Natural gas price forecasting using recurrent neural networks

Research Proposal for Master's Thesis

to be submitted to

Prof. Dr. Stefan Lessmann

Humboldt-Universität zu Berlin
School of Business and Economics
Chair of Information Systems

by

Christian Koopmann
(572485)

Acknowledgement

I would like to thank

Abstract

This is the template for a thesis at the Chair of Econometrics of Humboldt–Universität zu Berlin. A popular approach to write a thesis or a paper is the IMRAD method (Introduction, Methods, Results and Discussion). This approach is not mandatory! You can find more information about formal requirements in the booklet 'Hinweise zur Gestaltung der äußeren Form von Diplomarbeiten' which is available in the office of studies.

The abstract should not be longer than a paragraph of around 10 to 15 lines (or about 150 words). The abstract should contain a concise description of the econometric/economic problem you analyse and of your results. This allows the busy reader to obtain quickly a clear idea of the thesis content.

Contents

List of Abbreviations

List of Figures

List of Tables

1 Introduction

1.1 Topic

1.2 Motivation

1.3 Structure

2 European Gas Market

In this section I will give a very brief overview of the history and current situation in the European natural gas market. A 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 and then spreading to the United States and Continental Europe in 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 building 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 has 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 just as 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. 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.

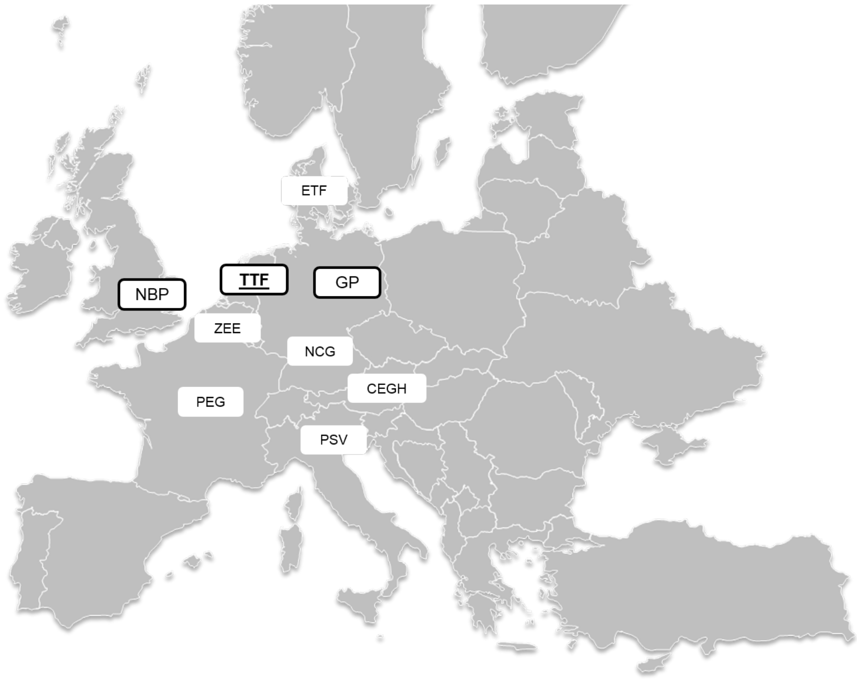

Figure 1: Selected European Gas Hubs. Primary Focus: TTF, Secondary Focus: NBP, GP

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.

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.

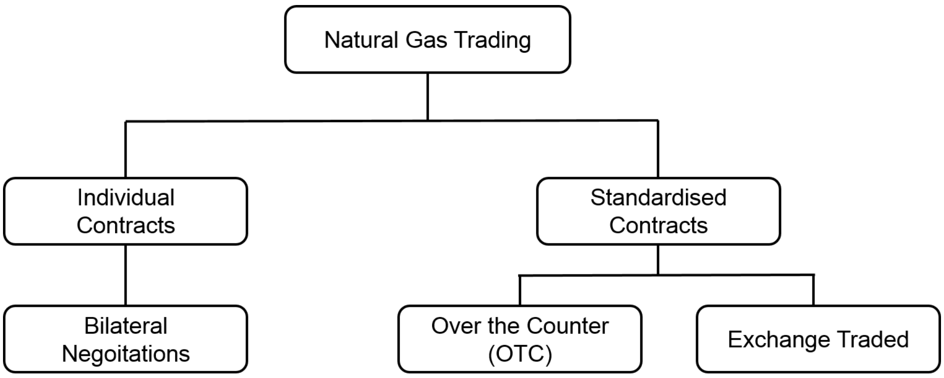

Figure 2: Types of Natural Gas Trading

2.3 Natural Gas Derivatives

The natural gas contracts to be analysed in the course of 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, -Quarter etc. 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 1: Future vs. Spot Markets

3 Literature Overview

In the following I 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. As explained above, natural gas shares characteristics with both of the other categories. The literature is categorised into three ares: Linear Models (LR), Feed Forward Neural Networks (FFNN) and Recurrent Neural Networks. 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 models 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 prediction of future prices.

3.1 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, 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 my modelling approach. 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 Fritz-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 2: Electricity Price Forecasting Literature

3.2 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 Des et al. (2008) 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 by Bu (2011) where the author uses reported trading positions of speculative traders as regressors and a GARCH model for the residuals. Moshiri and Foroutan (2006) forecasts the same target variable 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. 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 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
Des 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 3: Oil Price Forecasting Literature

3.3 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 is focused more on explaining certain market characteristics rather than explicit price forecasting. Special attention of 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 commodity markets, is the connection between future and spot prices of natural gas (Herbert (1993), Chinn and Coibion (2014)).

Linear Models The work done inMishra 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 and its relation to the spot 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 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. 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
Salehnia et al. (2013)	HH Spot	FFNN
Busse et al. (2012)	NCG Spot	RNN

Table 4: Gas Price Forecasting Literature

4 Method

In the following section I will give a brief overview over the theory behind Recurrent Neural Networks in General and Long Short-Term Memory (LSTM) network architecture in particular

4.1 RNN

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. Here we separated the structure of the model at each time step 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. The simplest specification is to just use the output h_t of a single non-linearity f as input for both. This architecture is called Simple Recurrent Neural Network or Elman Network. In Figure 4 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

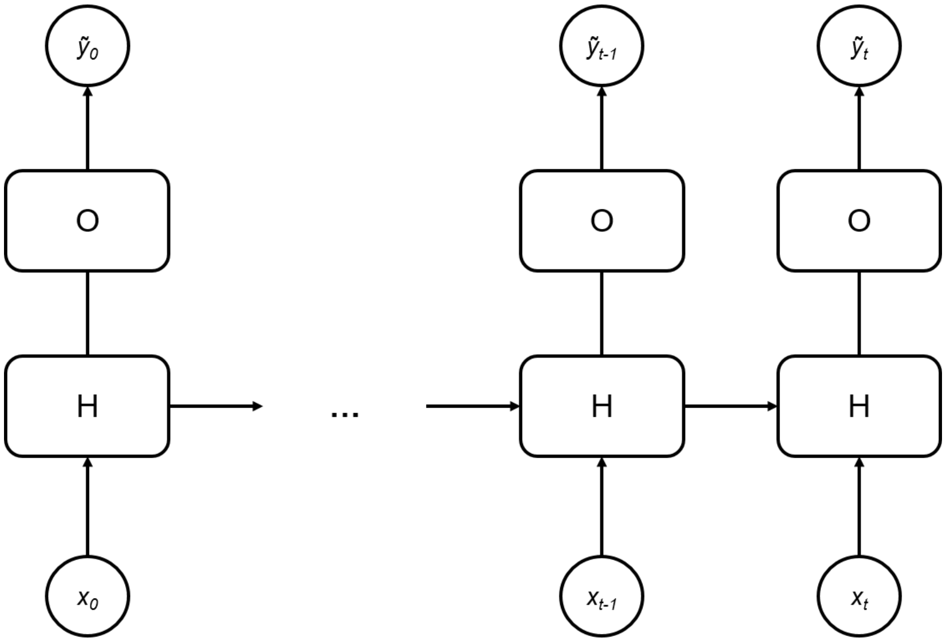

Figure 3: Simplified RNN Architecture

and d_x the number of hidden layer outputs and input variables respectively than the total number of trainable parameters is: $n_{par} = d_h * (d_h + d_x)$.

4.2 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 I will assume an additive loss function. Therefore the analysis will be limited to the derivative of the loss function for one observation $l_t = l(y_t, \tilde{y}_t)$. From the previously defined general structure of an RNN network we now that the prediction $\tilde{y}_t$ is a function (represented by the Output Layer in figure 3) 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:

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

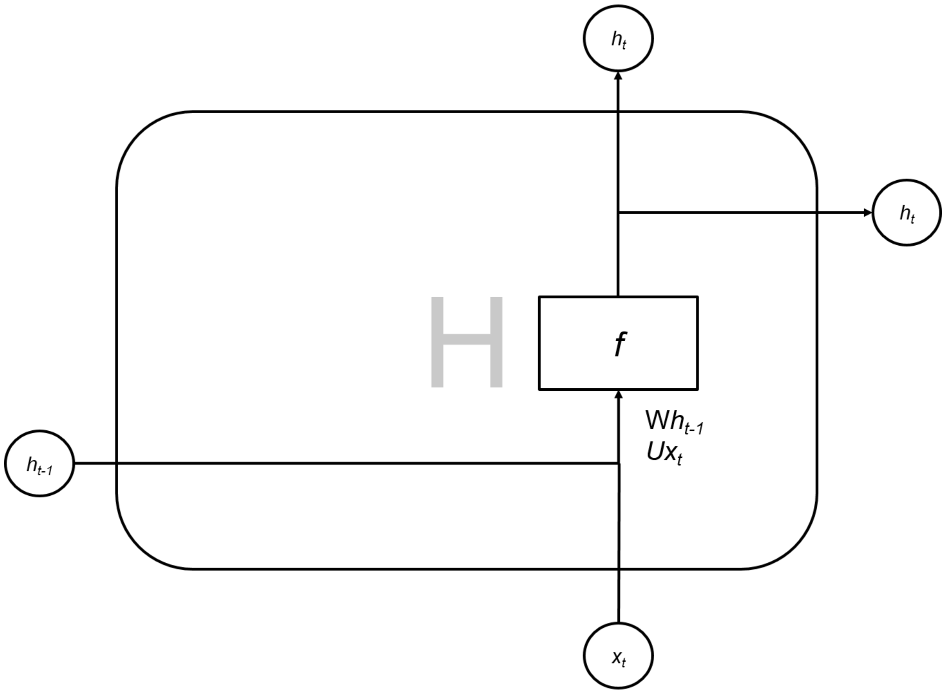

Figure 4: Hidden Layer of Elman Network

The first two terms on the right hand side are independent of the structure of the recurrent / hidden layer and in the following I 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 we get 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 first 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 a very similar structure and learning these parameters would therefore suffer from the same problems.

4.3 LSTM

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 I visualised in Figure 5. 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. 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.

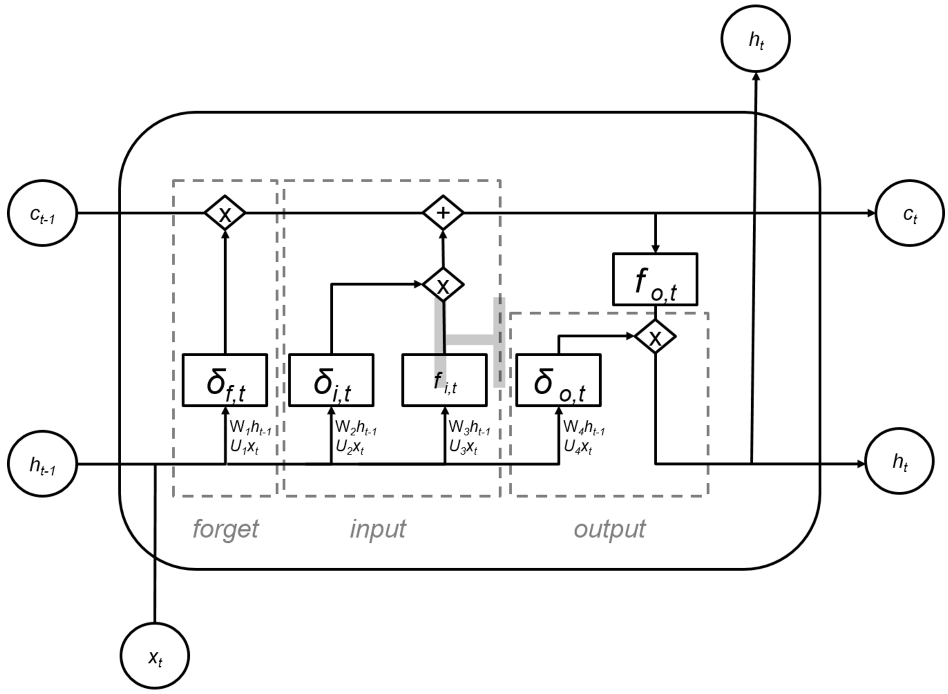

Figure 5: Hidden Layer of LSTM Network

5 Experimental Design

Designing an experiment to evaluate the performance of the LSTM model in comparison to reference models one has to solve several problems:

1. Reference Models
   - Selecting reference models
2. Parameter Tuning
   - Defining Parameter Tuning process
   - Selecting hyper parameters to tune and choosing tuning grid
3. Variable Selection
   - Defining variable selection process
   - Selecting candidate variables
4. Model Evaluation
   - Selecting hold out data to exclude from parameter tuning and variable selection.
   - Defining Testing Scenario

A major challenge arising from this problem is the interdependence between the parameter tuning and the variable selection steps. To tune the parameters we have 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 *purely autoregressive* 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 autoregressive 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 *purely autoregressive* in this context is a model trained using only past observations of the target variable as input. In the following paragraphs the selected approach for each of the above mentioned decision problems is outlined.

5.1 Reference Models

The prediction results 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. Regarding the input variables I use both a *short* and *long* version of the FFNN. Whereas the *short FFNN* only takes in the last observation of each input variable the *long FFNN* takes as input a larger number of values based on the *length* hyper-parameter (see below).

Regression Models: In the case of the price level prediction this reference model consists of a standard OLS estimation. In the autoregressive case it is an AR(1) model whereas in the multivariate case it is a linear regression on the last available value for each variable. For the binary prediction problem a logistic regression on the same set of variables is applied using maximum likelihood estimation.

5.2 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.

5.2.1 Fixed Hyper-Parameters

In the following I 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 the gradient as well as the layer activations can flow infinitely far through the network. However 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 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. 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* is used as loss for the price level prediction whereas *binary cross entropy* is used for the binary prediction problem.

Optimizer: The *optimizer* 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.

5.2.2 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. 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 For recurrent networks this is applied to both external inputs as well as past hidden layer outputs. Although the dropout values for both types could be set independently they are always set to the same level in the tuning process to limit the dimensionality of the tuning grid. Dropout is only used in multivariate models.

Architecture: The architecture tuning 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.

5.2.3 Tuning Process

6 Evaluation

The following section will describe both the experimental design used to evaluate the performance of the LSTM model and the results of that evaluation. The experimental design contains explanations on the chosen reference methods, the error measure, variable selection, the data preprocessing and the testing scenario.

6.1 Experimental Design

Overall the evaluation of LSTM models is separated in two parts. In the first part the LSTM is used as a purely autoregressive model. That means that the only input variables used for the forecasting are past values of the target variable. In the second part we extend the model to the multivariate case where past values of some of the other variables described above are included as inputs.

6.1.1 Reference Methods

The aim of this work is to evaluate the relative merit of the LSTM method for time series predictions. Therefore any reference model should use the same set of input variables as the LSTM network so that differences in accuracy can be attributed solely to the forecasting method and not difference in predictive values of the inputs. For both evaluation scenarios we use a linear model as well as a feed forward neural network as reference models. In the autoregressive case the linear model would be an AR-model whose order would have to be specified. Whoever both the analysis of (Partial) Auto Correlations in the explanatory analysis as well as the experimental application of various order selection methods strongly suggest a model of order one. Fitting these kind of models to various subsets of the data all produced estimates of the only regression coefficient very close to one. Instead of refitting the AR model on every new train/test split the coefficient is instead held constant at a value of one. This is equivalent to just using the current value of the target variable as a prediction for the next value. Apart from the reduction in complexity this also produces a reference model that can be seen as a representation of the hypothesis of an efficient market where all information about future prices is incorporated in the current price. In the multivariate case the linear reference models consist of both this lagged value as well as a linear regression on the same variables that are used by the LSTM. The non-linear reference model for comparison with the LSTM is a feed forward neural network with the same number of layers as the LSTM. In the univariate case this model only takes the last value as an input where as in the multivariate case it again uses the same variables as the LSTM.

6.1.2 Error Measures

The following error measures are used to compare the predictive quality of different models:

Mean Squared Error

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

Mean Absolute Error

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \tilde{y_i}|$$

Mean Absolute Percentage Error

$$MAPE = 100 * \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \tilde{y_i}|}{|y_i|}$$

6.1.3 Preprocessing

Before applying the different modelling techniques two preprocessing steps are applied to the data. In the first step missing values of both the target as well as the input variables are replaced using a forward filling procedure. This means that for each missing value the last available value from that time series is filled in. Only in the case that the missing value occurs at the beginning of the respective time series the whole observation is deleted. This procedure preserves the overall temporal structure of the data ensuring that the previous value in the data in fact represents the data point one time step (trading day) in the past. Additionally this forward filling procedure is in line with the observation that especially for market prices a missing value usually means that the asset was not traded on that day. Therefore it is somewhat natural to continue using the closing price of the previous trading day as the current price. For training and testing the neural network models both input and target variables have additionally been scaled using a *Min-Max-Scaler* as implemented in the *scikit-learn* Python-package. This method converts the original value x_i into a scaled value $\tilde{x}_i$ which lies in the range of $[a, b]$ using the following formula:

$$\tilde{x}_i = \frac{x_i - min_{i=1,\ldots,N}(x_i)}{max_{i=1,\ldots,N}(x_i) - min_{i=1,\ldots,N}(x_i)}(a - b) + b$$

To support the use of a *tanh* activation function in the output layer as well as following good practice observed in the literature the target range is set to $[-1, 1]$. After having trained the models on scaled data and generated predictions in the scaled range the respective backward transformation is applied to the predictions to get values in the original range.

6.1.4 Variable Selection

6.1.5 Network Architecture

6.1.6 Training Method

For the training of LSTM models using the *Keras* and *Tensorflow* packages the training procedure is characterised by a number parameters.

6.1.7 Testing Scenario

As testing method a self implemented month-wise leave-one-out cross validation is employed. That means in each step one month of data is selected as test data and the model trained on all remaining months. To limit the number of validation steps and thereby computational complexity the months to be selected as test data are limited to the time frame 01/2016 - 07/2017. However as training data the whole data set beginning in 01/2014 is used.

6.2 Results

In the following paragraphs I will illustrate the results that the FFNN and LSTM achieved in the univariate and multivariate cases. In each case the models will be compared among each other as well as with the respective linear reference model.

6.2.1 Univariate Models

In the univariate or autoregressive case the linear reference which in this case consist of the price value of the last day proves to be a very strong benchmark. In fact both the Feed Forward Neural Network as well as the Long Short Term Memory model after training result in predictions that resemble this benchmark very closely. When looking at the predictions over the whole testing period displayed in figures (6 and 7) the actual value of the target variable becomes indistinguishable from the model predictions as well as the reference predictions. When one looks at a close up of these time series for the front month contract for delivery in July 2016 (figures 10 and 11) one can see a slight difference between the model predictions and the lagged value. However this difference seems to result mostly from an overall shift in the level, while the relative movement over time seems to be almost identical in both cases. Overall the FFNN predictions seem to resemble the lagged value reference even more closely than those of the LSTM. This picture is relatively constant across the whole time frame. The strength of the current price as a predictor of tomorrows price becomes especially apparent when looking at the evaluation metrics in tables 8, 9, 10 and 11. Neither of the neural network models is able to outperform the reference in any metric, a picture that is constant across the time frame. Again the performance of the FFNN is close to the reference model due to the fact that the predictions coincide more closely. To determine weather this performance might be due to under- or over-fitting one might look at the behaviour of training and test loss across training steps. Since the training process is restarted for each test month in the cross validation, one would get a different curve for each month such as those pictured in figures 12 and 13 . To get a picture for the whole time frame and remove some of the volatility of the test error the data was averaged across test months for both model types and the results plotted in figures 8 and 9. The fact that in all cases there seems to be no visible additional loss reduction for the second half of the training process suggests that there is no under-fitting and performance of these models. Additionally the low difference between training and test error as well as the relative constant test error towards the end of the process speak against the possibility of over fitting.

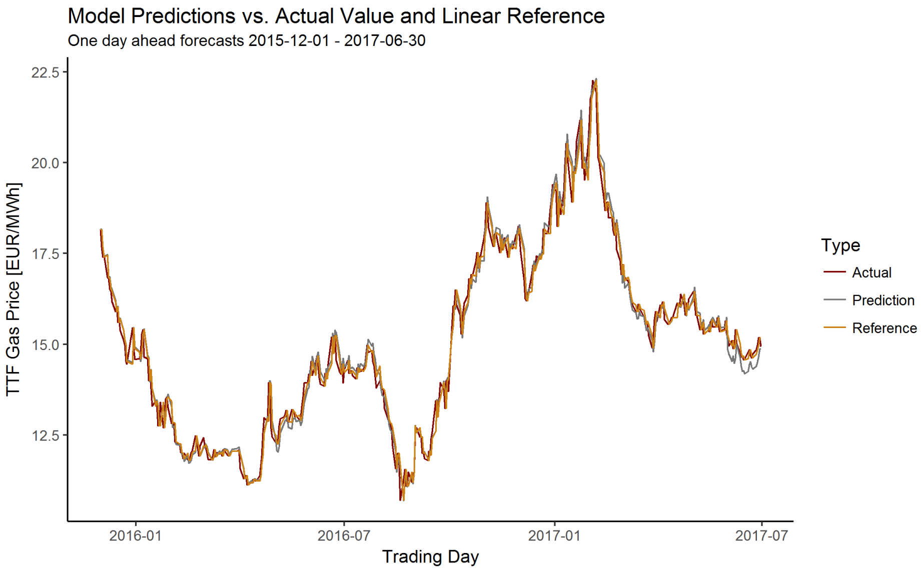

Figure 6: LSTM Predictions vs. Actual and Lagged Values for TTF Front Month 2016-17

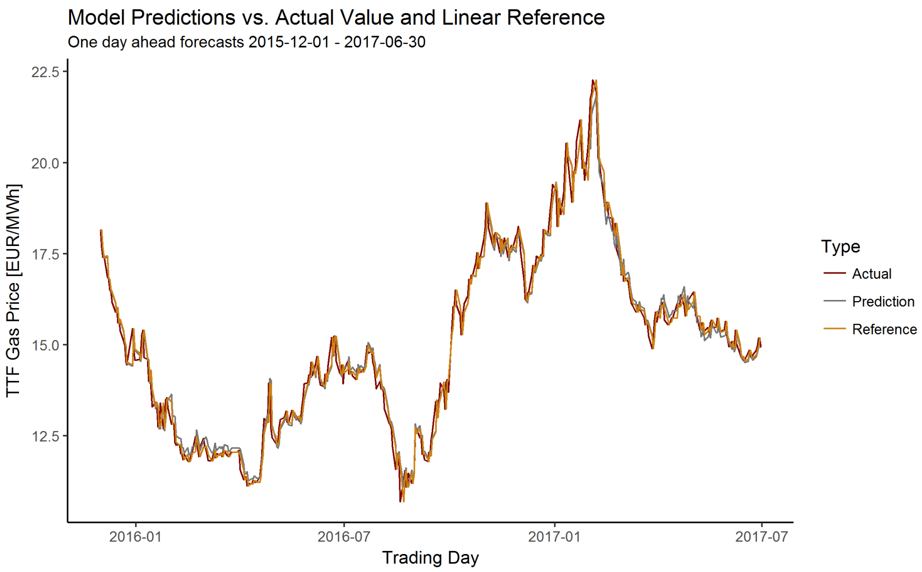

Figure 7: FFN Predictions vs. Actual and Lagged Values for TTF Front Month 2016-17

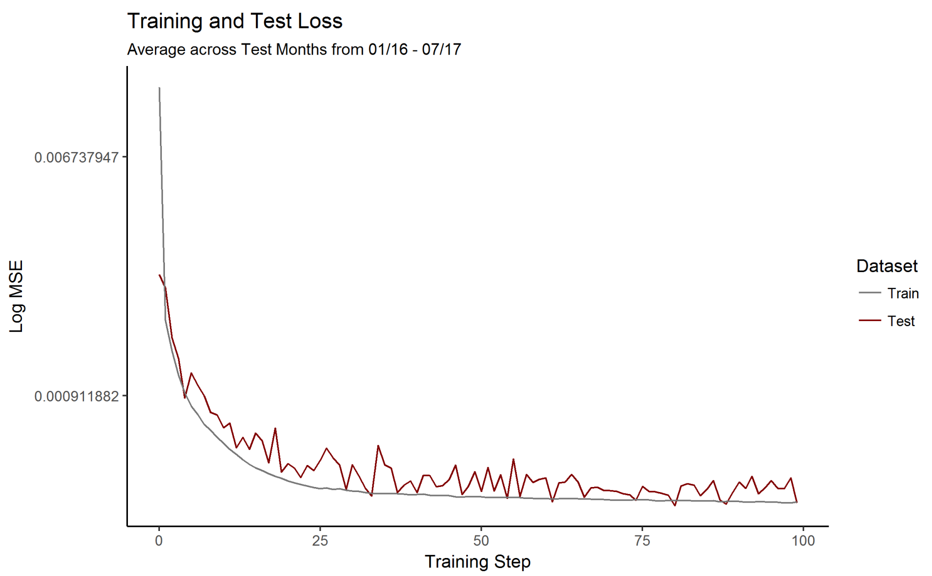

Figure 8: MSE along training steps averaged across Test Months - LSTM Model

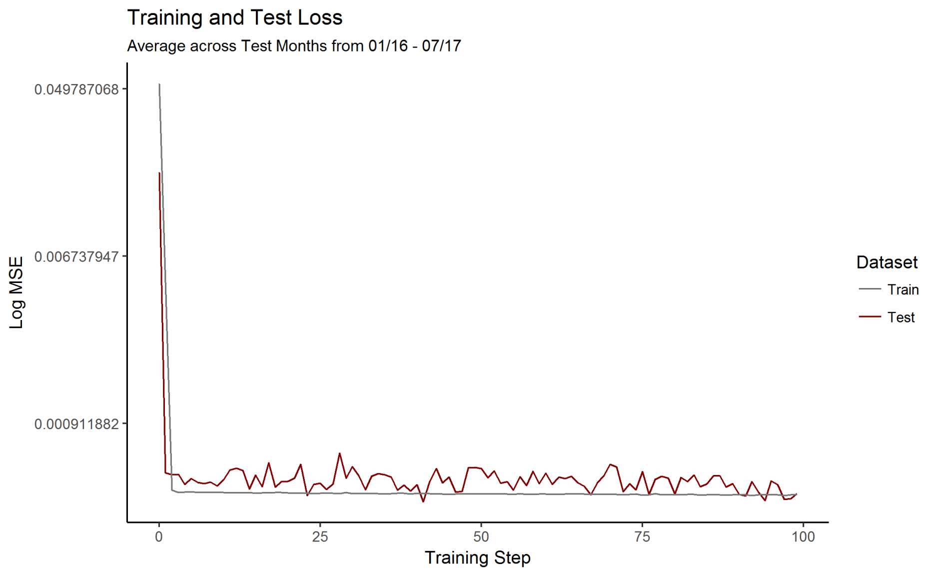

Figure 9: MSE along training steps averaged across Test Months - FFNN Model

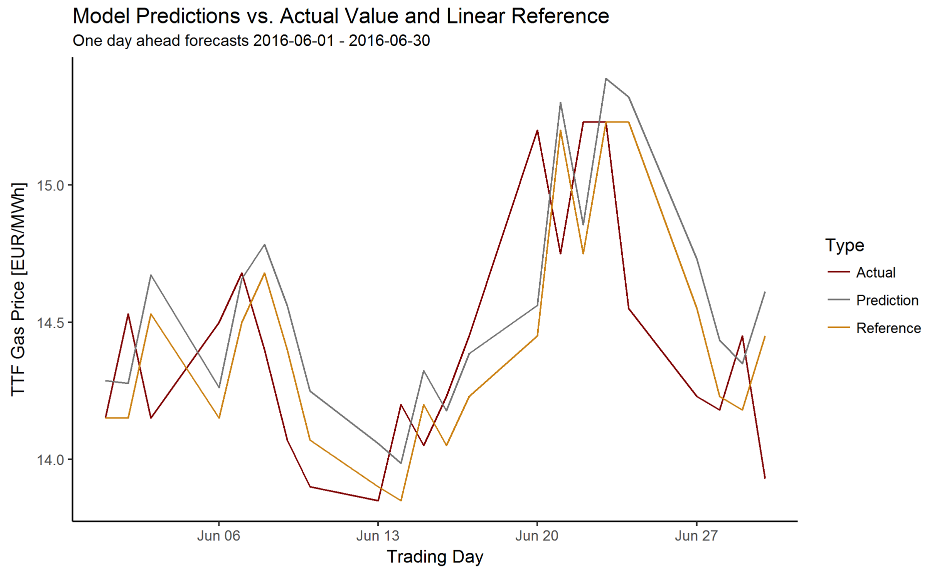

Figure 10: LSTM Predictions vs. Actual and Lagged Values for TTF July 2016 delivery

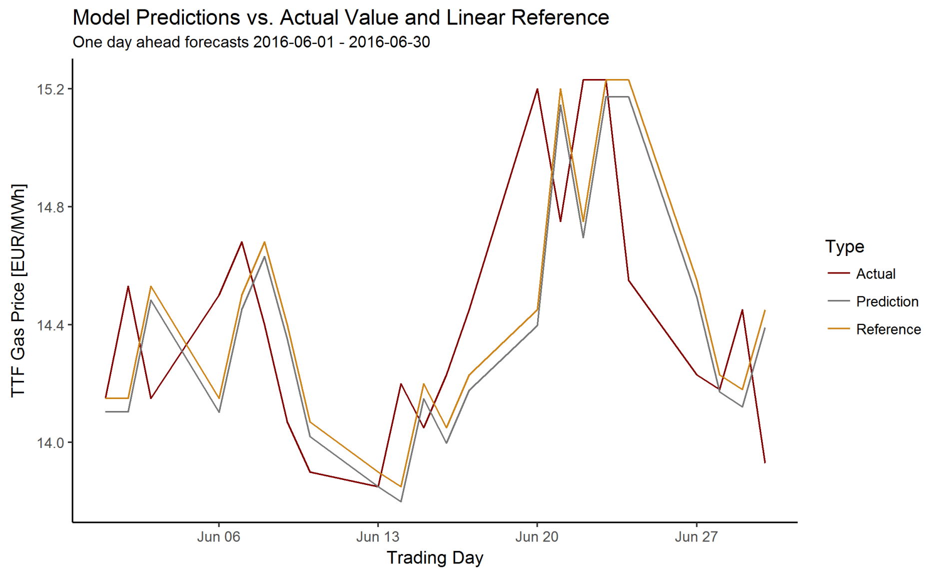

Figure 11: FFN Predictions vs. Actual and Lagged Values for TTF July 2016 delivery

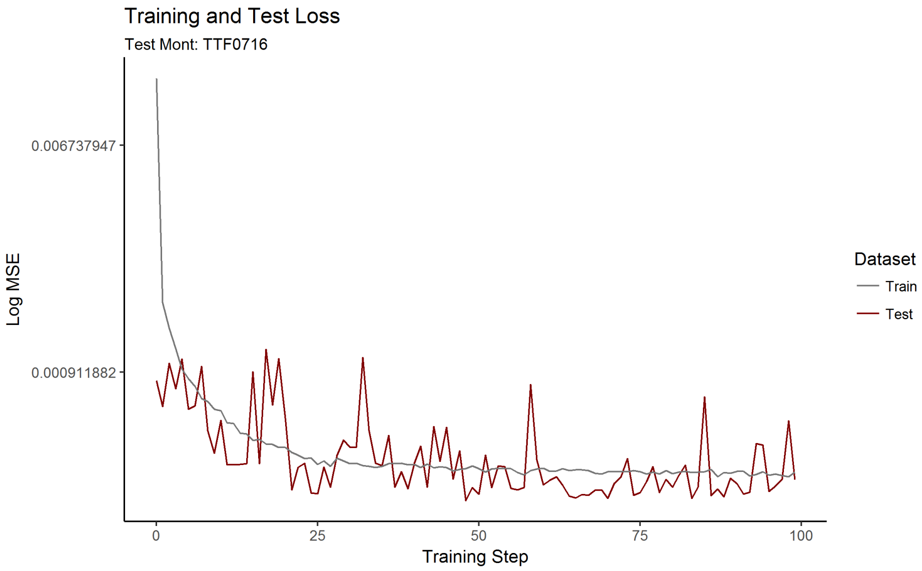

Figure 12: MSE along training steps for Test Month delivery July 2016 - LSTM Model

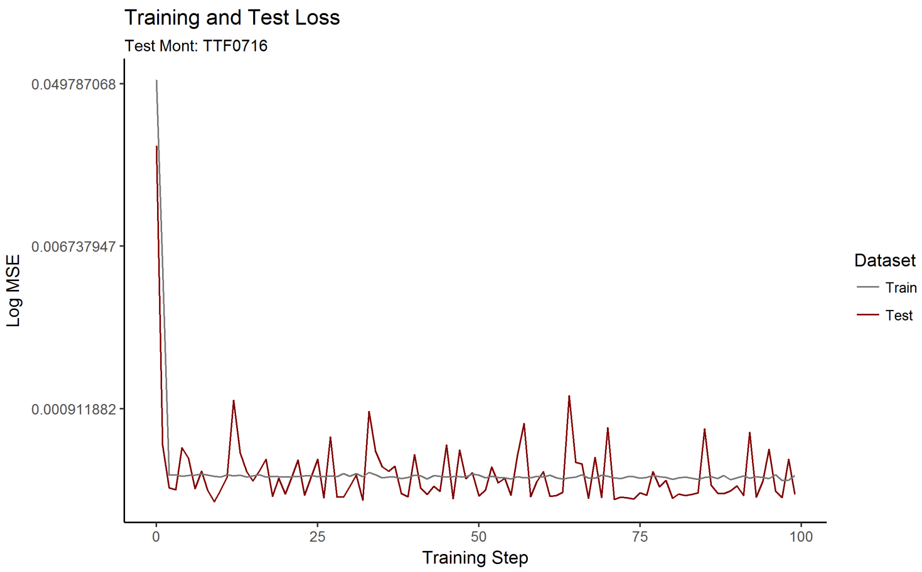

Figure 13: MSE along training steps for Test Month delivery July 2016 - FFNN Model

7 Conclusions

7.1 Results

7.2 Outlook

References

Abrishami, H. and V. Varahrami (2011): "Different methods for gas price forecasting," *Cuadernos de economa*, 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 gasAn 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 oilnatural 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. Muoz, J. L. Zamora, and R. Espnola (2011): "The effect of wind generation and weekday on Spanish electricity spot price forecasting," *Electric Power Systems Research*, 81, 1924–1935.

Des, S., A. Gasteuil, R. Kaufmann, and M. Mann (2008): "Assessing the factors behind oil price changes," .

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.

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. Espnola (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, Zrich.*

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. Caizares, 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.

A Figures

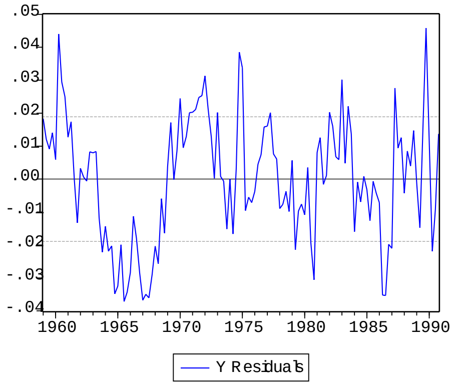

Figure 14: Estimated residuals (2) from model XXX. ...

B Tables

	Year	Correlation
1	2010	0.34
2	2011	0.10
3	2012	0.05
4	2013	-0.06
5	2014	-0.29
6	2015	0.78
7	2016	0.50
8	2017	0.74

Table 5: Annual Correlation of Oil and TTF price levels

	Year	Correlation	Sign
1	2010	-0.07	0.49
2	2011	0.10	0.57
3	2012	0.09	0.58
4	2013	0.01	0.60
5	2014	0.18	0.61
6	2015	0.20	0.63
7	2016	0.20	0.61
8	2017	0.20	0.59

Table 6: Annual Correlation of Oil and TTF Daily Returns

	Year	Correlation	Sign
1	2010	-0.07	0.49
2	2011	0.10	0.57
3	2012	0.09	0.58
4	2013	0.01	0.60
5	2014	0.18	0.61
6	2015	0.20	0.63
7	2016	0.20	0.61
8	2017	0.20	0.59

Table 7: Annual Correlation of Oil and TTF Daily Returns

Declaration of Authorship

I hereby confirm that I have authored this Bachelor's/Master's thesis independently and without use of others than the indicated sources. All passages which are literally or in general matter taken out of publications or other sources are marked as such.

Berlin, September 30, 2007

your name (and signature, of course)

	Month	LaggedValue	LSTM	FFNN
1	2016-01-01	1.85	1.96	1.76
2	2016-02-01	2.55	2.62	2.55
3	2016-03-01	1.44	1.45	2.18
4	2016-04-01	0.67	0.83	1.52
5	2016-05-01	2.17	2.24	2.51
6	2016-06-01	1.22	1.99	1.45
7	2016-07-01	2.05	2.29	2.05
8	2016-08-01	1.18	1.55	1.32
9	2016-09-01	2.48	2.17	2.63
10	2016-10-01	2.22	2.18	2.31
11	2016-11-01	1.82	2.09	1.82
12	2016-12-01	1.43	1.45	1.52
13	2017-01-01	1.60	1.55	1.65
14	2017-02-01	2.71	2.72	2.70
15	2017-03-01	2.04	2.50	1.87
16	2017-04-01	1.26	1.22	1.54
17	2017-05-01	1.13	1.09	1.55
18	2017-06-01	1.15	1.23	1.33
19	2017-07-01	1.03	2.34	1.04

Table 8: MAPE of models by target month (delivery)

	Month	LaggedValue	LSTM	FFNN
1	2016-01-01	0.29	0.31	0.28
2	2016-02-01	0.35	0.36	0.35
3	2016-03-01	0.18	0.18	0.27
4	2016-04-01	0.08	0.10	0.18
5	2016-05-01	0.27	0.28	0.31
6	2016-06-01	0.16	0.26	0.19
7	2016-07-01	0.30	0.33	0.30
8	2016-08-01	0.17	0.22	0.19
9	2016-09-01	0.29	0.26	0.31
10	2016-10-01	0.29	0.28	0.30
11	2016-11-01	0.30	0.35	0.30
12	2016-12-01	0.26	0.26	0.27
13	2017-01-01	0.28	0.27	0.29
14	2017-02-01	0.54	0.53	0.53
15	2017-03-01	0.40	0.48	0.37
16	2017-04-01	0.20	0.20	0.24
17	2017-05-01	0.18	0.17	0.25
18	2017-06-01	0.18	0.19	0.21
19	2017-07-01	0.15	0.35	0.16

Table 9: MAE of models by target month (delivery)

	Month	LaggedValue	LSTM	FFNN
1	2016-01-01	0.14	0.15	0.13
2	2016-02-01	0.18	0.18	0.17
3	2016-03-01	0.06	0.05	0.11
4	2016-04-01	0.01	0.01	0.05
5	2016-05-01	0.18	0.17	0.20
6	2016-06-01	0.04	0.10	0.05
7	2016-07-01	0.13	0.15	0.13
8	2016-08-01	0.04	0.08	0.05
9	2016-09-01	0.14	0.11	0.16
10	2016-10-01	0.13	0.13	0.14
11	2016-11-01	0.12	0.15	0.12
12	2016-12-01	0.10	0.11	0.11
13	2017-01-01	0.13	0.12	0.14
14	2017-02-01	0.37	0.37	0.36
15	2017-03-01	0.24	0.31	0.25
16	2017-04-01	0.05	0.07	0.08
17	2017-05-01	0.05	0.05	0.08
18	2017-06-01	0.05	0.06	0.06
19	2017-07-01	0.03	0.16	0.04

Table 10: MSE of models by target month (delivery)

	Model	CE	AUC	MAE	MeanPrices
1	Reference	0.54	0.81	0.27	15.07
2	lstm_min_sep_l20_b10_e100_n32	0.50	0.80	0.31	15.04
3	lstm_min_sep_l20_b10_e200_n16-8_univar	0.54	0.81	0.34	15.08
4	lstm_min_sep_l20_b10_e200_n16_TTFDA	0.52	0.81	0.33	15.08
5	lstm_min_sep_l20_b10_e200_n16_univar	0.52	0.78	0.33	15.12
6	lstm_min_sep_l20_b10_e200_n32_TTFDA	0.52	0.77	0.32	15.02
7	lstm_min_sep_l20_b10_e200_n32_univar	0.52	0.80	0.31	15.05
8	lstm_min_sep_l20_b10_e200_n8_TTFDA	0.52	0.81	0.34	15.15
9	lstm_min_sep_l20_b10_e200_n8_univar	0.50	0.81	0.32	15.06
10	lstm_min_sep_l20_e200_b10_CO2FM	0.54	0.80	0.34	15.06
11	lstm_min_sep_l20_e200_b10_CoalFM	0.52	0.81	0.33	15.08
12	lstm_min_sep_l20_e200_b10_EURGBPFX	0.51	0.81	0.33	15.08
13	lstm_min_sep_l20_e200_b10_EURUSDFX	0.51	0.80	0.32	15.08
14	lstm_min_sep_l20_e200_b10_ElectricityBaseFM	0.51	0.80	0.32	15.17
15	lstm_min_sep_l20_e200_b10_ElectricityPeakFM	0.52	0.78	0.33	15.15
16	lstm_min_sep_l20_e200_b10_NBPDA	0.60	0.58	0.35	15.23
17	lstm_min_sep_l20_e200_b10_NBPFM	0.57	0.79	0.34	15.13
18	lstm_min_sep_l20_e200_b10_OilFM	0.55	0.78	0.32	15.08
19	lstm_min_sep_l20_e200_b10_StorageNL	0.56	0.80	0.35	15.18
20	lstm_min_sep_l20_e200_b10_TTFDA	0.51	0.81	0.32	15.08
21	mlp_min_sep_l1_b10_e100_n32	0.52	0.81	0.33	15.12
22	mlp_min_sep_l20_b10_e100_n32	0.53	0.81	0.34	15.17
23	mlp_min_sep_l20_b10_e200_n16-8_ttfda	0.59	0.78	0.33	17.18

Table 11: Results of univariate minimum models 2016-2017