing regime developed in the 1970s in the United States. This pricing regime is based on defining regional gas trading hubs and developing independent markets for gas traded at each of these hubs. In the US the Henry Hub in Louisiana has evolved as the main trading hub. Hub pricing spread from the US to the UK with the privatisation of the British gas industry in the 1980s and the establishment of the National Balancing Point (NBP) as the main trading hub for the UK market. Whereas the Henry Hub evolved from a physical hub connecting various long distance pipelines the NBP was established as a Virtual Trading Point (VTP) with the aim of developing a new price reference for natural gas trading. Regulatory reform and privatisation of the gas industry was a key requirement for the establishment of these markets since it allowed for the separation of distribution networks from natural gas trading activities and thereby ensured that all market participants had equal access to the distribution infrastructure. Aiming to establish an integrated European energy market these regulatory changes have later been adopted by the European Union and spread to the other member countries. This has lead to the establishment of Continental European virtual trading points with the Dutch Title Transfer Facility (TTF) emerging as the main hub. The establishment of these hubs has allowed natural gas to evolve into a commodity traded on spot and future markets similar to crude oil. With the improvement of the technology and infrastructure for trading Liquified Natural Gas a new alternative to pipelines has emerged and the gas market has become more globally integrated. Due to the recent boom in domestic shale gas production the United States has drastically reduced its LNG imports and is becoming a net exporter (IEA (2017)). This has lead to an oversupply in LNG markets with a resulting downward pressure on gas prices. As a result of this the long term oil indexed contracts have become a lot less attractive for buyers, who have pushed for switching to hub pricing. The share of oil indexed pricing in the North West European market (including France, Germany, UK and Netherlands) has fallen from 70 percent in 2005 to just over 10 percent in 2014. Accordingly hub pricing accounted for 90 percent in 2014 (Heather (2015), p. 14). In the past years various new national trading hubs were created throughout Europe with the two German hubs NetConnect Germany (NCG) and Gaspool (GP) chief among them. However these new exchanges still lack far behind the established hubs (NBP, TTF) in terms of liquidity and trading volume.

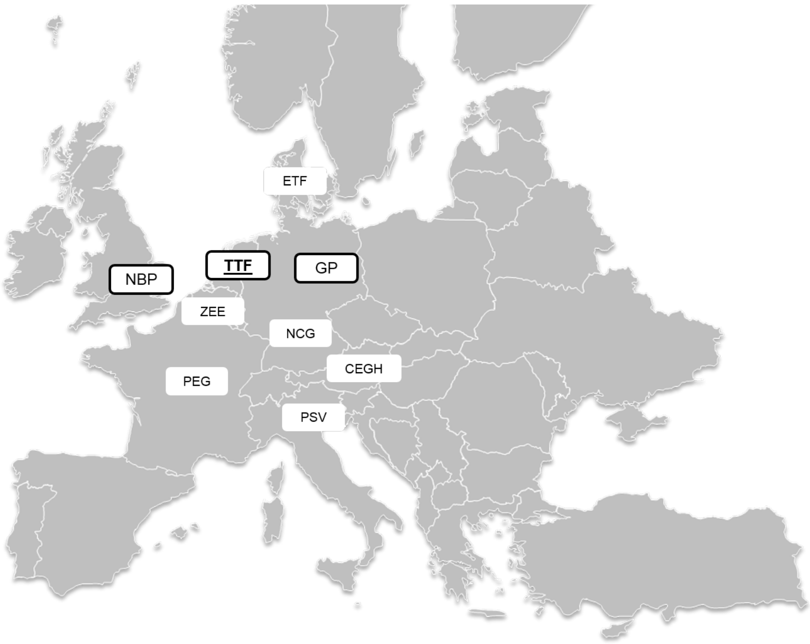

Figure 2.1: Selected European Gas Hubs. Primary Focus: TTF, Secondary Focus: NBP, GP

2.2 Types of natural gas trading

In the context of the above outlined history of natural gas three basic ways of trading have emerged, which are visualised in Figure 2.2.

Bilateral negotiations The most classical way of trading natural gas are individually negotiated contracts between supplier and buyer. These are often very long term contracts aimed at ensuring a reliable physical supply of natural gas. Just as the other contract features, pricing systems differ from contract to contract and are not transparent to other market participants. Prices are usually specified to follow a specific benchmark such as the oil price or the gas price at a certain trading hub.

Over-the-Counter Over the counter trading is a specific system of trading various financial assets. Although the contracts in this system are standardised the trading occurs individually between selling and buying party without the supervision of a financial exchange. This results in less transparency since both quotes and closed trades are not necessarily visible to other market participants. Also there is no clearing house involved in the trading which means that the trading parties are exposed to the credit risk of the opposite party defaulting on its obligations. The products available on natural gas OTC markets contain both spot trading as well as forwards and options.

Exchange trading Analogously to OTC trading, products traded at financial exchanges are standardised contracts such as futures or options. The key difference lies in the way trades between parties are conducted. At financial exchanges the market participants post sell or buy offers publicly to all other traders who then have an equal chance to trade on that offer. That way all offers and trades are visible to the whole market. Another difference is the existence of a clearing house which takes over the credit risk of both parties and pays out open debts in case of a default. Overall trading at exchanges is more tightly regulated and offers less risk with more transparency than OTC trading. The largest exchanges for trading natural gas derivatives of European hubs are the Intercontinental Exchange (ICE) and the European Energy Exchange (EEX). Both of these exchanges offer a variety of derivatives for TTF and NBP gas.

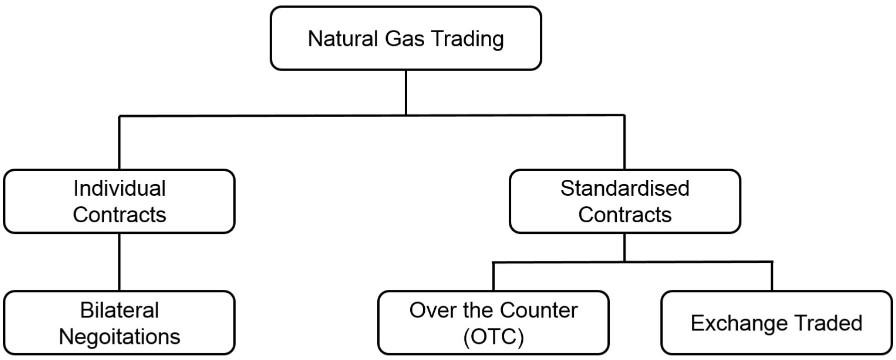

Figure 2.2: Types of Natural Gas Trading

2.3 Natural gas derivatives

The natural gas contracts relevant for the work done in this thesis can be divided into spot and future contracts. Fundamentally these contracts work in the same way. They cover the delivery of a certain amount of gas at an equal rate over a certain time frame in the future. The difference between these markets lie in the length of the time frame as well as how far the delivery time lies in the future. In the context of this work the delivery of a spot contract lies at most one day in the future and the delivery time frame is at most one day long. This means that the spot market contains basically the day ahead and within day products. Day ahead markets trade gas to be delivered over the next day, whereas within day markets trade hourly products for the remaining hours of the current day. Future markets contain all markets with delivery time frames longer than one day. These markets contain monthly as well as quarterly, seasonal and yearly products. The seasonal products are the winter season from October to March and the summer season. Usually the most intensively traded future of each kind is the one with the nearest delivery time, e.g. the next month or quarter. This future will be called the Front-Month and will be the main focus of this study.

Spot Markets	Future Markets
Within Day	Months
Day Ahead	Quarters
	Seasons (Winter / Summer)
	Calendar Year

Table 2.1: Future vs. Spot Markets

Part III

Methodology

Chapter 3

Recurrent neural networks

In the following section I will give a brief overview over the theory behind recurrent neural networks in general and the long short-term memory (LSTM) network architecture in particular. Both the content as well as the illustrations of this section are inspired by the blog post *Understanding LSTM Networks* by Google Researcher Christopher Olah.

3.1 General concept

Traditional feed-forward neural networks have shown to be successful in modelling a variety of non-linear input-output relations. However a major shortcoming when it comes to estimate variables that are part of a sequence is the fact that these models are limited to static input-output relations. This means that a set of values for the input variables produces the same output no matter where in the sequence it is located. Regarding many kinds of sequential data such as speech or financial time series, the input-output relation is presumed to change over the course of the sequence. A recurrent neural network offers a method to model these kinds of dynamic relationships. The main idea is to use the output of the hidden layer of observation $t-1$ in some way as input to the same layer in observation t. A simplified view of the general RNN architecture unfolded across time can be seen in figure 3.1. Here the structure of the model at each time step is separated into the hidden layer which contains the recurrent structure and the output layer which has no connections across time and produces the final predictions $\tilde{y}$. In an

RNN the prediction $\tilde{y}_t$ depends on the inputs at all points in the sequence up to index t. The different types of single layer RNN architectures mainly differ in what the hidden layer H actually consists of and which part of its output is passed to the output layer and which to the hidden layer in the next time step.

3.2 Simple RNN architecture

The simplest specification is to just use the output h_t of a single non-linearity f as input for both output layer at this time point and hidden layer at the next time step. This architecture is called simple recurrent neural network or Elman network. In Figure 3.2 the architecture of the hidden layer of this network is visualised with the parameter matrices W and U for the weight of the past hidden layer output and current explanatory variables. This results in the following recursive formal definition of the hidden layer output:

$$h_t = f(Wh_{t-1} + Ux_t)$$

For this definition to be complete one needs to specify some initial state of the hidden layer output h_0. This state can either be fixed to some reasonable level (e.g. 0) or can be treated as an additional set of parameter over which to train the model. Note that the dimension of the hidden output h can be chosen independently of the number of input variables. Be d_h and d_x the number of hidden layer outputs and input variables respectively than the total number of trainable parameters is: $n_{par} = d_h * (d_h + d_x + 1)$ for a model with bias term.

3.3 Vanishing gradient problem

Just as feed forward neural networks, an RNN is trained using the idea of backpropagation. Therefore the weights are updated using the derivative of the loss function with respect to each weight. In the following the loss function will be assumed to be additive across observations. Therefore the analysis will be limited to the derivative of the loss function for one observation $l_t = l(y_t, \tilde{y}_t)$ and the overall derivative will be the sum of these derivatives across time. From the previously defined general structure of an RNN

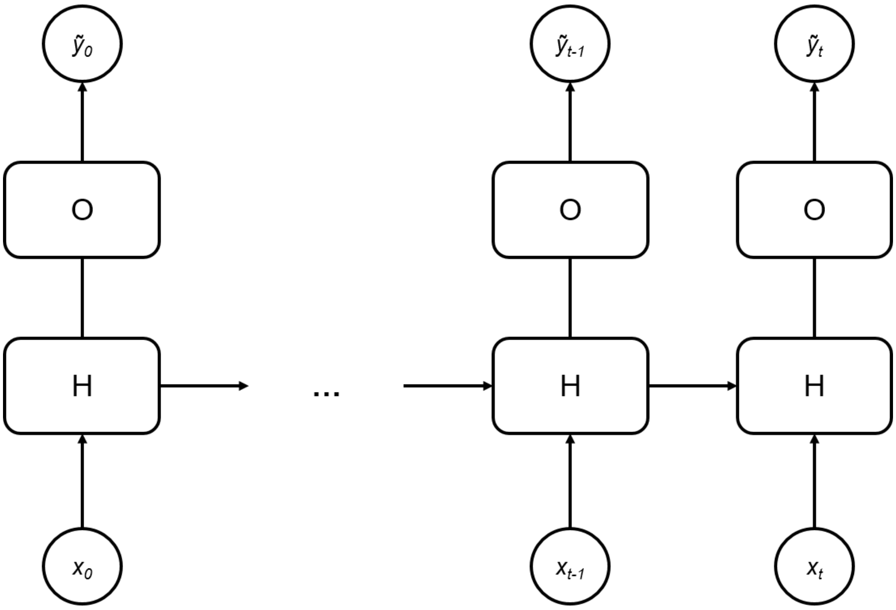

Figure 3.1: Simplified RNN Architecture

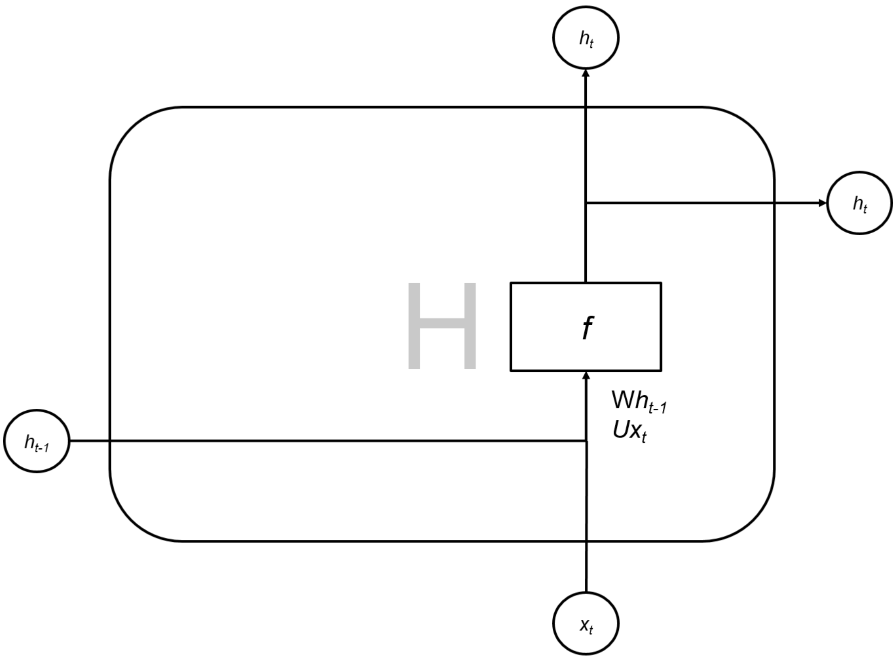

Figure 3.2: Hidden Layer of Elman Network

network it is known, that the prediction $\tilde{y}_t$ is a function (represented by the Output Layer in figure 3.1) of the hidden layer output h_t at that time point. Therefore the loss can be rewritten as:

$$l_t = l(y_t, \tilde{y}_t)$$
$$l_t = l(y_t, f_{output}(h_t))$$

To get the derivative of this loss with respect to one of the recurrent weights W_{ij} using the chain rule one gets:

$$\frac{dl_t}{dW_{ij}} = \frac{l(y_t, f_{output}(h_t))}{df_{output}(h_t)} \frac{df_{output}(h_t)}{dh_t} \frac{dh_t}{W_{ij}}$$

The first two terms on the right hand side are independent of the structure of the recurrent / hidden layer and in the following the analysis will therefore concentrate on the derivative of the hidden layer output $\frac{dh_t}{W_{ij}}$. Entering the definition of the hidden layer of a simple RNN architecture one gets the following expression using the chain and product rule:

$$\frac{dh_t}{dW_{ij}} = \frac{df(Wh_{t-1} + Ux_t)}{dW_{ij}}$$
$$\frac{dh_t}{dW_{ij}} = \frac{df(Wh_{t-1})}{dWh_{t-1}} \frac{dW}{dW_{ij}} h_{t-1} + \frac{df(Wh_{t-1})}{dWh_{t-1}} W \frac{dh_{t-1}}{dW_{ij}}$$

Due to the right hand term $\frac{dh_{t-1}}{dW_{ij}}$ this again is a recursive formula which we can expand up to $\frac{dh_0}{dW_{ij}} = 0$ and get:

$$\frac{dh_t}{dW_{ij}} = \sum_{k=1}^{t} (\prod_{l=1}^{k} \frac{df(Wh_{t-l})}{dWh_{t-l}}) W^{k-1} \frac{dW}{dW_{ij}} h_{t-k}$$

This means that the effect of the hidden layer output k time steps in the past on the current loss gradient is multiplied by a factor of $(\prod_{l=1}^{k} \frac{df(Wh_{t-l}}{dWh_{t-l}}) W^{k-1} \frac{dW}{dW_{ij}}$. Due to its exponential structure the absolute value of this term will either decay towards zero or exponentially rise, depending on the structure of the function f as well as the determinant of the matrix W. These cases are called the *vanishing* and *exploding gradient problem* respectively. While in practice the second problem could be solved relatively easily by clipping the gradient to a certain maxim value, the problem of a vanishing gradient cannot be solved within the framework of this recurrent architecture. This problem severely limits the ability of simple RNNs to learn long term dependencies across time. While in theory simple RNNs could learn dependencies across arbitrarily large time lags in practise they are limited to influences across just a few time steps. Note that the derivative with regard to one of the parameters of the external variables U_{ij} would have a very similar structure and learning these parameters would therefore suffer from the same problems.

3.4 LSTM architecture

The LSTM architecture tries to overcome the vanishing gradient problem using a number of gates to control how the error gradient is passed through the network. This leads to a somewhat more complicated architecture of the hidden layer H which is visualised in Figure 3.3. This architecture is based on two main ideas. The first idea is to separate the Hidden layer output into two parts: The cell state c_t and the hidden layer output h_t. The second idea is to use logical gates to manipulate the input and output data. This manipulation can be separated into the *forget* (f_t), *input*(i_t) and *output*(o_t) modules. The forget and output gates consist of sigmoid layers which return values between 0 and 1. In the case of the forget gate the output values are multiplied by the previous cell state to "forget" certain parts by multiplying them with values close to 0. The output gate does the same to the hidden layer output of the current time step which is generated by applying some non-linear function f_o to the current cell state. The current cell state is determined by additively updating the past cell state after it has passed through the forget gate using the values returned by the input gate. The input gates values are the result of element wise multiplication of the output of a sigmoid layer and a non-linear layer f_i. The formal definition of the LSTM architecture with a $tanh$ activation function is the following:

$$h_t = output_t * tanh(c_t) \tag{3.1}$$

$$c_t = forget_t * c_{t-1} + input_t \tag{3.2}$$

$$output_t = \sigma(W_4 h_{t-1} + U_4 x_t) \tag{3.3}$$

$$input_t = \sigma(W_2 h_{t-1} + U_2 x_t) * tanh(W_3 h_{t-1} + U_3 x_t) \tag{3.4}$$

$$forget_t = \sigma(W_1 h_{t-1} + U_1 x_t) \tag{3.5}$$

$$\tag{3.6}$$

This architecture avoids the vanishing gradient problem in two ways. Firstly the cell state is only updated multiplicatively and additively by the output of layers trained on h_{t-1} and x_t. This avoids updating the cell state by repeatedly passing it through a function. The second factor is the use of sigmoid gates which set different parts of the signal to zero at each time step. The combination of these factors avoids the kind of exponential decay observed in the previous section, which enables the network to learn longer dependencies. Since the cell state c_t has the same shape as the hidden layer output, an LSTM has four times as many trainable parameters as a simple RNN architecture with the same shape of h_t and x_t.

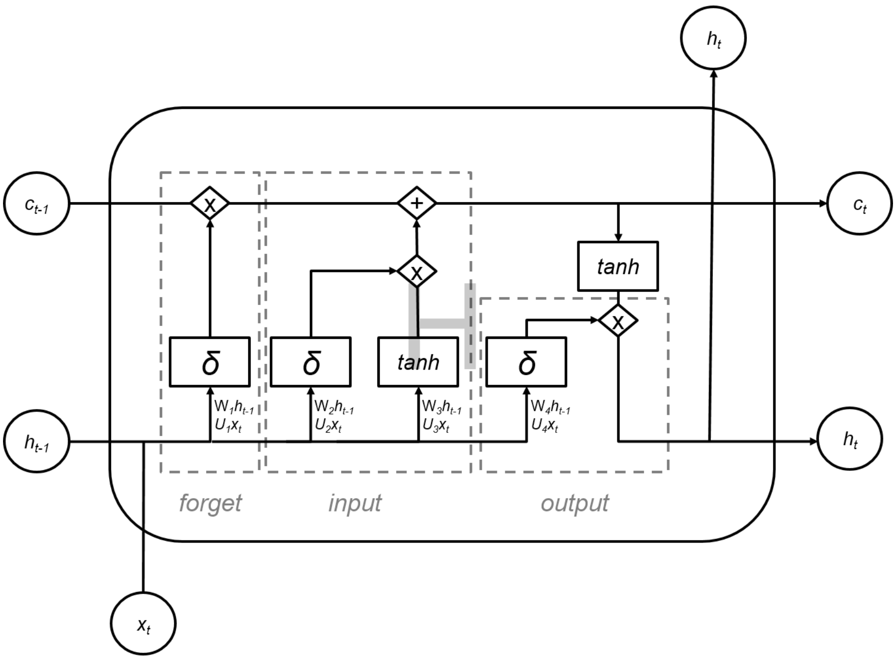

Figure 3.3: Hidden Layer of LSTM Network

Part IV

Literature review

Chapter 4

Literature review

The following paragraphs will give an overview over the current state of research regarding energy price predictions. Since the literature on gas price predictions is rather limited, this review is extended into the areas of electricity and oil prices. These areas of extension are chosen based on the observation that gas lies somewhere in between these two commodities regarding storage, transmission and pricing mechanisms. Electricity cannot yet be stored in an economically sensible way leading to the need to constantly balance demand and supply in each transmission network and strong variability in prices across regions and seasons. Oil on the other hand can be both stored and shipped at relatively low cost leading to a relatively integrated global market with lower seasonality. Natural gas can be stored and shipped at relatively high cost and is mostly transported through local distribution networks akin to electricity grids. Regarding methodology the literature is categorised into three areas: linear models (LR), feed forward neural networks (FFNN) and recurrent neural networks (RNN). The linear category contains both classical Linear regression models as well as time series models such as AR-Type models and generalised linear models such as logistic regression. For each of the different energy products a table is included listing the reviewed sources with the energy market that was modelled in the respective study and the category of the methods used. In the course of this overview the focus will be on the forecasting methods used as well as the choice of predictor variables. Deliberately excluded from this overview is literature that focuses on statistical modelling of energy prices for risk management or purposes other than the explicit forecasting of future prices.

4.1 Electricity

With the rising importance and liberalisation of electricity markets the problem of predicting prices on these markets has received increasing attention by the research community. Apart from regulatory changes another factor that has lead to an increasing price volatility in some electricity markets is the rising share of renewable producers. This has lead to an increased volatility on the supply side of the market. Other factors that make electricity prices unique with respect to other commodities is the lack of storage as well as international integration of markets and different market regimes across countries. Regarding the different markets one can generally divide them into future markets, day ahead auctions and intra day spot markets. Whereas the first type works analogously to other commodity future markets the day ahead and intra day markets are unique to the electricity domain. While there is some deviation between different countries, the day ahead auctions usually work in the way that suppliers place sell offers independently of each other. After the end of the bidding period the market operator clears the market by accepting the cheapest offers up until the estimated demand is met. All sellers whose offers were accepted then receive the price of the highest accepted offer. This price is called the market clearing price (MCP) or sometimes marginal price in the literature and is the focus of virtually all price forecasting research in this area. The intra day markets on the other hand offer continuous trading between market participants up to around 30 minutes to delivery. Intra day markets are often used to correct for unexpected changes on the supply or demand side with respect to the expectations at the day ahead auction. The research referenced in this section is only a small subset of electricity price forecasting literature selected based on relevance and conceptual proximity to the modelling approach in this thesis. For a comprehensive review on electricity price forecasting the reader is referred to *Weron (2014)* which illustrates the extend of research activity in this area with almost 500 publications from 1989 to 2013 and was the starting point for this overview.

Linear Models An early attempt to predict electricity prices using a regression purely based on exogenous variables was done by Schmutz and Elkuch (2004) who used natural gas prices and weather data as input variables. In an even earlier paper Kim et al. (2002) use multiple regression in combination with wavelet decomposition. The time series is decomposed into wavelets for different frequencies and these wavelets are then used as the target variable in the regression. Crespo Cuaresma et al. (2004) have applied various univariate time series models including $AR(1)$ as well as different $ARMA$ models to produce short term predictions of German day ahead electricity prices. Weron et al. (2005) applies a similar approach to Californian electricity prices with a focus on the time period of the power crash 2000/01. Despite these examples of models purely based on external regressors or purely based on past observations of the target variable, most linear models in the literature are models which include both types of input variables. One example of this is the extension of the above mentioned work on predicting Californian electricity prices in Misiorek et al. (2006). Here the authors include the system wide load or its forecast as additional input variable to predict the day ahead prices. Another explanatory variable that is often used in these kind of models is temperature data. Knittel and Roberts (2005) add a third order polynomial of the temperature to an ARIMA model in order to predict Californian electricity prices. The authors find the temperature variables to be highly significant in a model predicting power prices in the time period 1998-2000. A linear model combining a larger number of exogenous variable is presented in Zareipour et al. (2006). In this work the authors predict power prices in Ontario using supply and demand forecasts as well as simulation based price forecasts for the Ontario, New York and New England markets as input variables. They find that adding these variables increases model accuracy for forecasts both 3 and 24 hours ahead relative to a standard ARIMA model. Depending on the local market conditions, production forecasts of electricity sources with a high local market share are also used as input variables. Following this approach water reservoir levels are used to predict prices in Columbia (Lira et al. (2009)) and wind power production forecasts are used for the Nordic electricity market (Kristiansen (2012)).

Feed Forward Neural Networks In one of the earlier works applying artificial neural networks to electricity price prediction Yamin et al. (2004) generate forecasts for the day ahead price in the Californian power market. They use an architecture with a single hidden layer and one output node for each of the 24 hours. As input parameters they propose various variables describing the status of power lines, load patterns, power outages as well as different time factors such as the week day and conclude that this approach outperforms traditional methods. Gareta et al. (2006) applies a similarly simple feed forward architecture with just one hidden layer. However the authors do not train one model with 24 output dimensions but cluster the hours according to variable importances and train models with smaller output dimensions on each of these clusters. Although the authors report satisfactory accuracies on their test set, they do not include reference models for comparison. In Cruz et al. (2011) feed forward neural networks are evaluated in comparison to a variety of alternative models including ARIMA, exponential smoothing and dynamic regression. The authors fit one univariate model to the data for all hours and include the hour of the day as an input variable. Beyond this variable a further eight variables are chosen from a wider range consisting of demand forecasts and various past values of the electricity price. Based on the mean absolute percentage error for predictions on data from the Spanish electricity markets for the years 2007 / 2008 the neural network approach has only mixed success being ranked in the middle of the alternative approaches. In a more recent study Panapakidis and Dagoumas (2016) compare different neural networks that both vary in their architectures as well as input variables. In this study models solely based on past observations are compared with models incorporating demand, natural gas prices or market data from neighbouring countries. While these models are again of a simple single layer architecture, the authors also implement a stacked approach where they train a neural network to combine the predictions of these separate models into an improved price prediction. In fact this stacked model beats each of the base models on the test data.

Recurrent Neural Networks Recurrent neural networks have only more recently been used for electricity price prediction and there exists overall less literature based on this approach compared to feed forward neural networks. In Anbazhagan and Kumarappan (2013) the authors use an Elman architecture recurrent neural network to predict electricity prices in both Spain and New York. The model in this work takes as sole input variables the prices for the respective hour of the last 16 days. This model is compared with time series methods such as AR, ARIMA and feed forward neural networks with past prices as inputs. While the RNN in their test scenario outperforms more traditional modelling approaches such as ARIMA it has a higher MAPE than hybrid approaches such as a combination of a neural network and wavelet transformation. In Mirikitani and Nikolaev (2011) the authors develop a new maximum likelihood based approach to train a fully connected recurrent neural network to predict electricity prices in Spain and Ontario. Using only past prices as input variables and 6 nodes in the hidden layer they achieve significant improvements over the reference methods (FFNN, ARIMA) regarding the mean absolute percentage error in both markets. The approach presented in Sharma and Srinivasan (2013) combines an RNN with a so called Fitz-Hugh Nagumo model to accommodate large price spikes and applies a feed forward neural network to predict the residuals from this model. The authors benchmark their model against the models of Mirikitani and Nikolaev (2011), Zareipour et al. (2006) on the respective datasets and find significant error reductions in the area of 10-30 percent of the original reported error. Overall RNNs are one of the most intensely researched modelling approaches for electricity price forecasting at the moment with some very promising results. Another observation one can take away from the literature is that many authors use RNN as a ”pure time series” model, meaning they limit the input variables to past observations of the target variable and do not include exogenous variables.

Source	Market	Category
Schmutz and Elkuch (2004)	Italy	LR
Crespo Cuaresma et al. (2004)	Germany	LR
Weron et al. (2005)	California	LR
Contreras et al. (2003)	Cal., Spain	LR
Misiorek et al. (2006)	California	LR
Nogales et al. (2002)	Cal., Spain	LR
Nogales and Conejo (2006)	Spain	LR
Knittel and Roberts (2005)	California	LR
Zareipour et al. (2006)	Ontario	LR
Weron and Misiorek (2008)	Cal., Nordic	LR
Lira et al. (2009)	Columbia, Nordic	LR
Kristiansen (2012)	Nordic	LR
Yamin et al. (2004)	California	FFNN
Gareta et al. (2006)	Spain	FFNN
Cruz et al. (2011)	Spain	FFNN
Shafie-khah et al. (2011)	Cal., Spain	FFNN
Panapakidis and Dagoumas (2016)	Italy	FFNN
Anbazhagan and Kumarappan (2013)	Spain, New York	RNN
Mirikitani and Nikolaev (2011)	Spain, Ontario	RNN
Sharma and Srinivasan (2013)	Spain, Ontario	RNN

Table 4.1: Electricity Price Forecasting Literature

4.2 Oil

Being one of the most widely traded commodities and the leading primary source of energy it is not surprising that the problem of forecasting oil prices has received a lot of attention by the scientific community. Unlike electricity, the oil market is globally integrated and has been relatively lightly regulated for a long time. This has lead to the fact that research activity regarding forecasting of oil prices reaches further into the past compared to electricity prices which became interesting only with the liberalisation of the respective markets. Another difference is the fact that oil can be stored at relatively low costs and there is no need to constantly balance demand and supply. The target variables in oil price prediction can be categorised according to two aspects: the type of oil (WTI, Brent) and the type of market or financial product (Spot, Future).

Linear Models The longer history of research in this area, coupled with the higher popularity of linear modelling techniques in earlier years leads to the fact that overall a higher share of literature on oil price forecasting focuses on linear models. In the studies Ye et al. (2002), Ye et al. (2005) and Ye et al. (2006) the same group of authors develop models to forecast the WTI spot price based on OPEC inventory levels. However instead of just taking absolute values of this variable as input they use various transformations such as deviations of the inventory level from average values for this time of the year as well as year on year changes. The authors reach good forecasting results for testing periods throughout the 1990s and early 2000s. They also point to the long history of studying the relationship between commodity prices and inventory levels which reaches back nearly a century (Ye et al. (2006), p.1). In a working paper for the European Central Bank (Dées et al. (2008)), the authors predict WTI spot prices using forward prices, inventories and OPEC capacity utilisation. Regarding test data from the period of 2004-2006 the authors find their model to outperform both a random walk benchmark as well as oil price expectations implied by the futures market. An example of using exogenous variables to explain changes in oil future prices is the work done in Bu (2011), where the author uses reported trading positions of speculative traders as regressors and a GARCH model for the residuals.In Moshiri and Foroutan (2006) the same target variable is predicted using an ARMA model purely based on past observations of the WTI future price, which is however outperformed by the other models in his study.

Feed Forward Neural Networks The model beating the ARMA approach in Moshiri and Foroutan (2006) is indeed a feed forward neural network trained on past prices. This network with five neurons in the hidden layer is found to significantly outperform ARMA and GARCH models for daily test data in the time period 2000-2003. In Haidar et al. (2008) the past future and spot prices are used in a neural network to predict the next days spot price and outperforms an alternative network trained with gold and stock prices as input. The authors of Alizadeh and Mafinezhad (2010) use a wide range of input variables for their network including the US Nominal Effective Exchange Rate, refinery capacity and historical prices. As is the case for electricity price prediction, many more recent papers that apply neural networks to oil price predictions focus on hybrid approaches. One approach that is especially popular is combining machine learning methods with a wavelet transformation of the price series. Two examples of this trend are (Pang et al., 2011) and He et al. (2012), both of which manage to reduce model errors following this approach. An example of applying neural networks to the prediction of Oil Future prices is the work done in Shambora and Rossiter (2007).

Recurrent Neural Networks In Mingming and Jinliang (2012) the above mentioned work integrating neural networks with wavelet transformations is extended to recurrent neural networks. The authors find that the resulting model outperforms standard neural networks in forecasting WTI spot and gold prices. A recent example of applying a recurrent neural network directly to oil price data without transformation is presented in Wang and Wang (2016). In this work the authors use an Elman architecture to predict oil markets for the WTI spot and Chinese markets as well as stock prices. They find that this architecture outperforms the reference feed forward neural network significantly regarding multiple metrics.

Source	Market	Category
Ye et al. (2002)	WTI Spot	LR
Ye et al. (2005)	WTI Spot	LR
Ye et al. (2006)	WTI Spot	LR
Dées et al. (2008)	WTI Spot	LR
Bu (2011)	WTI Future	LR
Moshiri and Foroutan (2006)	WTI Future	LR, FFNN
Haidar et al. (2008)	WTI Spot	FFNN
Alizadeh and Mafinezhad (2010)	Brent Spot	FFNN
Shambora and Rossiter (2007)	WTI Future	FFNN
Amin-Naseri and Gharacheh (2007)	WTI Spot	FFNN
Pang et al. (2011)	WTI Spot	FFNN
He et al. (2012)	WTI Spot	FFNN
Wang and Wang (2016)	WTI Spot	RNN
Mingming and Jinliang (2012)	WTI Spot	RNN

Table 4.2: Oil Price Forecasting Literature

4.3 Natural Gas

Compared to both electricity as well as oil prices there is a lot less research activity and literature in the field of natural gas price prediction. In fact a lot of the literature that features statistical modelling of gas prices focuses more on explaining certain market characteristics rather than explicit price forecasting. An area of particular attention in this kind of research has been focused on the connection between oil and natural gas prices (Villar and Joutz (2006), Hartley et al. (2008)) and how this connection has changed since the shale boom in the US market (Geng et al. (2016), Caporin and Fontini (2017)). Another research question, which has been intensively researched in most other commodity markets, is the connection between future and spot prices of natural gas (Herbert (1993), Chinn and Coibion (2014)).

Linear Models The work done in Mishra and Smyth (2016) extends the question of the Spot-Future relationship and tries to derive trading strategies for the spot market based on the current future price. Some of the models used in this work are equivalent to a linear regression of the spot price on its past value and the interaction with the logarithmic ratio between future and spot price. One of the earliest attempts at natural gas price forecasting has been implemented in Buchanan et al. (2001) where the authors implement a logistic regression based on trading positions to predict the direction of spot prices changes. In Woo et al. (2006) the authors use a linear regression to predict local natural gas spot prices in California based on prices observed for the Henry Hub trading point. An example of a linear model using the natural gas future price as target variable has been developed in Mu (2007) where the authors use a linear regression with weather data, oil price returns and treasury yields to model the return series of the front month Henry Hub future. One of the models presented in Malliaris and Malliaris (2008) uses a linear regression with price data from other energy commodities such as crude oil and heating oil as regressors to forecast natural gas spot prices.

Feed Forward Neural Networks In Nguyen and Nabney (2008) and Nguyen and Nabney (2010) the authors combine a wavelet transformation and several filters with various modelling techniques to predict natural gas forward prices on the UK market. While they included neural network models they found them performing worse then linear regression and GARCH models with wavelet transformation. A similar experiment is conducted in Jin and Kim (2015) using weekly spot prices at the Henry Hub as target variable. Here the authors find the exact opposite results with neural networks outperforming ARIMA. However in both cases wavelet decomposition seems to improve model performance for almost any type of model. A feed forward neural network on past price values is combined with a feature selection algorithm in Salehnia et al. (2013) to predict the Henry Hub spot price. In this study the neural network is found to be more efficient at short-term forecasting than the reference models. Another paper that concentrates on the application of Feature Selection methods to this problem is Čeperić et al. (2017). While the authors in this paper find that both for FFNN and SVR models feature selection improves performance they still only slightly outperform classical time series models. The authors also observe that the power of machine learning models in this area is generally exaggerated in the literature. The authors of Panella et al. (2012) apply various different feed forward neural network architectures with past prices as input to predict spot prices and find that all those models outperform the reference GARCH model.

Recurrent Neural Network Judging from the examined literature research, the application of recurrent neural networks to natural gas price predictions seems to be extremely limited. In fact the only paper found to apply a recurrent architecture to this problem is Busse et al. (2012). In this study the authors use a recurrent architecture called *Nonlinear Autoregressive Neural Network with eXogenous inputs (NARX)* to predict gas spot prices at the NetConnect hub in Germany. They find that the optimal selection of input variables to the model are temperature forecasts, the USD/EUR exchange rate and the past prices at the NCG, NBP and TTF hubs. The proposed model significantly outperforms a naive prediction based on the last available price value.

Source	Market	Category
Woo et al. (2006)	Cal. Spot	LR
Mu (2007)	HH Future	LR
Malliaris and Malliaris (2008)	HH Spot	LR, FFNN
Nick and Thoenes (2014)	NCG Spot	LR
Nguyen and Nabney (2008)	NBP Future	FFNN
Nguyen and Nabney (2010)	NBP Future	FFNN
Abrishami and Varahrami (2011)	HH Spot	FFNN, LR
Panella et al. (2012)	HH Spot	FFNN
Jin and Kim (2015)	HH Spot	FFNN
Čeperić et al. (2017)	HH Spot	FFNN
Salehnia et al. (2013)	HH Spot	FFNN
Busse et al. (2012)	NCG Spot	RNN

Table 4.3: Gas Price Forecasting Literature

4.4 Summary

From the review of the above mentioned literature one can gain the following conclusions:

Research on natural gas: Despite the growing importance of natural gas in the global energy market, natural gas prices have received significantly less interest from the forecasting research community than oil and electricity prices.

Application of LSTM: Long short term memory networks have not yet been widely applied in the forecasting of energy commodity prices.

Comparability: Especially for electricity and natural gas prices the results of the different studies are hard to compare to each other. This is due to the fact that the analysed time series do not only vary in time frame and resolution (daily, weekly etc.) but also refer to different geographical markets.

Evaluation methods: Another factor that makes the interpretation of the results of the different studies harder is the way the authors evaluate their results. Especially the choice of reference model varies significantly across the literature. Many authors only compare different machine learning or regression models among each other and do not benchmark their models against the lagged price value. This limits the extend to which the practical use of these models to market participants can be assessed.

Usefulness of neural networks: Especially in the area of natural gas price forecasting, no clear picture on the performance of neural networks has emerged yet. While some studies suggest relatively good performance others see neural networks outperformed by classical time series methods.

Feature selection and wavelet decomposition: A relatively consistent finding across the different types of commodities is the positive impact of both feature selection and wavelet decomposition of the time series.

While the first two of these observations can serve as additional justification of the choice of topic as a relatively sparsely researched one, the other observations should be kept in mind to put the results of this thesis in perspective.

Part V

Empirical study

Chapter 5

Data

The data used for this analysis is downloaded from the *Thomson Reuters Eikon* database. For this purpose a Python script was implemented that downloads all necessary time series through the *Eikon Web and Scripting APIs*. The following paragraphs give an overview over the structure and content of the data both on the target variable as well as the input variables used for the model.

5.1 Target Variable

5.1.1 Price Level Prediction

The target variable to be predicted in this prediction problem is the closing price for the TTF gas future with delivery in the next month, referred to as front month. An advantage of this specification of the target variable is, that one gets one continuous time series for which data can be downloaded for a long period of time. However this definition also poses a significant challenge. The challenge lies in the fact that the product, which is traded, effectively changes at the change of the month as the delivery period changes. For example the Front Month price quoted on September 30 refers to natural gas delivered in the month of October, whereas on October 1 it refers to delivery in November. Due to the strong seasonality in the natural gas price this can lead to structural breaks and large price changes at the end of each month. As an illustration of this problem both the front month price as well as the price of each months future as traded before it becomes the Front are plotted for the second half of 2016 in figure 8.1.

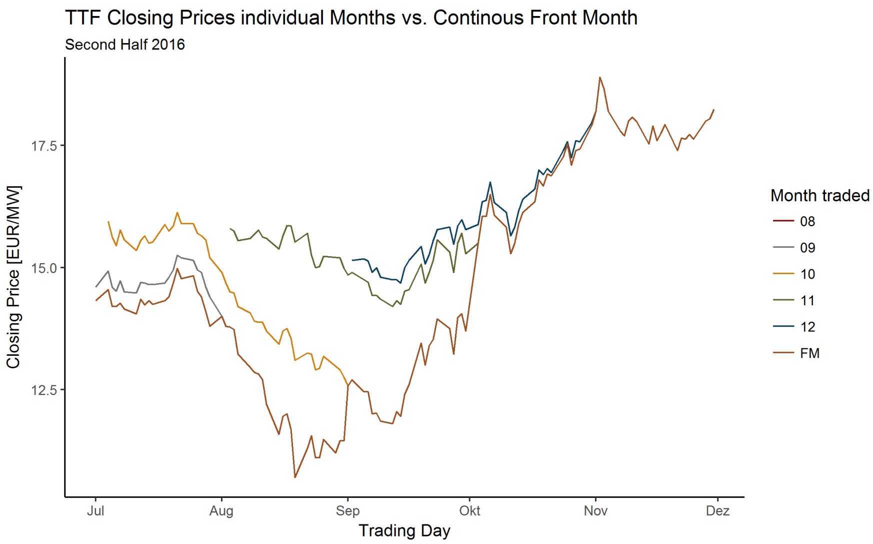

Figure 5.1: TTF Front Month vs. Individual Months

As one can see the price of the front month jumps at the end of each month to the level at which the next month was traded previously. This means, that while in this thesis the front month is referred to as one continuous target variable, it might also be viewed as a collection of separate time series. To ensure adequate performance of the models, this idea is taken into consideration in the implementation of the model training. That means that, when generating predictions for observations at the beginning of each month past price values of the monthly future for this delivery period are used as input variables. This is done instead of choosing the price values of what was the front month before to prevent the structural breaks a the turn of the month from affecting model performance.

To get a first overview of the inter temporal dependencies across time in the target variable as well as a hint on the performance of the autoregressive reference models both the autocorrelations as well as partial autocorrelations of the TTF front month are plotted in figures 5.2 and 5.3. The very high values of all auto correlations as well as the near zero partial autocorrelations for lags above one day suggest modelling the target variable using an AR(1) model, which will probably result in a parameter estimate very close to one.

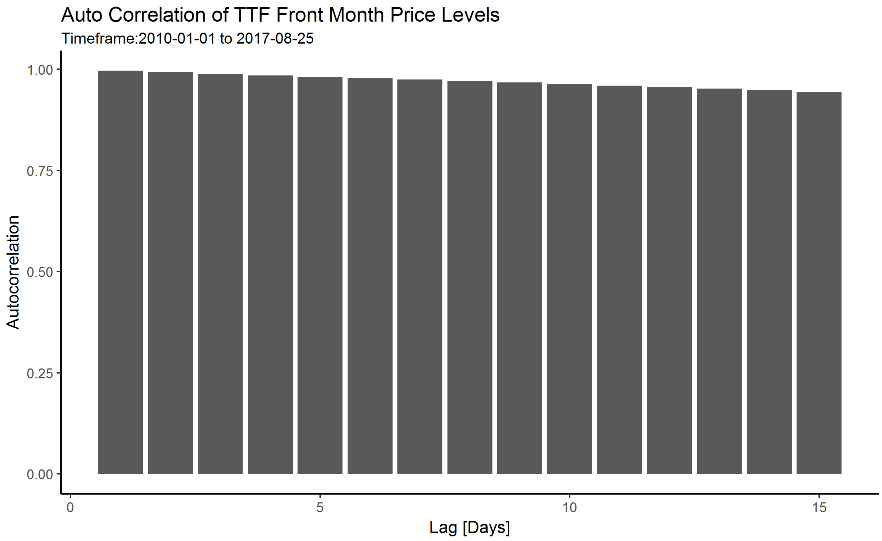

Figure 5.2: Auto Correlations TTF Front Month Closing Prices

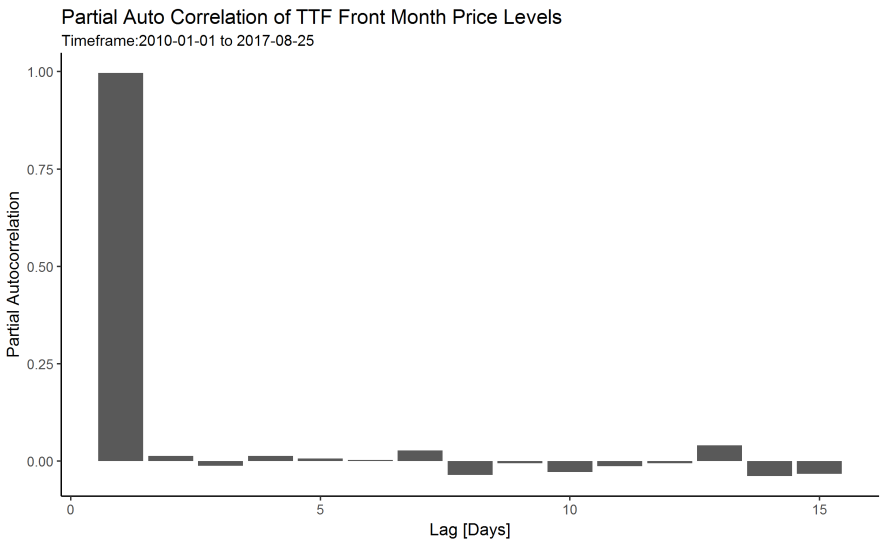

Figure 5.3: Partial Auto Correlations TTF Front Month Closing Prices

5.1.2 Binary Prediction

The target variable of the binary prediction problem can best be described as the event: ”Today’s front month closing price is minimal among all remaining trading days of this month.” (True = 1, False = 0). This target variable can be generated from the price level data in the following way, where the index d denotes the trading day within each month and the index M the month. I is the indicator function return one or zero depending on weather the statement in the argument is true or false:

$$y_{d,M}^{binary} = \prod_{i>=d} I(y_{d,M}^{level} \leq y_{i,M}^{level})$$

To avoid misunderstanding it is important to note that this variable does not correspond to the question of whether a certain trading day is minimal among all days of that month. In fact there is no theoretical limit on the amount of positive observations of this variable. To illustrate this figure In case of constantly rising prices all observations of this variable would be positive. 5.4 shows the binary target variable and the front month price level for a month with a lot of positive observations. For the whole data around

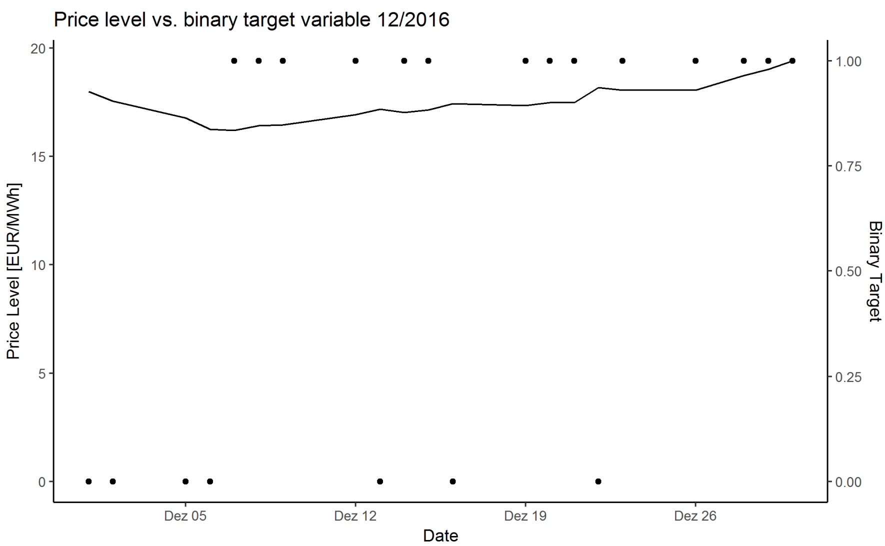

Figure 5.4: Binary target variable and price level for December 2016

a quarter of the observations are positive, however this share can vary significantly as shown in table 8.1. It is important to note that the value of this variable at any point in time contains information regarding future prices that is not available at that time point. Therefore the value of that variable for each trading day can only be observed after the end of the month and not on the trading day itself. For this reason values of this target variable are never used as an input variable and the univariate models in the binary prediction case also take the past values of the price level as inputs.

5.2 Input Variables

The following paragraphs will briefly describe all potential input variables that were included in the variable selection. These variables can be sorted into three categories: Energy commodity prices, gas market fundamentals and exchange rates, each of which will be illustrated separately. The initial choice of candidate variables was based on input from energy traders at BASF SE. The development of each input variable across time is plotted in figures 8.3 - 8.18.

5.2.1 Energy Commodity Prices

The variables conceptually most closely related to the target variable are surely prices of other energy commodities. As argued in section **??** natural gas shares characteristics with both oil and electricity markets. Therefore it is somewhat natural to include prices for these commodities in addition to natural gas prices for other hubs and delivery periods.

TTF Day Ahead /Spot: At the TTF Day Ahead (TTFDA) market, which is sometimes also referred to as the spot market, natural gas is traded every day for delivery during the next day. This market is one of the most liquid markets at any gas trading hub and enables market participants to react to daily fluctuations in both demand and supply.

NBP Front Month: The front month price at the National Balancing Point is the equivalent of the target variable for the UK market. As explained above this is the oldest, most established and most liquid virtual trading point in Europe. Unlike the TTF, which is traded in Euro, this product is traded in British Pound and therefore the relative price difference between these markets can be influenced by the exchange rate.

Oil Front Month: This time series describes the price of Brent crude oil with contract conditions equivalent to that of the target variable. This product is priced in US Dollars.

Electricity Base Load Front Month: This is the price of the so called *Phelix* Base Future for delivery of electricity the next month from 0:00 - 24:00.

Electricity Peak Load Front Month: This price is equivalent to the base load price with the difference that the delivery is limited to the peak hours 09:00 - 20:00. Both base and peak load futures are traded in Euros.

All energy commodity prices are represented by daily closing prices.

5.2.2 Gas Market Fundamentals

The second group of input variables contain data on the four main fundamental factors presumed to drive the natural gas market: consumption, production, storage and pipeline flows.

Consumption: The consumption can be divided into two types. The first type describes the amount of natural gas consumed through Local Distribution Zones (LDZ), this is the channel through which residential and the vast majority of industrial users receive their gas. The Non-LDZ consumption describes the demand of users directly connected to the high pressure gas grid, which are mostly power stations and very large industrial consumers. Both demands are used as separate input variables on a daily resolution for the Dutch market.

Production: Daily output of British gas wells on the Continental Shelf as well as daily output of all Dutch fields are aggregated into one variable each and used as separate input variables.

Storage: Natural gas storage levels in the Netherlands and UK are each used as a separate variable. These storage levels are highly seasonal since depots are usually filled during the summer months and then drained to fulfil the higher demand in the heating season.

Pipeline Flows: The *Interconnector* (IUK) and *Balgzand Bacton Line* (BBL) are the two main natural gas pipelines connecting the British and European markets. Net gas flows on both pipelines are additional sources of demand or supply on the European market and are therefore included as additional variables.

5.2.3 Exchange Rates

As mentioned above, some of the commodity prices included as inputs are denoted in US Dollar or British Pound. Therefore the exchange rates of both currencies to the Euro are included as daily closing prices to enable the models to react to the effects of foreign exchange rate volatility.

5.2.4 Data Availability

In table 5.1 one can see that the different categories of input variables vary strongly regarding the amount of data available. Whereas the commodity prices and exchange rates are available starting in 2010, the gas market fundamentals data starts between 2012-2014 with many variable providing around half as many non missing observations as the target variable. When training, tuning and evaluating models the data will be limited to complete observations where none of these variables is missing. This significantly reduces the amount of data available to each model and poses challenges to models with a lot of trainable parameters.

Variable	MinDate	MaxDate	NumObs
ElectricityBaseFM	2010-01-01	2017-09-01	1979
ElectricityPeakFM	2010-01-01	2017-09-01	1979
NBPFM	2010-01-01	2017-09-01	1977
OilFM	2010-01-04	2017-09-01	1969
EURUSDFX	2010-01-01	2017-09-01	1981
EURGBPFX	2010-01-01	2017-09-01	1981
TTFFM	2010-01-01	2017-09-01	1981
TTFDA	2010-01-04	2017-09-01	1966
ConLDZNL	2014-01-01	2017-09-01	949
ConNLDZNL	2014-01-01	2017-09-01	949
ProdNL	2012-09-03	2017-09-01	1287
ProdUKCS	2012-09-03	2017-09-01	1284
StorageNL	2013-01-02	2017-08-30	952
StorageUK	2014-01-01	2017-08-30	946
TradeBBL	2014-01-01	2017-09-01	949
TradeIUK	2012-09-03	2017-09-01	1287

Table 5.1: Data Availability of all Input Variables

Chapter 6

Experimental Design

While designing an experiment to evaluate the performance of the LSTM model in comparison to reference models one has to solve several problems regarding the following topics:

1. Preprocessing
   - How should missing data be treated ?
   - Should input and output features be scaled, if yes how?
2. Reference models
   - Which models should be selected as references ?
3. Parameter tuning
   - How should the tuning process be designed (Train / validation split, tuning grid)?
   - Which hyper parameters should be tuned and which should be set to a fixed value ?
4. Variable selection
   - How should the selection process be designed (Forward vs. backward selection)?

5. Model evaluation
    - Which data should be selected as test data by excluding it from parameter tuning and variable selection ?
    - How should the selection process be designed ((Train / test split)?

A major challenge arising from this problem is the interdependence between the parameter tuning and the variable selection steps. To tune the parameters one has to have selected a certain set of input variables and vice versa. One way to solve this problem would be to view the subset of selected input variables as just another tuning parameter and integrate both steps. However in this process the number of training iterations that would have to be performed would increase significantly to the product of the iterations in both steps. To avoid excessive computational costs the following alternative approach was chosen:

1. Perform parameter tuning on *univariate* models
2. Perform variable selection using parameter values determined in step 1.
3. Tune parameters again for models using input variables determined in step 2.
4. Evaluate both the tuned univariate models from step 1 as well as multivariate models from step 3.

Each of these steps is executed both for the LSTM model as well as for each reference model for which it is applicable. A *univariate* model in this context is a model trained using only past observations of the front month price as input, whereas multivariate models use both the front month price as well as at least one additional input variable. In the following paragraphs the selected approach for each of the above mentioned decision problems is outlined.

6.1 Preprocessing

Due to the high quality of the time series data on the *Thomson Reuters Eikon* data platform, the amount of preprocessing necessary for this prediction problem was relatively limited. The preprocessing steps that are probably most interesting and have the most effect on the results were the treatment of missing data and the feature scaling for the price level prediction.

6.1.1 Treatment of missing data

As mentioned in section 5.2.4 the majority of incomplete observations are due to the fact that some variables are available only from a relatively late starting point relative to the target variable. To ensure that the different models are trained on the same amount and quality of data independent of the input variables of each model, all observations for the time period where some of the variables have not yet been observed at all is dropped from the training and testing data set entirely. The cases of missing data for some variables after that time period are mostly due to different number of trading days for the various financial assets as well as some variables not being recorded on a daily basis. Due to this fact and the observations that variables are rarely missing for more than a few days in a row a forward filling replacement strategy has been chosen, replacing missing observations with the last non-missing observation of that variable.

6.1.2 Feature scaling

For both the binary as well as the price level prediction problem the input data is scaled to values in the interval $[-1, 1]$ using the Min-Max scaling approach as implemented in the *scikit-learn* package. This scaling approach is based on the following formula:

$$\tilde{x}_t = 2\frac{x_t - \min_{t \in T_{train}}(x_t)}{\max_{t \in T_{train}}(x_t) - \min_{t \in T_{train}}(x_t)} - 1$$

Note that the minimum and maximum is determined on the training data only to avoid including information from the test dataset. Although theoretically this could lead to values of the target variable outside of the desired interval in the test set changing this approach has not shown any effect on the results and therefore the more rigorous approach with regard to train-test separation has been chosen. For the price level prediction the target variable is also scaled using this approach in order to ensure that it is within the value range of the $tanh$ function.

6.2 Reference Models

The predictions of the LSTM model are compared to each of the following reference models:

Simple Recurrent Neural Network (RNN): This model is the implementation of the above outlined simple RNN architecture where the hidden layer output of the past time step is used as another input variable.

Feed Forward Neural Network (FFNN): The FFNN model is an implementation of the classical Neural Network without any recurrent connection, modelling a static input output mapping.

Regression Models: This reference model consists of a Feed Forward Neural Network without any hidden layer. The model therefore just applies the output activation function on a linear combination of the input variables. In the case of the binary prediction problem where a logistic activation function is used, this is equivalent to a logistic regression, with the difference that the model is trained using gradient descent instead of maximum likelihood. In the case of the price level prediction the $tanh$ function is applied to the linear combination resulting in a somewhat unconventional variant of a generalised linear model.

Lagged Price Level: This reference forecast for the price level prediction is generated by just choosing today's price level as prediction for the closing price of the next trading day. It therefore represents the market expectation of the price.

Equal Distribution: This reference for the binary prediction problem is based on the assumption that all remaining trading days are equally likely to deliver the minimal closing price. Therefore the probability forecast for the current trading day to be minimal is:

$$\tilde{y}_{t,M}^{binary} = \frac{1}{N_{t,M}^{remaining}}$$

In this case $N_{t,M}^{remaining}$ represents the number of trading days that remain in month M starting from and including trading day t.

6.3 Parameter Tuning

The detailed structure of each model as well as the training process is controlled by a number of hyper parameters. The selection of parameters, that are considered in this paragraph, as well as the terminology is inspired by the *Tensorflow* and *Keras* Python packages, which were used to implement the neural network based models. Theoretically the models could be tuned on all of these parameters. However using the standard grid search used in this tuning approach the total number of parameter combinations would grow exponentially. Since each of the models have to be retrained on every parameter combination this would generate excessive computational costs. To avoid these costs the parameters are separated into a set of fixed hyper-parameters which are set to constant values derived from experience and literature and a complimentary set of tuning parameters.

6.3.1 Fixed Hyper-Parameters

This section will give a brief description of each hyper-parameter that was excluded from the tuning process. This selection was done based on observations from literature as well as the online machine learning community. Except for the *length* parameter none of these parameters alter the actual structure of the respective model but are limited to the training process. The *length* and *batchsize* parameters can be considered features of the *Tensorflow* package and are not necessarily present in the theoretical work on neural networks or alternative implementations.

Length: From the definition of a recurrent neural network there is no theoretic limit on the length of inter temporal dependencies that the model can learn. This is due to the fact, that the gradient as well as the layer activations can flow infinitely far through the network. Regarding the implementation this would also mean that there is no limit on the amount of calculations that would have to be done to determine the gradient based updates to the network weights. Therefore the *Tensorflow* implementation of recurrent neural networks requires the user to pass a length argument which limits the number of time steps that are included in the calculation of the gradients for each observation. This length thereby also limits the maximum length of inter temporal dependencies that the model can learn.

Batchsize: The *Tensorflow* package processes the training data in batches. In practice that means that at a given training epoch the gradients are calculated for a certain subset of training observations at once. Then the weights are updated using the gradients calculated from this subset. The *batchsize* parameter controls the size of this subset. Therefore in each training step the weights are updated $\frac{n_{obs}}{batchsize}$ times (rounded up to the next integer). Given the same number of training steps a larger *batchsize* therefore means less frequent weight updates, slower learning and less computational complexity.

Epochs: The *epochs* parameter controls the number of training steps, which is the number of times the weight updates are repeated for each training batch. Therefore the total number of weight updates is $epochs * \frac{n_{obs}}{batchsize}$.

Loss: The *loss* parameter determines the loss-function that is minimized during the training of the model. In this work the *Mean Squared Error* (MSE) is used as loss for the price level prediction whereas *Binary Cross Entropy* (BCE) is used for the binary prediction problem. These loss functions are defined in the following way:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y_i})^2$$
$$BCE = -\frac{1}{n}\sum_{i=1}^{n}(y_i)log(\tilde{y_i}) + (1 - y_i)log(1 - \tilde{y_i})$$

Optimizer: The *optimizer* parameter chooses an optimization algorithm to use when minimizing the loss function on the training set. Following practice and advice from the machine learning community the *Stochastic Gradient Descent* optimizer is used for the FFNN models. Following the advice in the *Keras* documentation the *RMSProp* optimizer is used for recurrent models. This optimizer is an extension of the *Stochastic Gradient Descent* method that divides the learning rate at any given iteration by the recent average magnitude of the gradient for each weight.

Output Activation: The output activation is the function that is applied to the linear combination of hidden layer outputs fed into the output layer. This activation function depends on the target variable. For the binary target variable the $logit$

function is chosen to generate predictions in the interval $[0, 1]$. In the case of the price level prediction a $tanh$ activation is chosen generating predictions in the interval $[-1, 1]$ which is the value range of the scaled target variable.

Hidden Activation: For the activation function on the hidden layer the $tanh$ function is chosen for all networks in both prediction problems.

6.3.2 Tuning Parameters

In the parameter tuning stage each model was tuned over the following parameters

Learning Rate: The learning rate is the value by which the gradient values in each iteration are multiplied to get the weight updates in the standard *Stochastic Gradient Descent* method. This means that the weight value at epoch $n + 1$ ($W_{ij,n+1}$ is calculated as $W_{ij,n+1} = W_{ij,n} - learningrate * \frac{dl(W_{ij,n})}{dW_{ij,n}}$. A lower learning rate slows down the learning process and increases the number of training epochs necessary to converge on a local minimum of the loss function. If the learning rate is too high on the other hand the gradient descent algorithm might not converge at all, constantly "over-shooting" the minimum.

Dropout: One way to avoid the issue of over-fitting is to randomly drop different input variables at each time step. Using this approach the value of the dropout tuning parameter sets the probability with witch any individual input variable is dropped. Dropout is only tuned in multivariate models whereas in the univariate cases it is set to zero.

Hidden neurons: This parameter sets the number of neurons in the hidden layer. In the case of the LSTM network it sets the number of neurons for every trainable layer in all gates. Therefore the total number of neurons in the hidden layer of the LSTM is four times that number.

6.3.3 Tuning Process

For the parameter tuning a simple grid is expanded across the selected values for each tuning parameter. The model is then trained on all data up to December 2015 and evaluated using data from 2016 separately for each parameter combination. At the end the parameter combination with the best performance on the test set is selected.

6.4 Variable Selection

To select the most relevant input factor among all variables described above a forward selection method is used. This means that in each iteration of the process separate models are trained adding each of the remaining candidates to the model. Using the same train-test split as in the tuning process the most promising variable is selected and removed from the set of candidates for the next iteration. This is repeated until five variables have been added. Afterwards the model configuration with the best performance among all sets of selected variables of size one to five is chosen. Note that in all cases the training and test data is limited to complete observations containing values for all candidate variables. This is done to ensure that differences in performance can be attributed to the model choice as opposed to differences in data availability.

6.5 Model Evaluation

In the model evaluation phase the performance of the LSTM is compared to each of the reference models listed above both in its univariate as well as multivariate version. For the univariate version the parameter configurations are used that were determined in the first tuning step, whereas the parameters and input variables of the multivariate models are set according to the results of the second tuning iteration and the variable selection steps. Evaluation is done in a month wise rolling prediction, meaning that for each month from January - July 2017 the model is retrained on all data previous to that month and then tested based on its prediction for that month. Afterwards the mean value of the evaluation metric is calculated across the test months.

Chapter 7

Results

In the following paragraphs, the results of the above described experiment will be presented and the performance of each model evaluated accordingly for both the price level as well as the binary prediction problem. The interpretation of these results will be left to section 8.

7.1 Price Level Prediction

As mentioned above, the tuning and variable selection steps were done with a single train test split. This resulted in a training data set that contained 744 observations and a test data set of 372 observations.

7.1.1 Univariate Parameter Tuning

In the first step where the univariate version of each model is tuned over the number of hidden neurons and the learning rate, the chosen parameter combinations differ significantly among the different model types. For example for the learning rate the recurrent models end up with a value of 0.001 whereas the feed forward neural network and the neural network based regression are tuned to higher values. Apart from the difference in the model architecture these results might actually be influence by the different optimizers chosen for each type of model. Regarding the number of hidden neurons it should be noted that this parameter is not tuned for the regression model since by definition this model does not contain any hidden neurons. With regards to this parameter the LSTM is tuned to the highest value in the parameter grid whereas the other two neural networks are tuned to the lowest one. Combined with the effects of the LSTM architecture this causes the LSTM to contain by far the highest number of trainable parameters with a total of 4385 against 89 and 25 parameters for the RNN and FFNN models.

Model	HiddenNeurons	Dropout	LearningRate	MSE
LSTM	32	0	0.0010	0.3908
RNN	8	0	0.0010	0.1668
FFNN	8	0	0.1000	1.1153
Regression	0	0	0.0100	1.5947

Table 7.1: Selected parameter combinations in univariate tuning step of price level prediction

7.1.2 Variable Selection

Table 7.2 shows the best performing variable combinations of each model when iteratively adding up to five additional variables in a forward selection manner. When analysing the results it is important to keep in mind that using this approach starting with 15 candidate variables each model was trained 65 times, which is much higher than the number of parameter combinations in the parameter tuning step. Since best model is chosen over a larger set, it is therefore not surprising that the MSE values are much lower than in those steps. Regarding the different model types it can be observed that the recurrent models are tuned to a lower number of additional input variables than the FFNN and Regression model. UK Storage levels, the front month future at the NBP hub and the electricity base load front month are variables that are selected as inputs across different model types. This indicates that they might actually contain some predictive value as opposed to being just a chance selection resulting from random fluctuations in the tuning process.

Model	Variables	MSE
LSTM	TTFFM OilFM	0.1738
RNN	TTFFM StorageUK ConNLDZNL	0.1502
FFNN	TTFFM ProdUKCS ElectricityBaseFM StorageUK NBPFM	0.1424
Regression	TTFFM ElectricityBaseFM NBPFM StorageUK TTFDA	0.1569

Table 7.2: Selected variable combinations in variable selection step of price level prediction

7.1.3 Multivariate Parameter Tuning

In the multivariate tuning the same approach is used as in the univariate case with the difference of adding the dropout as additional tuning variable and using multivariate models with the input variables selected above. Again one can see somewhat similar results to the univariate tuning step regarding the selection of the learning rate. However in the case of the number of hidden neurons the LSTM is tuned to a lower number whereas the chosen number of neurons for both the RNN as well as FFNN increases. Dropout seems not to bring any additional benefit to model performance with it being tuned to zero for all models but the LSTM.

Model	HiddenNeurons	Dropout	LearningRate	MSE
LSTM	16	0.2500	0.0010	0.1358
RNN	16	0.0000	0.0010	0.2911
FFNN	32	0.0000	0.0100	1.1548
Regression	0	0.0000	0.0100	1.3348

Table 7.3: Selected parameter combinations in multivariate tuning step of price level prediction

7.1.4 Evaluation

Resulting from the rolling prediction approach selected for this step all models are trained on a larger amount of training data than in the tuning and variable selection steps. This number increases with each month chosen as test month. In table 7.4 the average value of the MSE across months is shown for each model. From these results one can draw several observations. Firstly the LSTM model outperforms all other neural networks as well as the regression approach both within the univariate as well as multivariate models. Secondly within each model category the univariate version outperforms its equivalent version containing additional inputs. However the most significant observation might be the fact that none of the models is able to outperform the lagged value approach representing the current market prices. Also there is a big gap between the worst two models represented by the multivariate feed forward neural network and regression models and the rest of the models.

Model	Variables	MSE
LaggedValue	TTFFM	0.1221
LSTM	TTFFM	0.1294
FFNN	TTFFM	0.1333
RNN	TTFFM	0.1578
LSTM	TTFFM OilFM	0.1706
RNN	TTFFM StorageUK ConNLDZNL	0.1746
Regression	TTFFM	0.2004
FFNN	TTFFM ProdUKCS ElectricityBaseFM StorageUK NBPFM	0.4918
Regression	TTFFM ElectricityBaseFM NBPFM StorageUK TTFDA	0.5022

Table 7.4: Average MSE across months for each model in evaluation step

Figures 7.1 and 7.2 give a more detailed impression of the performance of each model across test months, by showing the mean squared error on the test set for each test month selected in the evaluation process. From these graphs a few additional observations can be obtained. Regarding the relative performance of the different models, one can observe that the differences in performance generally diminish over time and that the overall differences are mainly driven by the earlier months. Also the mean squared error of the best models closely follow that of the lagged value reference model. In fact the remaining difference in performance between the univariate LSTM and the lagged value reference vanishes almost completely in the later months.

The consistent similarity in performance among the LSTM and the lagged value ref-

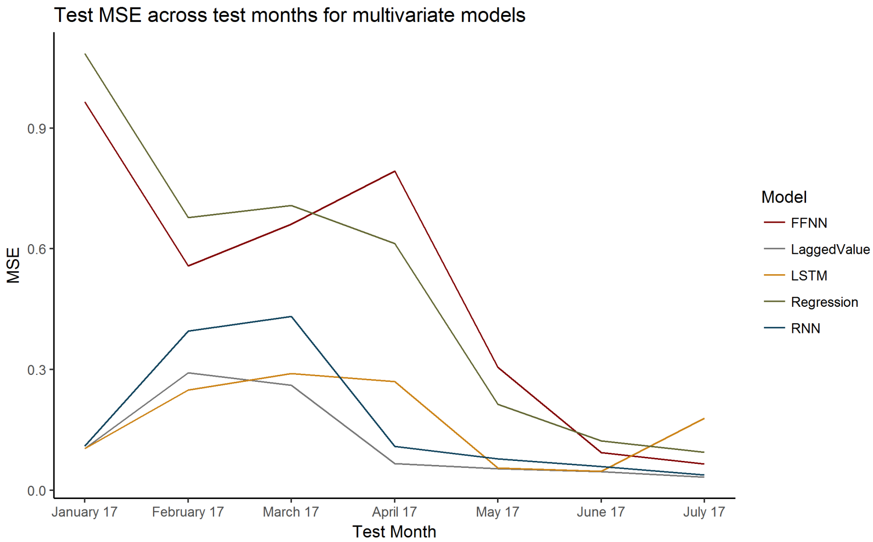

Figure 7.1: Month wise MSE for multivariate models

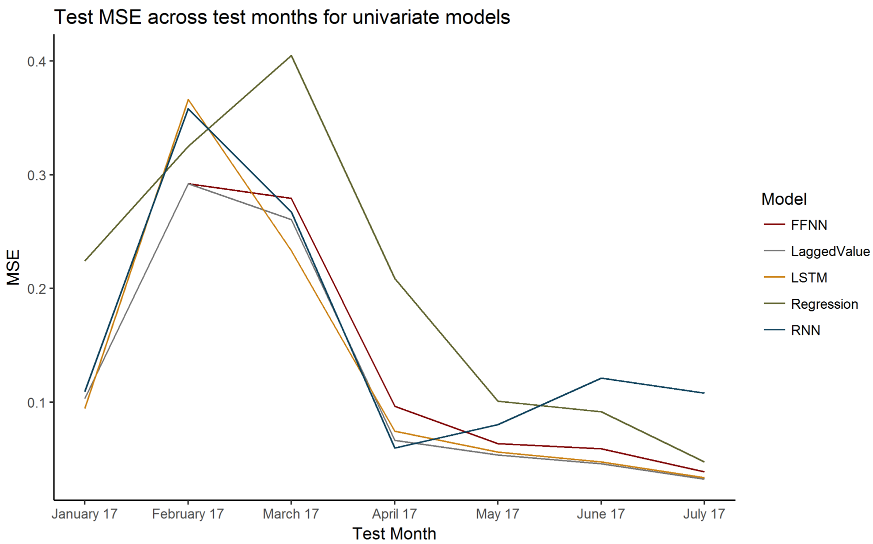

Figure 7.2: Month wise MSE for univariate models

erences suggest that they might provide very similar predictions. As figure 7.3 shows this in fact true. Graphically the predictions from both approaches are almost indistinguishable even after zooming in on the time frame starting in March 2017.

The mean absolute difference between both predictions is in fact only around 6.8 Euro cents or about 0.38 percent of the lagged value prediction over the whole test

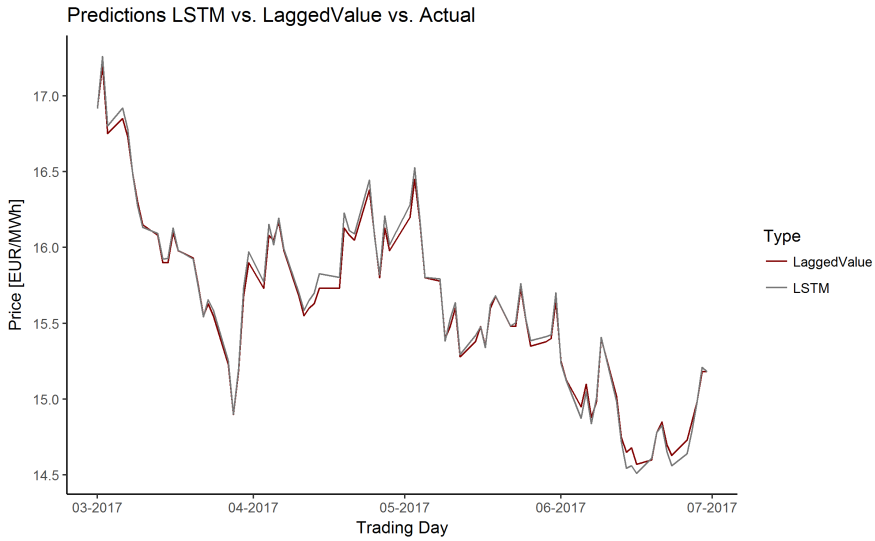

Figure 7.3: Predictions LSTM vs. Lagged Value

period. In fact this difference is highest in February, which is also the month with the worst performance of the LSTM relative to the Lagged Value approach.

7.2 Binary Prediction

The binary prediction was evaluated using the same approach, with the same amounts of data and tuning parameters as the price level prediction. To generate a monetary evaluation measure however a simple trading strategy was derived from the predictions of each model and the strategies were compared based on the average price paid for each MWh of natural gas.

7.2.1 Univariate Parameter Tuning

Regarding the learning rate, the univariate parameter tuning results are very similar to those of the price level prediction, with the recurrent models being tuned to significantly lower learning rates than the feed forward neural network and regression model. For the number of hidden layer neurons however both the LSTM and RNN models are tuned to different values.

Model	HiddenNeurons	Dropout	LearningRate	BCE
LSTM	8	0	0.0010	0.4676
RNN	16	0	0.0001	0.4595
FFNN	8	0	0.0100	0.4748
Regression	0	0	0.0100	0.4834

Table 7.5: Selected parameter combinations in univariate tuning step of binary prediction

7.2.2 Variable Selection

For the variable selection one can see a number of similarities to the results of the price level prediction problem. One similarity is the fact that for the LSTM model again only one additional variable is selected, which is the price of TTF gas on the day ahead market in this case. Another similarity is again the selection of the UK Storage levels, NBP front month prices and electricity futures as input variables to the other models. This reinforces the impression that these variables do in fact have some influence on the target variable.

Model	Variables	BCE
LSTM	TTFFM TTFDA	0.4516
RNN	TTFFM NBPFM StorageUK	0.4708
FFNN	TTFFM ConNLDZNL StorageUK	0.4741
Regression	TTFFM NBPFM ElectricityPeakFM EURGBPFX ElectricityBaseFM StorageUK	0.4878

Table 7.6: Selected variable combinations in variable selection step of binary prediction

7.2.3 Multivariate Parameter Tuning

Just as in the price level prediction case the results of the multivariate tuning process differ significantly from the values selected for the univariate models. A difference that might be noted however is that the values of the target function are much closer to those in the variable selection step. This might be seen as an indicator of more stable results.

Model	HiddenNeurons	Dropout	LearningRate	BCE
LSTM	32	0.2500	0.0010	0.4494
RNN	16	0.5000	0.0100	0.4721
FFNN	8	0.0000	0.0100	0.4734
Regression	0	0.0000	0.1000	0.4904

Table 7.7: Selected parameter combinations in multivariate tuning step of binary prediction

7.2.4 Evaluation

With respect to the relative ranking of the models according to the evaluation on the test set the results for the binary prediction differ substantially from those of the first prediction problem. One immediate observations is the very good performance of both the univariate as well as multivariate LSTM which make up the two best performing models according to binary cross entropy. Also unlike in the first prediction problem there is no clear preference for either univariate or multivariate models across model types. Furthermore the base line reference model, which in this case is based on the assumption of equal distribution does not dominate the other models in the way previously observed. In fact all of the models except for the multivariate FFNN and Regression outperform this reference model.

Model	Variables	BCE
LSTM	TTFFM	0.4519
LSTM	TTFFM TTFDA	0.4677
FFNN	TTFFM ConNLDZNL StorageUK	0.4776
RNN	TTFFM	0.4798
Regression	TTFFM	0.4812
FFNN	TTFFM	0.4815
EqualDistribution		0.4968
Regression	TTFFM NBPFM ElectricityPeakFM EURGBPFX ElectricityBaseFM StorageUK	0.5026
RNN	TTFFM NBPFM StorageUK	0.5171

Table 7.8: Average BCE across months for each model in evaluation step

When looking at the performance of the different models for each test month depicted in figures 7.4 and 7.5 the relative performances seem a lot more consistent over time than before. Apart from the equal distribution reference the relative ranking among the models rarely changes, with the LSTM model showing a lower error than other neural networks in almost all of the months. Also the size of the difference among the models does not decrease over time, as was observed above.

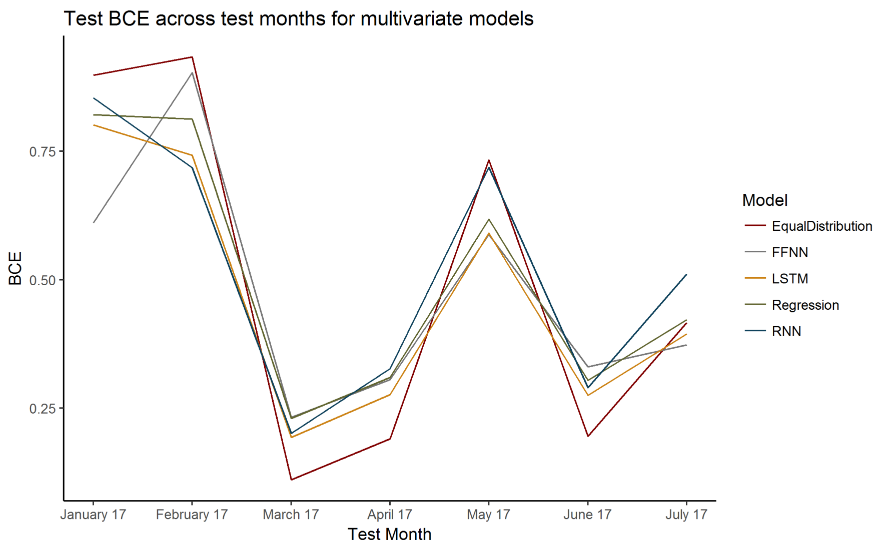

Figure 7.4: Month wise BCE for multivariate models

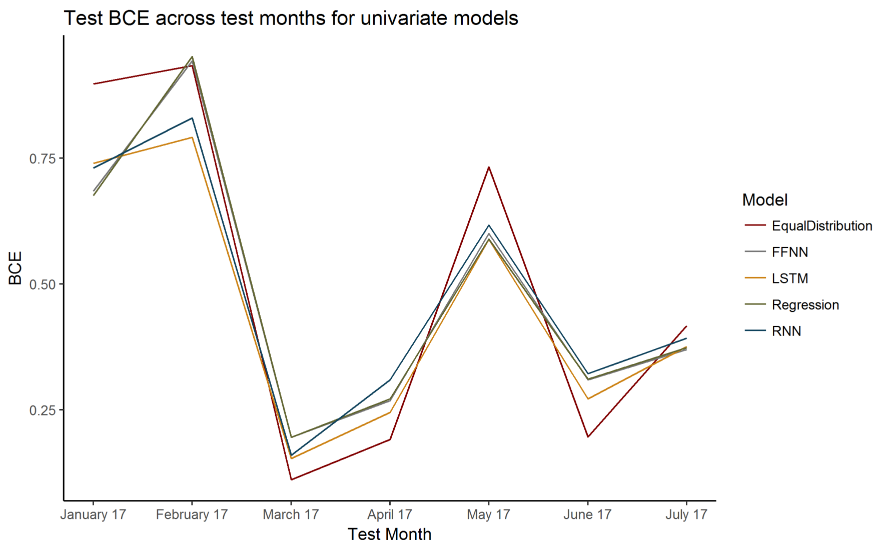

Figure 7.5: Month wise BCE for univariate models

7.2.5 Trading Strategy

As stated in the introduction, one of the reasons why this thesis was extended to include the binary prediction problem, was the aim to better support trading decisions of gas buyers in the chemical industry. One advantage of the predictions of the binary model in this respect is the fact that its predictions can be interpreted as the probability of the event that today's gas price is minimal among all remaining trading days. For any given trading day this event occurring would mean that the buyer should purchase all of its remaining projected demand today in order to minimise the price paid. Therefore it gives the decision maker a clearer recommendation than the price level prediction models. Another advantage of these models is the fact that one can very easily derive a trading strategy based on these predictions. One such trading strategy would be to use the prediction of the binary model as the percentage of the remaining demand to be bought on this trading day. Based on the predictions $\tilde{y}_{t,M}$ for trading day t in month M the traded share $\tilde{s}_{t,M}$ of the total demand is calculated as:

$$\forall t > t_0 : \tilde{s}_{t,M} = \left(\prod_{i=t_0}^{t-1}(1 - \tilde{y}_{i,M})\right)\tilde{y}_{t,M}$$

$$\tilde{s}_{t_0,M} = \tilde{y}_{t_0,M}$$

Besides its simplicity this strategy offers various advantages. One advantage is the flexibility to choose the parameter t_0 which represents the earliest trading day to begin buying natural gas. By setting the final prediction to one, it is ensured that the traded shares always add up to one and the whole demand is met. While this did require changing these predictions for models other than the equal distribution model it should be noted that these changes are rather small. This is due to the fact that these predictions were already very close to one since the binary target variable is by definition equal to one at the end of the month. Another advantage of this strategy is that applying it to the equal distribution reference model results in buying the same constant share for all trading days after t_0. This can be shown by plugging into the above formula the reference prediction $N_{t,M}$, which is the number of trading days in month M starting from trading day t:

$$\forall t > t_0 : \tilde{s}_{t,M} = (\prod_{i=t_0}^{t-1}(1 - \frac{1}{N_{i,M}}))\frac{1}{N_{t,M}}$$
$$= (\prod_{i=t_0}^{t-1}(\frac{N_{i,M} - 1}{N_{i,M}}))\frac{1}{N_{t,M}}$$

Obviously the number of remaining trading days decreases by one with each trading day and it holds $N_{i,M} - 1 = N_{i+1,M}$. Using this fact one gets:

$$\forall t > t_0 : \tilde{s}_{t,M} = (\prod_{i=t_0}^{t-1}(\frac{N_{i+1,M}}{N_{i,M}}))\frac{1}{N_{t,M}}$$
$$= \frac{N_{t,M}}{N_{t_0,M}}\frac{1}{N_{t,M}}$$
$$= \frac{1}{N_{t_0,M}}$$

This is a somewhat natural and well interpretable buying strategy to use as reference against which to evaluate the strategies resulting from the data driven models. The different trading strategies are compared against each other based on the average price that is paid per MWh, which is calculated for each month and model in the following way using the closing prices $P_{t,M}$:

$$\bar{P_M} = \sum_{t \geq t_0} \tilde{s}_{t,M} P_{t,M}$$

For the strategy resulting from the equal distribution this corresponds to the equally weighted price average across the trading period. In figures 7.6 and 7.7 the monthly values for this metric are plotted for univariate and multivariate models when choosing t_0 so as to limit trading to the last two weeks of every month.

Although the picture might be less clear from the graphical analysis, the results in table 7.9 show that in fact all data driven models result in cheaper prices than the equally weighted trading strategy. The prices reported in this table are calculated by taking the average of the prices paid according to each strategy across all months in the test period. The best performing models on this metric, which are the univariate RNN and LSTM models are able to save almost 6 Euro cents per MWh compared to the benchmark. At

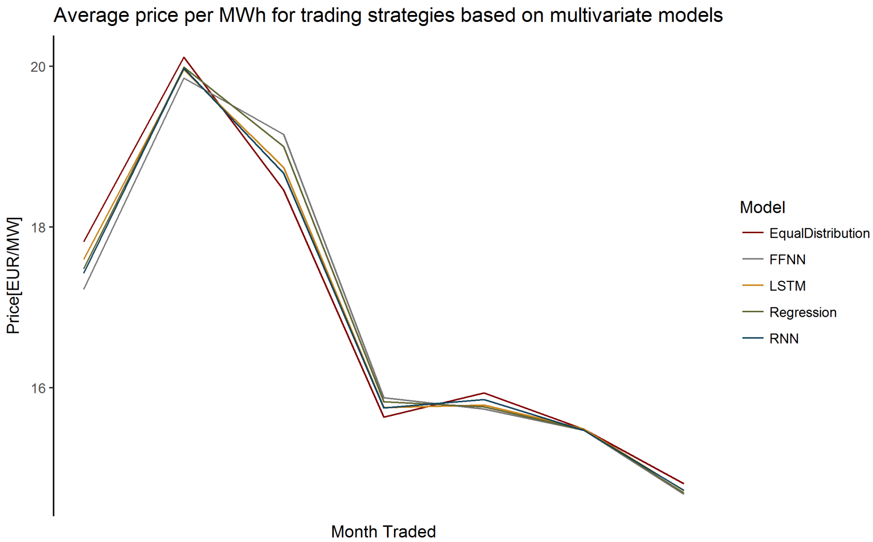

Figure 7.6: Average prices for multivariate models

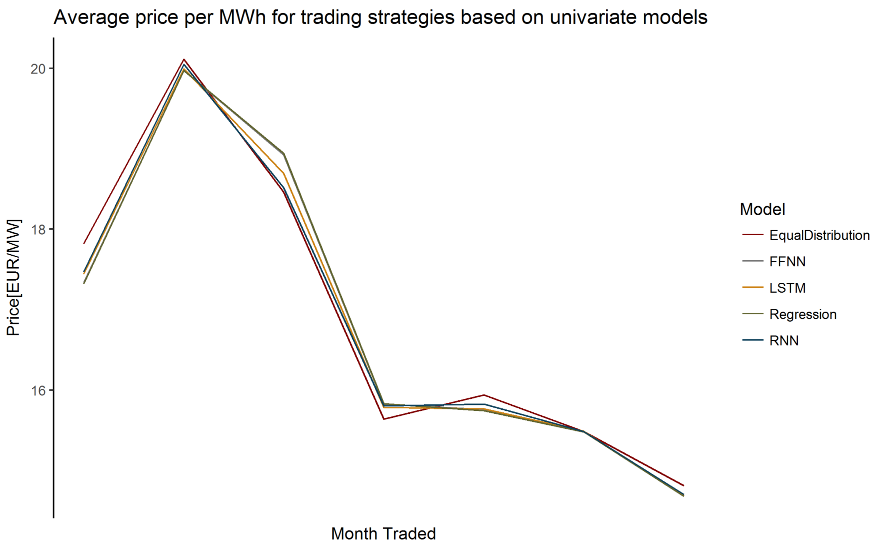

Figure 7.7: Average prices for univariate models

first sight this might be considered a very modest saving. However if one applies this saving to a gas user such as a power station with a constant gas demand of one Gigawatt these savings would amount to around half a million Euro per year.

As the trading strategy that results from each model depends on the choice of t_0 so do the average prices achieved by each model. However while the relative ranking

Model	Variables	AveragePrice
RNN	TTFFM	16.8342
LSTM	TTFFM	16.8358
RNN	TTFFM NBPFM StorageUK	16.8397
Regression	TTFFM	16.8513
FFNN	TTFFM	16.8538
FFNN	TTFFM ConNLDZNL StorageUK	16.8558
LSTM	TTFFM TTFDA	16.8649
Regression	TTFFM NBPFM ElectricityPeakFM EURGBPFX ElectricityBaseFM StorageUK	16.8905
EqualDistribution		16.8939

Table 7.9: Average price per MWh across months for simple trading strategy based on each model

among the data driven models does vary for different trading periods, the savings that these models achieve over the equal distribution reference are surprisingly stable. In table 8.2 the savings per MWh of the strategy resulting from the univariate LSTM are shown when choosing t_0 so as to limit trading to the last 1 - 15 trading days. Except for very short trading periods, where the strategies are almost identical, the LSTM achieves positive savings between 3 and 6.8 Euro cents.

Part VI

Conclusions

Chapter 8

Conclusions

In the following paragraphs the above illustrated results will be used to answer the research questions posed in the introduction. Afterwards a few additional observations and learnings obtained during the work on this thesis will be summarised and an outlook on future possible extensions of this work will be given.

8.1 Research Questions

8.1.1 Performance of recurrent neural networks

When assessing the performance of recurrent neural networks as a forecasting tool the results of the two prediction problems differ quite substantially. In the case of the price level prediction the simple lagged value approach using today's price as prediction for tomorrow outperforms both recurrent and feed forward neural networks. The best performing data driven models in this context are in fact those, whose predictions most closely resemble the lagged value. This lack of a predictive advantage might be explained by two different factors. One possible explanation is the assumption that the highly liquid natural gas market represents a very efficient pricing mechanism whose current price already takes all available information into account. When looking at the literature reviewed above regarding price forecasting of natural gas futures, one does in fact see that the results are quite mixed. Even many of the papers reporting positive performance of machine learning algorithms only compare their models to very similar approaches to evaluate the advantage of a particular feature such as wavelet decomposition or feature selection. Many of these papers do not include a comparison of the model against the price expectations of the market.

The alternative explanation might be a lack of data. Since the data is of only daily granularity and the model training was limited to complete observations, the models were only trained on around 1000 training observations. Especially recurrent neural networks and particularly LSTM models contain a very large number of trainable parameters. In this case the parameters sometimes outnumbered the available training observations by a factor of more than four to one. This makes efficient training of these models as well as avoiding overfitting particularly hard and might also be an explanation of the relatively bad performance of the multivariate models.

With respect to the binary prediction problem the evaluation yields a much more positive conclusion. On the one hand this might be due to the relatively weaker base line reference model which is based on an equal distribution of the probability of being minimal across the remaining days. Unlike the reference model in the price level prediction this forecast does not contain any information of the current market expectations. This reference is in fact outperformed by all but two of the eight different data driven models. However even among these models the performance of recurrent neural networks compares quite well, with three of the four top ranked models regarding cross entropy being of recurrent structure. Looking at the monetary evaluation of the derived trading strategies this picture is confirmed with the top three models all being recurrent.

8.1.2 Relative performance of LSTM

Comparing the relative performance of the LSTM model with that of the simpler RNN architecture the results are more consistent across the two prediction problems. When looking at the ranking of the models according to the loss function within the uni- and multivariate groups, the LSTM model outperforms the simple RNN in both prediction problems. The picture is slightly different in the monetary evaluation where the univariate simple RNN model slightly outperforms the LSTM, however it should be noted that the model was not optimized on this metric and the ranking slightly changes when choosing a different value for the t_0 parameter. Across all prediction problems and evaluation metrics the univariate LSTM model shows a consistently good performance being the best or second best model in all rankings.

8.1.3 Business value

The potential business value that the recurrent neural networks developed above can offer as a tool to support the purchasing of natural gas mostly derives from the results of the binary prediction problem. The good performance of the LSTM regarding binary cross entropy suggest that the predictions when interpreted as probability estimates already provide a useful piece of information for human decision making. However the most direct and tangible value is displayed by the good performance of the simple trading strategy derived from these predictions and the savings that can be realised compared to the equal distribution reference. While the performance of a human trader might be a more competitive reference, it should be taken into account that these models could run automatically and thereby realise further savings in personnel costs.

8.2 Other observations

Beyond the above findings some additional observations were made conducting the work presented in this thesis.

Input Variables: Across prediction problems and model type some input variables were repeatedly selected in the variable selection step. These input variables which seem to be of particular importance include the natural gas storage levels in the United Kingdom, the Dutch non-LDZ consumption, Electricity Futures and the front month future price for natural gas at the NBP hub.

Variable Scaling: During the first attempts at implementing the various models the data was not scaled. Instead both input and target variable were used in the format in which they were downloaded. For the price level prediction that meant using a linear output layer. Especially for this prediction problem but also for the binary prediction the variable scaling improved model performance more than expected.

8.3 Outlook

There are a number of possible ways in which the work in this thesis could be extended:

Further input variables: The candidates, that were included in the variable selection, were pre-selected based on expert input from a wider range of possibly relevant variables available on the *Thomson Reuters Eikon* database. This selection could be adjusted to include factors such as LNG markets or weather data.

Extended parameter tuning grid: Parameter tuning in this thesis was restricted to three tuning parameters and three to four candidate values for each parameter. This might be extended by either including more candidate values or adding additional tuning parameters such as the batch size or the number of training iterations.

Multi-layer architectures: All neural networks evaluated in this thesis were limited to architectures with at most one hidden layer. This might be changed to include multi-layer architectures which are present in some of the literature reviewed.

Multi-step forecasts: One advantage of recurrent neural networks are their presumptive strength in providing forecasts multiple steps into the future. Extending either of the presented prediction problems to multi-step prediction might unlock additional value in the models.

Wavelet transformation: One method especially popular in the literature on time series forecasting and often reported to bring significant improvements in accuracy is the combination of machine learning methods with wavelet decomposition. Applying this method to the price level prediction problem could also be a promising way to improve upon these results.

Monetary cost function: Based on feedback from industry experts the most popular aspect of the work presented above was the performance of the trading strategies based on the binary prediction problems. This success is somewhat surprising considering that the models where not trained to optimize this problem at all. This could be changed by implementing a cost function that represents the average price of this strategy and training the models to minimise this loss. It is reasonable to expect the performance of the strategy to improve thereby increasing the savings an industry user could realise.

Bibliography

ABRISHAMI, H. AND V. VARAHRAMI (2011): "Different methods for gas price forecasting," *Cuadernos de economía*, 34, 137–144.

ALIZADEH, A. AND K. MAFINEZHAD (2010): "Monthly Brent oil price forecasting using artificial neural networks and a crisis index," in *2010 International Conference on Electronics and Information Engineering*, vol. 2, V2–465–V2–468.

AMIN-NASERI, M. R. AND E. A. GHARACHEH (2007): "A hybrid artificial intelligence approach to monthly forecasting of crude oil price time series," in *The Proceedings of the 10th International Conference on Engineering Applications of Neural Networks, CEUR-WS284*, 160–167.

ANBAZHAGAN, S. AND N. KUMARAPPAN (2013): "Day-ahead deregulated electricity market price forecasting using recurrent neural network," *IEEE Systems Journal*, 7, 866–872.

BU, H. (2011): "Price dynamics and speculators in crude oil futures market," *Systems Engineering Procedia*, 2, 114–121.

BUCHANAN, W. K., P. HODGES, AND J. THEIS (2001): "Which way the natural gas price: an attempt to predict the direction of natural gas spot price movements using trader positions," *Energy Economics*, 23, 279–293.

BUSSE, S., P. HELMHOLZ, AND M. WEINMANN (2012): "Forecasting day ahead spot price movements of natural gas–An analysis of potential influence factors on basis of a NARX neural network," *Paper Presented by the Multikonferenz Wirtschaftsinformatik, Braunschweig, Germany*.

CAPORIN, M. AND F. FONTINI (2017): "The long-run oil–natural gas price relationship and the shale gas revolution," *Energy Economics*, 64, 511–519.

CHINN, M. D. AND O. COIBION (2014): "The predictive content of commodity futures," *Journal of Futures Markets*, 34, 607–636.

CONTRERAS, J., R. ESPINOLA, F. J. NOGALES, AND A. J. CONEJO (2003): "ARIMA models to predict next-day electricity prices," *IEEE transactions on power systems*, 18, 1014–1020.

CRESPO CUARESMA, J., J. HLOUSKOVA, S. KOSSMEIER, AND M. OBERSTEINER (2004): "Forecasting electricity spot-prices using linear univariate time-series models," *Applied Energy*, 77, 87–106.

CRUZ, A., A. MUÑOZ, J. L. ZAMORA, AND R. ESPÍNOLA (2011): "The effect of wind generation and weekday on Spanish electricity spot price forecasting," *Electric Power Systems Research*, 81, 1924–1935.

DÉES, S., A. GASTEUIL, R. KAUFMANN, AND M. MANN (2008): "Assessing the factors behind oil price changes," .

ČEPERIĆ, E., S. ŽIKOVIĆ, AND V. ČEPERIĆ (2017): "Short-term forecasting of natural gas prices using machine learning and feature selection algorithms," *Energy*, 140, 893–900.

GARETA, R., L. M. ROMEO, AND A. GIL (2006): "Forecasting of electricity prices with neural networks," *Energy Conversion and Management*, 47, 1770–1778.

GENG, J.-B., Q. JI, AND Y. FAN (2016): "How regional natural gas markets have reacted to oil price shocks before and since the shale gas revolution: A multi-scale perspective," *Journal of Natural Gas Science and Engineering*, 36, 734–746.

HAIDAR, I., S. KULKARNI, AND H. PAN (2008): "Forecasting model for crude oil prices based on artificial neural networks," in *Intelligent Sensors, Sensor Networks and Information Processing, 2008. ISSNIP 2008. International Conference on*, IEEE, 103–108.

HARTLEY, P. R., K. B. MEDLOCK III, AND J. E. ROSTHAL (2008): "The relationship of natural gas to oil prices," *The Energy Journal*, 47–65.

HE, K., L. YU, AND K. K. LAI (2012): "Crude oil price analysis and forecasting using wavelet decomposed ensemble model," *Energy*, 46, 564–574.

HEATHER, P. (2015): *The evolution of European traded gas hubs*, Oxford Institute for Energy Studies.

HERBERT, J. H. (1993): "The relation of monthly spot to futures prices for natural gas," *Energy*, 18, 1119–1124.

IEA, I. (2011): "World energy outlook 2011," *Int. Energy Agency*, 666.

——— (2017): "World energy outlook 2017," *Int. Energy Agency*, 666.

JIN, J. AND J. KIM (2015): "Forecasting Natural Gas Prices Using Wavelets, Time Series, and Artificial Neural Networks," *PLOS ONE*, 10, e0142064.

KIM, C.-I., I.-K. YU, AND Y. H. SONG (2002): "Prediction of system marginal price of electricity using wavelet transform analysis," *Energy Conversion and Management*, 43, 1839–1851.

KNITTEL, C. R. AND M. R. ROBERTS (2005): "An empirical examination of restructured electricity prices," *Energy Economics*, 27, 791–817.

KRISTIANSEN, T. (2012): "Forecasting Nord Pool day-ahead prices with an autoregressive model," *Energy Policy*, 49, 328–332.

LIRA, F., C. MUNOZ, F. NUNEZ, AND A. CIPRIANO (2009): "Short-term forecasting of electricity prices in the Colombian electricity market," *IET generation, transmission & distribution*, 3, 980–986.

MALLIARIS, M. E. AND S. G. MALLIARIS (2008): "Forecasting inter-related energy product prices," *The European Journal of Finance*, 14, 453–468.

MINGMING, T. AND Z. JINLIANG (2012): "A multiple adaptive wavelet recurrent neural network model to analyze crude oil prices," *Journal of Economics and Business*, 64, 275–286.

MIRIKITANI, D. AND N. NIKOLAEV (2011): "Nonlinear maximum likelihood estimation of electricity spot prices using recurrent neural networks," *Neural Computing and Applications*, 20, 79–89.

MISHRA, V. AND R. SMYTH (2016): "Are natural gas spot and futures prices predictable?" *Economic Modelling*, 54, 178–186.

MISIOREK, A., S. TRUECK, AND R. WERON (2006): "Point and interval forecasting of spot electricity prices: Linear vs. non-linear time series models," *Studies in Nonlinear Dynamics & Econometrics*, 10.

MOSHIRI, S. AND F. FOROUTAN (2006): "Forecasting nonlinear crude oil futures prices," *The Energy Journal*, 81–95.

MU, X. (2007): "Weather, storage, and natural gas price dynamics: Fundamentals and volatility," *Energy Economics*, 29, 46–63.

NGUYEN, H. T. AND I. T. NABNEY (2008): "Combining the wavelet transform and forecasting models to predict gas forward prices," in *Machine Learning and Applications, 2008. ICMLA'08. Seventh International Conference on*, IEEE, 311–317.

——— (2010): "Short-term electricity demand and gas price forecasts using wavelet transforms and adaptive models," *Energy*, 35, 3674–3685.

NICK, S. AND S. THOENES (2014): "What drives natural gas prices?—A structural VAR approach," *Energy Economics*, 45, 517–527.

NOGALES, F. J. AND A. J. CONEJO (2006): "Electricity price forecasting through transfer function models," *Journal of the Operational Research Society*, 57, 350–356.

NOGALES, F. J., J. CONTRERAS, A. J. CONEJO, AND R. ESPÍNOLA (2002): "Forecasting next-day electricity prices by time series models," *IEEE Transactions on power systems*, 17, 342–348.

PANAPAKIDIS, I. P. AND A. S. DAGOUMAS (2016): "Day-ahead electricity price forecasting via the application of artificial neural network based models," *Applied Energy*, 172, 132–151.

PANELLA, M., F. BARCELLONA, AND R. L. D'ECCLESIA (2012): "Forecasting energy commodity prices using neural networks," *Advances in Decision Sciences*, 2012.

PANG, Y., W. XU, L. YU, J. MA, K. K. LAI, S. WANG, AND S. XU (2011): "Forecasting the crude oil spot price by wavelet neural networks using OECD petroleum inventory levels," *New Mathematics and Natural Computation*, 7, 281–297.

SALEHNIA, N., M. A. FALAHI, A. SEIFI, AND M. H. M. ADELI (2013): "Forecasting natural gas spot prices with nonlinear modeling using Gamma test analysis," *Journal of Natural Gas Science and Engineering*, 14, 238–249.

SCHMUTZ, A. AND P. ELKUCH (2004): "Electricity price forecasting: Application and experience in the European power markets," in *Proceedings of the 6th IAEE European Conference, Zürich*.

SHAFIE-KHAH, M., M. P. MOGHADDAM, AND M. K. SHEIKH-EL-ESLAMI (2011): "Price forecasting of day-ahead electricity markets using a hybrid forecast method," *Energy Conversion and Management*, 52, 2165–2169.

SHAMBORA, W. E. AND R. ROSSITER (2007): "Are there exploitable inefficiencies in the futures market for oil?" *Energy Economics*, 29, 18–27.

SHARMA, V. AND D. SRINIVASAN (2013): "A hybrid intelligent model based on recurrent neural networks and excitable dynamics for price prediction in deregulated electricity market," *Engineering Applications of Artificial Intelligence*, 26, 1562–1574.

VILLAR, J. A. AND F. L. JOUTZ (2006): "The relationship between crude oil and natural gas prices," *Energy Information Administration, Office of Oil and Gas*, 1–43.

WANG, J. AND J. WANG (2016): "Forecasting energy market indices with recurrent neural networks: Case study of crude oil price fluctuations," *Energy*, 102, 365–374.

WERON, R. (2014): "Electricity price forecasting: A review of the state-of-the-art with a look into the future," *International journal of forecasting*, 30, 1030–1081.

WERON, R. AND A. MISIOREK (2008): "Forecasting spot electricity prices: A comparison of parametric and semiparametric time series models," *International journal of forecasting*, 24, 744–763.

WERON, R., A. MISIOREK, AND OTHERS (2005): "Forecasting spot electricity prices with time series models," in *Proceedings of the European Electricity Market EEM-05 Conference*, 133–141.

WOO, C.-K., A. OLSON, AND I. HOROWITZ (2006): "Market efficiency, cross hedging and price forecasts: California's natural-gas markets," *Energy*, 31, 1290–1304.

YAMIN, H. Y., S. M. SHAHIDEHPOUR, AND Z. LI (2004): "Adaptive short-term electricity price forecasting using artificial neural networks in the restructured power markets," *International journal of electrical power & energy systems*, 26, 571–581.

YE, M., J. ZYREN, AND J. SHORE (2002): "Forecasting crude oil spot price using OECD petroleum inventory levels," *International Advances in Economic Research*, 8, 324–333.

——— (2005): "A monthly crude oil spot price forecasting model using relative inventories," *International Journal of Forecasting*, 21, 491–501.

——— (2006): "Forecasting short-run crude oil price using high-and low-inventory variables," *Energy Policy*, 34, 2736–2743.

ZAREIPOUR, H., C. A. CAÑIZARES, K. BHATTACHARYA, AND J. THOMSON (2006): "Application of public-domain market information to forecast Ontario's wholesale electricity prices," *IEEE Transactions on Power Systems*, 21, 1707–1717.

Part VII

Appendix

8.4 Figures

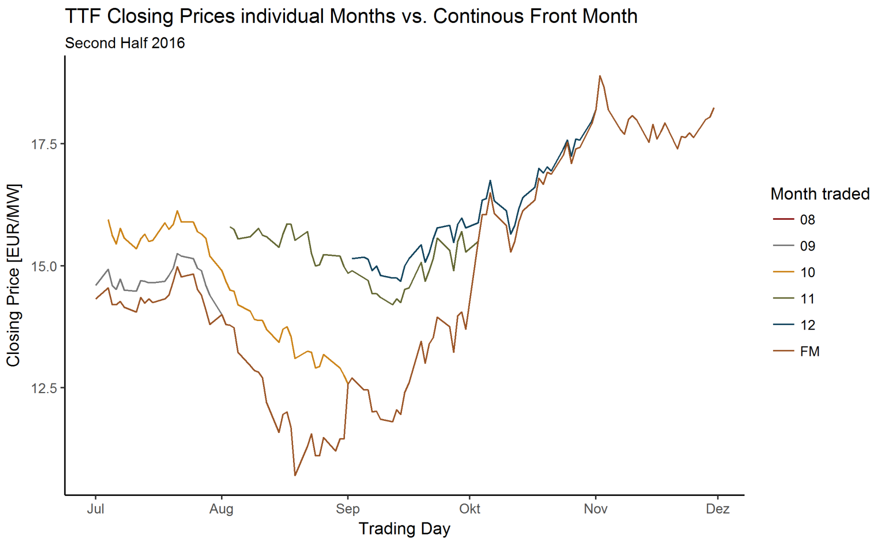

Figure 8.1: TTF Front Month vs. Individual Months

Figure 8.2: TTF Front Month Closing Prices

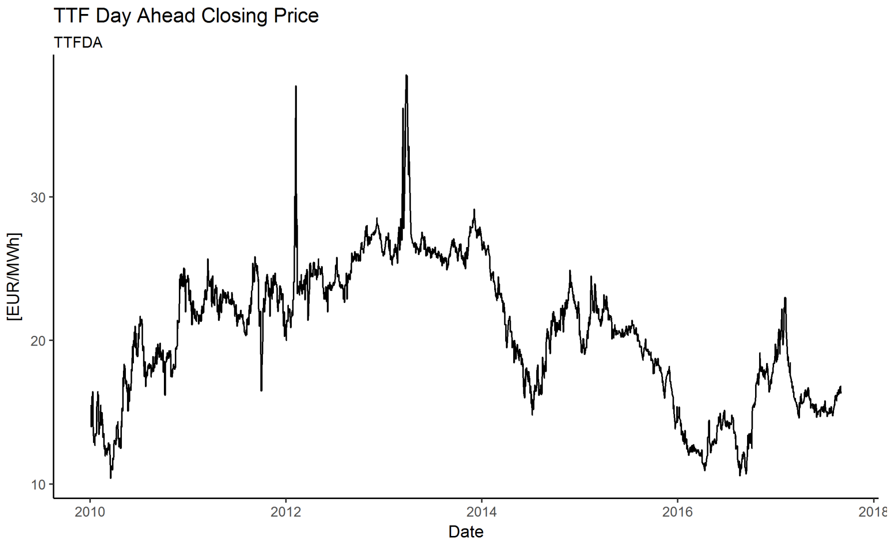

Figure 8.3: TTF Day Ahead

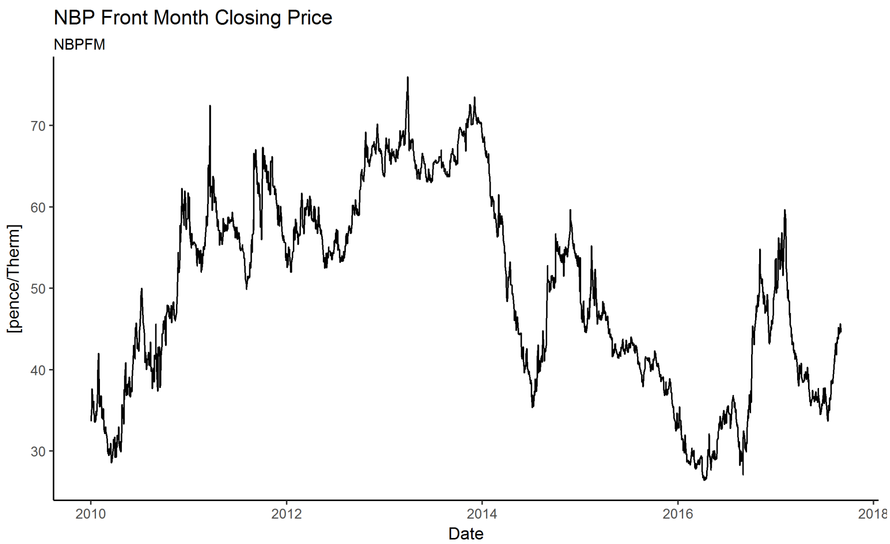

Figure 8.4: NBP Front Month

Figure 8.5: Brent Oil Front Month

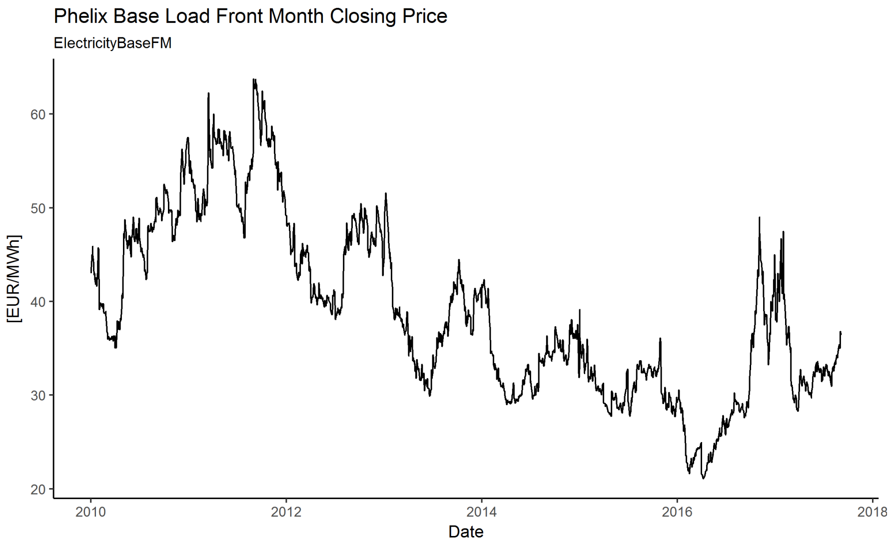

Figure 8.6: Phelix Base Load Front Month

Figure 8.7: Phelix Peak Load Front Month

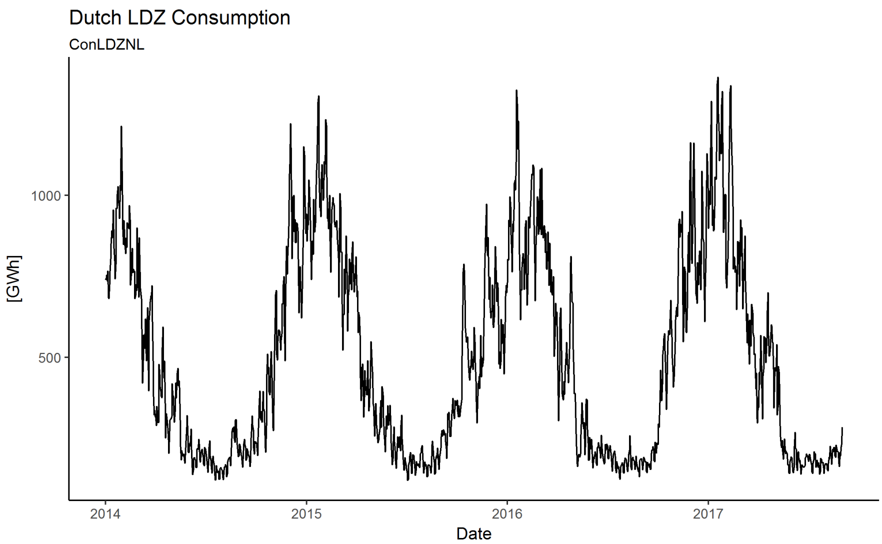

Figure 8.8: Dutch LDZ Consumption

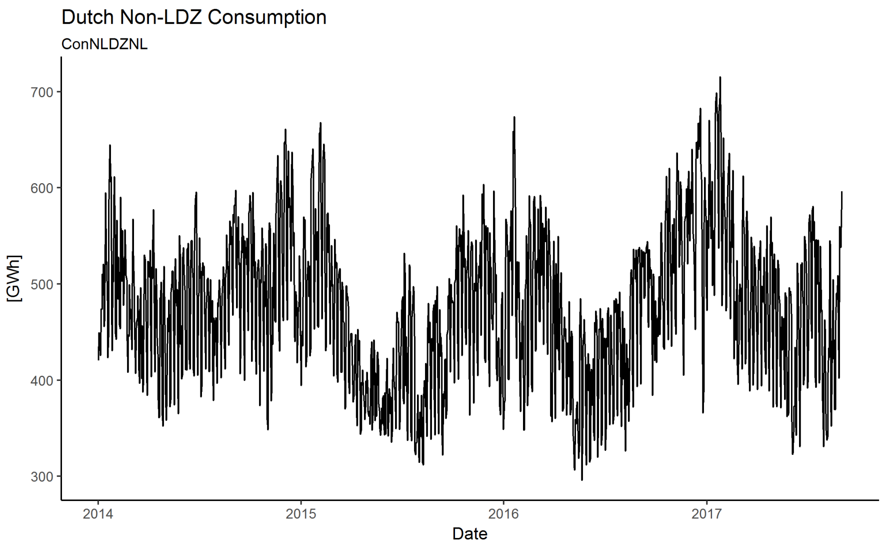

Figure 8.9: Dutch Non-LDZ Consumption

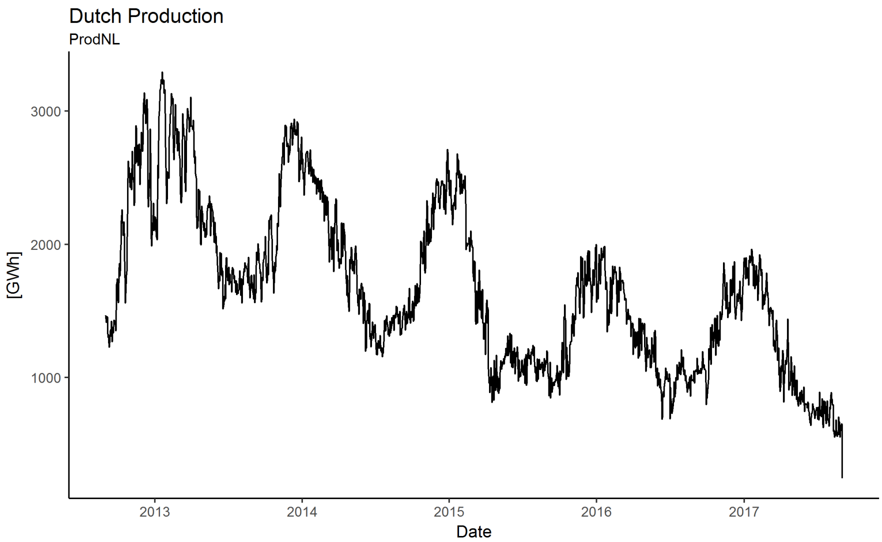

Figure 8.10: Dutch Production

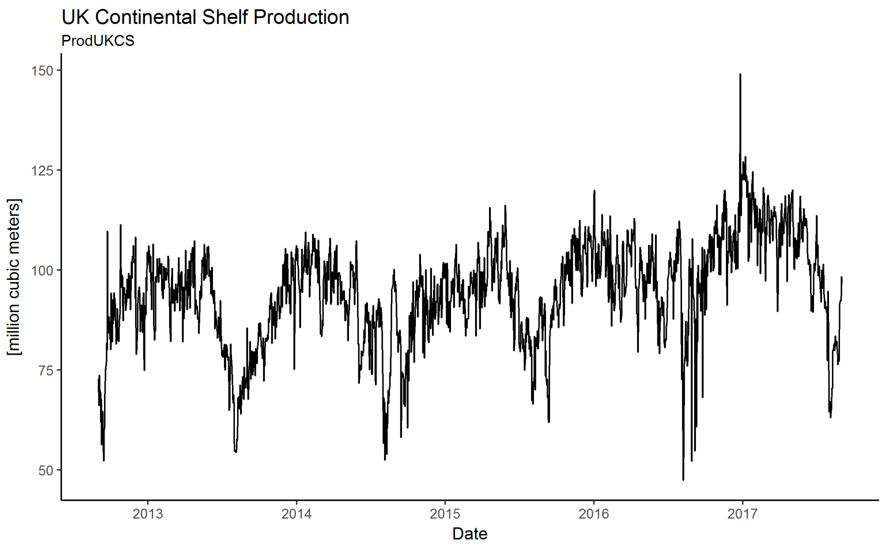

Figure 8.11: UK Continental Shelf Production

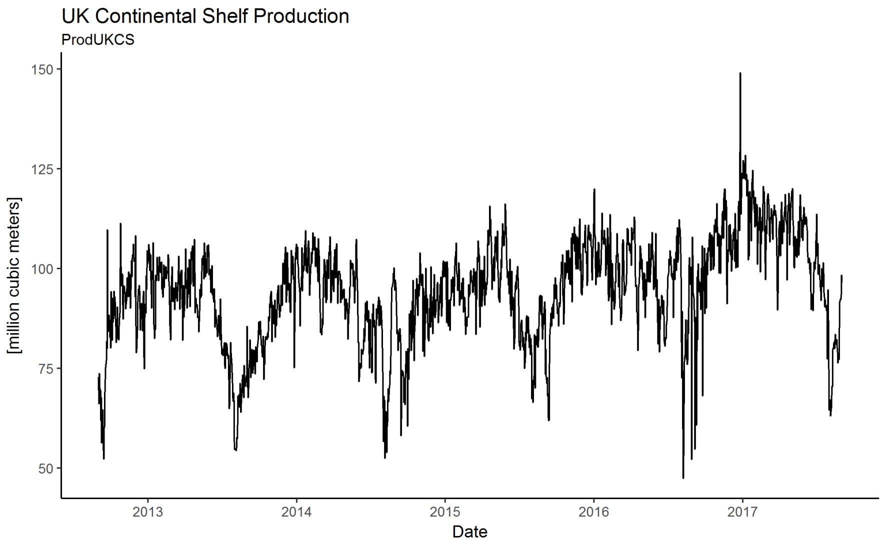

Figure 8.12: UK Continental Shelf Production

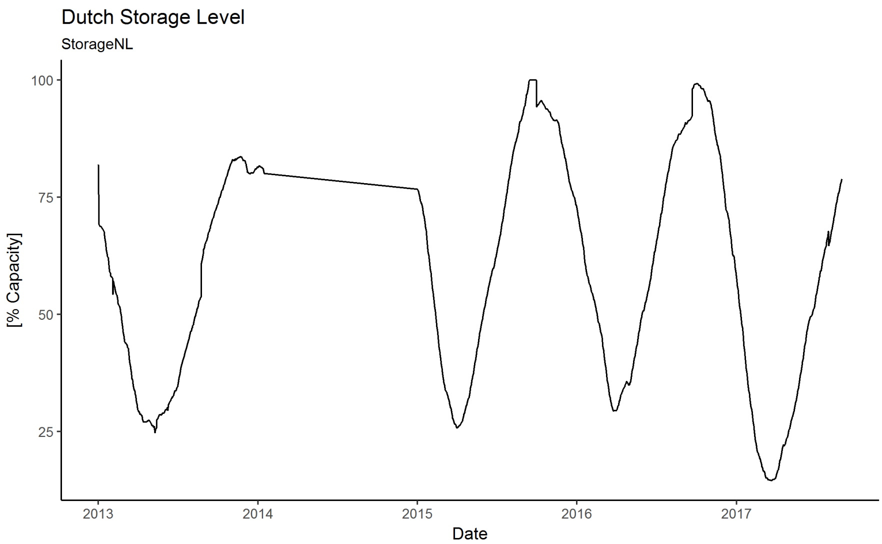

Figure 8.13: Dutch Storage

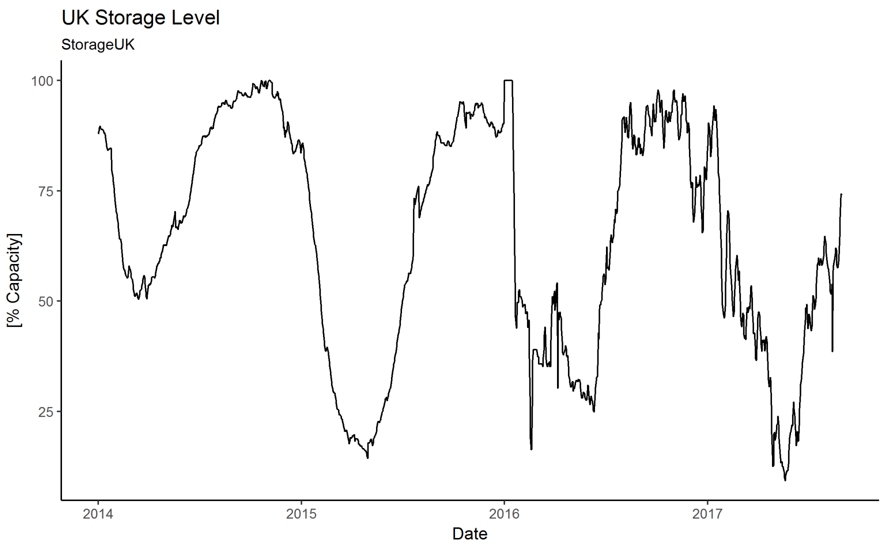

Figure 8.14: UK Storage

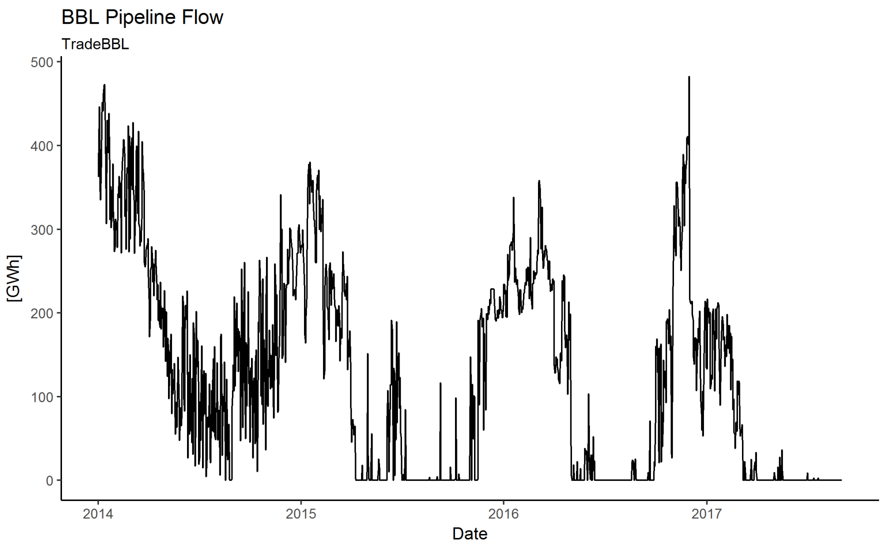

Figure 8.15: BBL Pipeline Flow

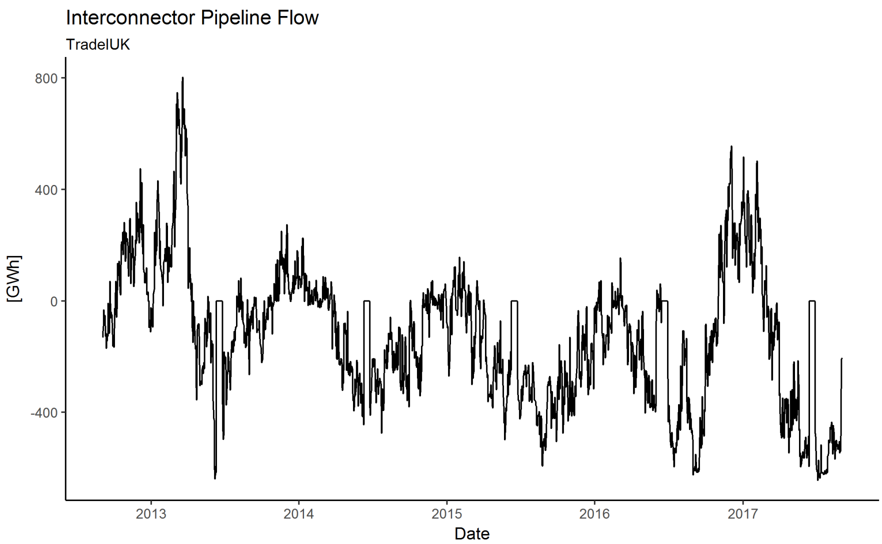

Figure 8.16: Interconnector Pipeline Flow

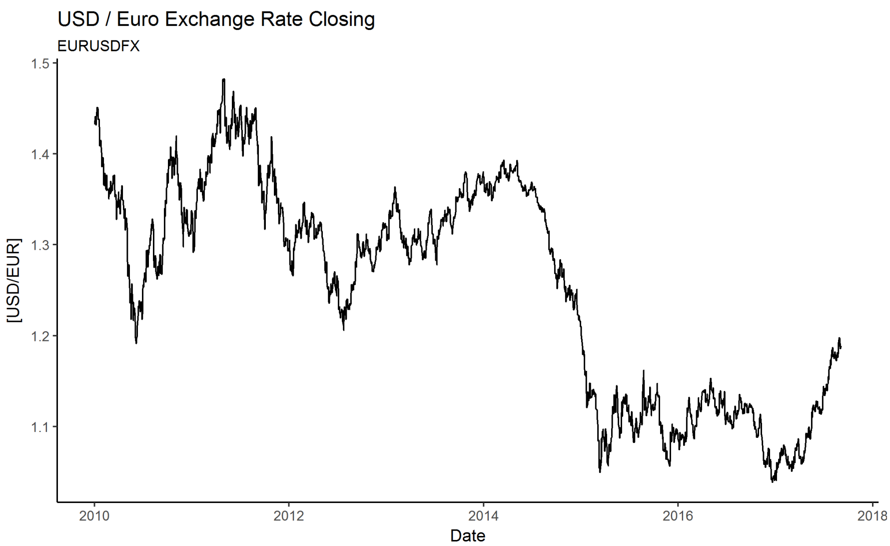

Figure 8.17: USD - EUR Exchange Rate

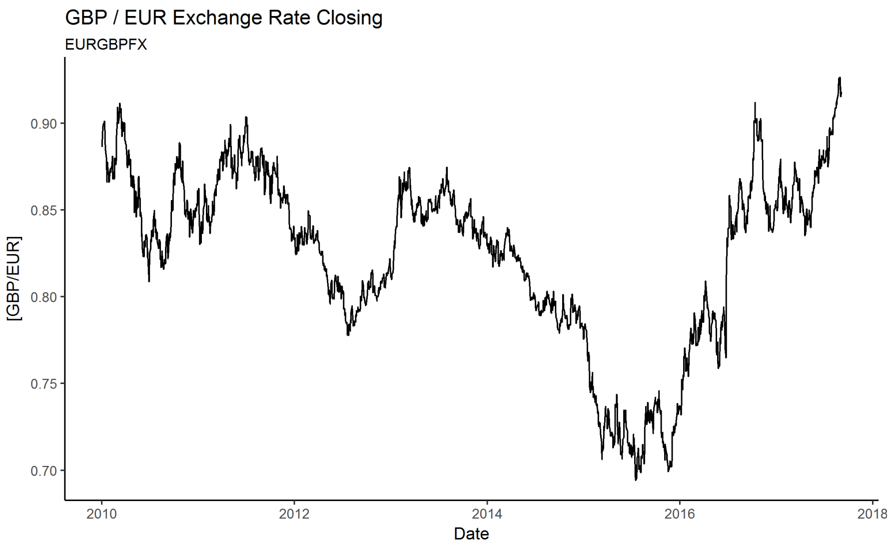

Figure 8.18: GBP - EUR Exchange Rate

8.5 Tables

Year	PositiveShare
2010	0.345
2011	0.228
2012	0.236
2013	0.220
2014	0.135
2015	0.194
2016	0.326
2017	0.291
Total	0.245

Table 8.1: Annual share of positive observation for binary target variable

TradingPeriod	LSTMSavings
1	0.0000
2	-0.0078
3	-0.0053
4	0.0309
5	0.0554
6	0.0392
7	0.0297
8	0.0299
9	0.0337
10	0.0401
11	0.0491
12	0.0679
13	0.0609
14	0.0580
15	0.0529

Table 8.2: Savings in EUR/MWh of univariate LSTM model relative to equal distribution benchmark for differenttrading periods