Charles University in Prague
Faculty of Social Science
Institute of Economic Studies

Bachelor's Project

Electricity market: Analysis and prediction of volatility

Vladimír Kunc

Supervisor: PhDr. Ladislav Krištoufek, Ph.D.

Study Programme: Ekonomie a Finance

Field of Study: Ekonomické teorie

July 30, 2015

Declaration

1. The author hereby declares that he compiled this thesis independently, using only the listed resources and literature.
2. The author hereby declares that all the sources and literature used have been properly cited.
3. The author hereby declares that the thesis has not been used to obtain a different or the same degree.

In Prague on July 30, 2015 Vladimír Kunc ....

Bibliographic note

Kunc, Vladimír. *Electricity market: Analysis and prediction of volatility.* 90 p. Bachelor thesis. Charles University, Fakulty of Social Sciences, Institute of Economic Studies. Supervisor PhDr. Ladislav Krištoufek, Ph.D.

Abstract

The last two decades can be characterized by restructuring of energy industry and the creation of new, competitive energy markets, where accurate forecasts of electricity prices and price volatility are valuable both to consumers and producers. The aim of this work is to analyse several models for prediction of the price volatility of electricity on the Czech Electricity Day-ahead market on price data provided by OTE, a.s. for years 2009-2014. This work compares 144 different models' configurations for three distinct classes of models — autoregressive models, GARCH models, and artificial neural network models. This work provides comparison based on five different criteria, each describing the model in different way.

Keywords: price prediction, volatility prediction, GARCH, neural networks, LSTM

Range of thesis: 64,536 characters or 11,034 words.

Abstrakt

Poslední dvě dekády jsou charakterizovány restrukturováním energetického průmyslu a vznikem nových, soutěživých energetických trhů, kde přesné předpovědi cen elektřiny a cenové volatility je cenná jak pro spotřebitele, tak pro výrobce. Cílem této práce ja popsat a porovnat několik modelů pro predikci cenové volatility na českém denním trhu s elektřinou na datech poskytnutých společností OTE a.s. za roky 2009 – 2014. Tato práce srovnává 144 rozdílných konfiguracích pro tři různé třídy modelu — autoregresivní modely, modely typu GARCH a modely založené na umělých neuronových sítích. Tato práce provádí srovnání modelů pomocí pěti různých kritérií, z nichž každe popisuje model z jiného pohledu.

Klíčová slova: predikce ceny, predikce volatility, GARCH, neuronové sítě, LSTM
Rozsah práce: 64 536 znaků nebo 11 034 slov.

Institute of Economic Studies
Bachelor thesis proposal

Overview

In recent years, the energy markets are getting more and more important. In the past, these markets were characterised by monopoly-based organisational structures but they do resemble free markets now and allows many entities to buy and sell energy on this market.One of the most important of energy markets is the electricity market which allows buying energy through bids and selling energy through offers to sell. The electricity market differs from others because the nature of electricity, it is very difficult to store it and also it has to be available on demand. Thus models for stock markets are not very good for use in the electricity market, because it is not possible - for example - keep electricity in stock or have customers queue for it.

Moreover, the demand and supply vary continuously which makes the electricity even harder to trade. Also, the electricity price time series differs a lot from electricity demand time series because they can exhibit variable means, major volatility and significant outliers. If traders were able to forecast the price volatility, they could better minimize their risks even if they would not have good models for predicting the prices themselves.

The aim of this work will be to analyse the price volatility on electricity and create a tool for prediction the volatility because the volatility determines how risky is the trade. Because there are many possible tools that can be used for the forecasting, this work also try to compare several possible approaches.

Thesis outline

1. How to measure volatility

2. Analysis of historical data

3. Short discussion of used methods

4. Discussion of results

Core Bibliography

1. Mills, T. C., & Markellos, R. N. (2008). *The econometric modelling of financial time series.* Cambridge, UK: Cambridge University Press.

2. Nogales, F. J., Contreras, J., Conejo, A.J., & Espinola, R. (2002). *Forecasting next-day electricity prices by time series models.* IEEE.

3. Ghysels, R., Santa-Clara, P., & Valkanov, R. (2004). *Predicting Volatility: Getting the most out of return data sampled at different frequencies.*

4. Mount, D. T., Ning, Y., & Cai, X. (2005). *Predicting price spikes in electricity markets using a regime-switching model with time-varying parameters.* Science Direct.

5. Amjady, N, & Keyinia, F. (2009). *Day-Ahead Price Forecasting of Electricity Markets by Mutual Information Technique and Cascaded Neuro-Evolutionary Algorithm.* IEEE.

Institut Ekonomických studií
Předběžná náplň práce

Úvod

V poslední době se trhy s energií stávají více a více důležité. V minulosti byly tyto trhy charakterizovány monopolistickými organizačními strukturami, ale v současnosti tyto trhy připomínají volný trh a dovolují mnoha stranám kupovat a prodávat energii. Trh s elektrickou energií patří k těm významnějším energetickým trhů a dovoluje kupovat energii. Tento trh se také liší od ostatních, protože díky podstatě elektřiny je velmi obtížné elektrickou energii skladovat a také je nutno zajistit, aby tato energie byla dostupná v případě potřeby. Z tohoto důvodu modely pro burzovní trhy se příliš nehodí k použití na trhu s elektrickou energií. Navíc, poptávka a nabídky elektrické energie se v čase neustále mění, což dělá elektřinu ještě těžší k obchodování. Časový vývoj cen elektřiny se také podstatně liší od časového vývoje poptávky, neboť často jeví výraznou volatilitu či výrazné vybočující hodnoty. Kdyby obchodníci byli schopni predikovat cenovou volatilitu, mohli by lépe minimalizovat jejich riziko i v případě, že nemají perfektní model pro předpovídání ceny samotné. Cílem této práce bude analyzovat cenovou volatilitu elektřiny a vytvořit nástroj pro predikci této volatility, neboť volatilita určuje rizikovost obchodu. Jelikož existuje mnoho různých nástrojů, které mohou být použity pro předpovídání, v této práci se také pokusím srovnat několik rozdílných přístupů k problému.

Struktura práce

1. Jak měřit volatilitu
2. Analýza historických dat
3. Krátký popis použitých metod
4. Shrnutí a diskuze výsledků

Základní literatura

1. Mills, T. C., & Markellos, R. N. (2008). *The econometric modelling of financial time series.* Cambridge, UK: Cambridge University Press.

2. Nogales, F. J., Contreras, J., Conejo, A.J., & Espinola, R. (2002). *Forecasting next-day electricity prices by time series models.* IEEE.

3. Ghysels, R., Santa-Clara, P., & Valkanov, R. (2004). *Predicting Volatility: Getting the most out of return data sampled at different frequencies.*

4. Mount, D. T., Ning, Y., & Cai, X. (2005). *Predicting price spikes in electricity markets using a regime-switching model with time-varying parameters.* Science Direct.

5. Amjady, N, & Keyinia, F. (2009). *Day-Ahead Price Forecasting of Electricity Markets by Mutual Information Technique and Cascaded Neuro-Evolutionary Algorithm.* IEEE.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and related works

In recent years, the energy markets are getting more and more important, moreover, as the energy industry around the world has undertaken significant restructuring [2, 23, 24], energy markets changed as well. In the past, these markets were characterized by monopoly-based organisational structures but they resemble free and competitive markets now [2, 9, 26]. The key goal of replacing the old structures by open market is to supply electrical energy to consumers with high reliability and low cost [2].

Commodities have been traded for decades, but electrical energy significantly differs from most other commodities and the whole market has several properties that are different from other markets - the electrical energy cannot be appreciably stored and the power system stability requires constant balance between supply and demand [2, 34]. Since the establishing of competitive energy markets around the world, electricity load and price prediction is becoming very important for both producers and consumers to maximize their profits.

Electricity price forecasting is extremely difficult not only because storing electrical energy is extremely inefficient but also because the demand for electricity depends on the structure of business intensity (weekdays vs. weekends, on-peak hours vs. off-peak hours) and weather (temperature, wind speed) [34]. Moreover, with recent increase in solar and wind electrical plants, the supply also depends on weather conditions. All these dependencies leads to price behaviour that is not present in any other markets - the price exhibits seasonalities at various levels (daily, weekly, annually) and abrupt, short-lived and generally unexpected spikes [33, 34]. Even though the cause of these spikes attracted extensive research in recent years, there is no consensus about the cause behind them and they are subjective and very difficult to forecast [37]. Thus the prediction of these spikes or generally of price volatility is crucial task for risk management of market participants.

The forecasting is crucial also in long-term (the producers might expand their production or the consumers might change their energy demand) and medium-term (six-month to one year) but our work focuses on the short-term volatility prediction for *day ahead* markets. The goal of this work is to compares several different models for volatility prediction — autoregressive models, GARCH models, and artificial recurrent neural network models. The description of various models used for the prediction together with the description of several measurements of volatility used in electricity price forecasting is provided in Chapter 2. Historical data analysis is in Chapter 3 and finally, comparison and evaluation of all used models is provided in Chapter 6

There are several types of related works, first and most common are the works studying price and/or volatility prediction of various financial time-series, however, as electricity market differs significantly from usual markets, not all common approaches are applicable. The second type of related works discuss price prediction on electricity markets, however, the direct prediction of electricity price is a complex process and the volatility can be simply computed from the predicted prices as *realized volatility.* The third type of related works is the most similar to this work — this type represents works studying forecasting of electricity price volatility.

Since the year 2000, the research of electricity price prediction has advanced — while being practically non-existent before 2000, the number of Scopus indexed journal articles reached 206 and the number of conference papers reached 274 in the year 2013 [34]. Most of these articles uses some kind of neural networks or time series analysis for price prediction, therefore this chapter focuses mostly on these two tools.

Haugom et al. [16] compare several models for predicting day ahead Nord Pool forward price volatility. The authors compare GARCH, Fractionally Integrated GARCH, RiskMetrics^{TM}, and HAR-CV-JV models.

GARCH:

$$\sigma_{t+1}^2 = \omega_{t+1} + \sum_{i=0}^{q} \alpha_i \epsilon_t^2 + \sum_{j=0}^{p} \beta_j \sigma_{t-j}^2 \tag{1.1}$$

RiskMetrics^{TM}:

$$\sigma_{t+1}^2 = \omega + (1-\lambda)\epsilon_t^2 + \lambda\sigma_t^2 \tag{1.2}$$

FIGARCH:

$$\sigma_{t+1}^2 = \sigma^2 + \lambda(L)(\epsilon_{t+1} - \sigma^2) \tag{1.3}$$

HAR-CV-JV:

$$RV_{t+1} = \beta_0 + \beta_1 CV_t + \beta_2 CV_{t-5,t} + \beta_3 JV_t + \beta_4 JV_{t-5,t} \tag{1.4}$$

These are only the most important models used by Haugom et al., more details about used models are available in [16].Authors used two different means of model comparisons. First of all, they fit model specified in [16], then they compared the explained variance by the model (R^2). They also compared models based on *mean absolute percentage error* (MAPE) and *symmetric MAPE* (sMAPE) for the last year of the sample. Authors have found that there is no clear evidence of a superior model from tested models when predicting day ahead volatility for electricity forward contracts, detailed results with discussion are again available in the original study [16].

Nogales et al. [26] used two different time-series tools to predict the electricity price for the electricity markets of mainland Spain and California. Their research focused on short-term forecasts — especially day-ahead price predictions. The data used were hourly values of the price and also of the demand for electricity. First of the approaches was a *dynamic regression* — a model, where the price at hour t is considered related to the past price at hours $t-1$,$t-2$,..., and also to the values of demand at hours t, $t-1$, ..., more specifically:

$$p_t = c + \omega^d(B)d_t + \omega^p(B)p_t + \epsilon_t \tag{1.5}$$

where p_t is the price at hour t, d_t is the demand at hour t, c is a constant, functions $\omega^d(B)$ and $\omega^p(B)$ are polynomial functions of the backshift operator and ϵ_t is the error term, which is assumed to be randomly drawn from $N(0, \sigma^2)$. More details about the polynomial backshift function and the iterative scheme used for parameter selections is available in [26]. Another used approach was a *transform function model*, which assumes that both the price and demand series are stationary:

$$p_t = c + \omega^d(B)d_t + N_t \tag{1.6}$$

where p_t is the price at hour t, d_t is the demand at hour t,and $\omega^d(B)$ is the polynomial function of the backshift operator. The N_t is a disturbance term following and ARMA model:

$$N_t = \frac{\Theta(B)}{\theta(B)}\epsilon_t \tag{1.7}$$

where ϵ_t is the error term, and both $\Theta(B)$ and $\theta(B)$ are the polynomial functions of the backshift operator, more details are available in [26]. The study found both approaches useful as predicting tools — the averages error for the Spanish market were roughly 5 % and 3 % for the Californian market.

Gonzalez et al. [14] used a hybrid GARCH model for modeling the electricity price series. The study highlights three different parts that make the electricity price:

- The cost of generation (efficiencies of power plants)
- The cost of the inputs of generation
- The margin

The most simple part of the price is the cost of generation as it stable in the short run and also easy to determine, however, the cost of the inputs of generation is a bit more problematics. Inputs of generation — e.g., coal, gas, and carbon prices — represent another price series that might be very volatile as well. And last but not least is the margin — when the residual capacity (margin) of the generators decreases too much, prices jump. The hybrid model used in the study has three different sources of input — historical spot prices, fundamental model, and margin. The fundamental model describes the costs of electricity generation, authors used supply stack modelling where they ranked different generation units. Generation costs were calculated for each of the generation units based on data from six representative power plants. Then the supply stuck is built for each day and the system cost (GBP/MWh) is calculated for each day. The resulting model seems to well approximate the lower price bound (with exceptions when the electricity is sold under lower price than is the approximated cost). The fundamental model together with the historical prices were used in three different models — ARX, ARX-GARCH, and

ARX-GARCHX, these models are than compared with respect to *mean absolute percentage error*. The whole model was used for predicting average daily electricity price.

Zhang et al. [36] also use GARCH models, more specifically they use ARMAX-AR-GARCH model for day-ahead prediction of electricity prices. The model uses historical hourly price data to perform rolling forecast for the next day prices $p_{\bar{r_{\hat{n}}}}$, which are then used together with the hourly load data by *least-square support vector machine* (LS-SVM). The advantage of such model is that a LS-SVM are able to reveal the non-linear relationships between load and price. Moreover, the LS-SVM has a good generalization performance and global optimization. The LS-SVM is defined as optimization problem:

$$\min(\frac{1}{2}||\omega||^2 + \frac{1}{2}\gamma\sum_{i=1}^{k}\xi^2) \tag{1.8}$$

with equality constraint:

$$y_i = \omega^T\phi(x_i) + b + \xi, i = 1, 2, \ldots, k \tag{1.9}$$

The overall hybrid model combing LS-SVM with the ARMAX-AR-GARCH model is with both plain ARMAX-AR-GARCH model and feed forward artificial neural network trained with backpropagation. The hybrid model with 2.26 % *mean absolute percentage error* (MAPE) is slightly better than the plain model with 2.72 % MAPE and significantly better than the neural network with 4.11 % MAPE.

Chapter 2

Used models

Two different approaches — time-series analysis and neural networks — were used for purposes of this work, more specifically, different AR and GARCH models and recurrent neural networks with LSTM units.

2.1 Realized volatility

Realized variance is an ex-post of the volatility based on the cumulative intradaily squared returns. According to [5, 6, 16], the daily realized volatility at the end of day t is defined in following way:

$$RV_{t+1}(\Delta) = \sum_{j=1}^{\frac{1}{\Delta}} r_{t+j\Delta,\Delta}^2 \tag{2.1}$$

with $r_t = p_t - p_{t-1}$ where p_t is log price at time t. The realized variation is the sum of the intradaily returns with Δ being the return period (e.g. 30 minutes or 1 hour). Thus when $\Delta \to 0$, $RV_{t+1}(\Delta)$ measures the latent integrated volatility perfectly in the absence of jumps [6, 10, 16]:

$$RV_{t+1}(\Delta) \to \int_t^{t+1} \sigma^2(i)di \tag{2.2}$$

Also, according to quadric variation theory, the realized volatility is a consistent estimator of the integrated volatility in the absence of jumps [16].

Moreover, when jumps are present, the realized variance measure may be defined by a continuous integrated variance component and a discontinuous jump component:

$$RV_{t+1}(\Delta) \to \int_t^{t+1} \sigma^2(i)di + \sum_{t<i\leq t+1} \kappa^2(i) \tag{2.3}$$

assuming the log price is determined by jump-diffusion process [16]:

$$dp(t) = \mu(t)dt + \sigma(t)dW(t) + \kappa(t)dq(t), t \in [0, T] \tag{2.4}$$

, where $p(t)$ s the log price, $\mu(t)$ is the drift, $\sigma(t)$ is a strictly positive, left continuous with right limits stochastic volatility process, W(t) is a standard Wiener process and q(t) is a counting process assuming the value 1 if a jump occurs at time t and 0 otherwise with intensity $I(t)$ and size $\kappa(t)$.

The bi-power defined in (2.5) is a consistent estimator of the integrated variance under the presence of jumps [7, 16].

$$BV_{t+1}(\Delta) = \mu_1^{-2} \sum_j = 2^{\frac{1}{\Delta}} |r_{t+j\Delta,\Delta}||r_{t+(j-1)\Delta,\Delta}| \tag{2.5}$$

with $\mu_1 = \sqrt{\frac{2}{\pi}} \doteq 0.79788$. Thus, with $\Delta \to 0$, the probability of picking up discontinuities disappears in the limit [7, 16]:

$$BV_{t+1}(\Delta) \to \int_t^{t+1} \sigma^2(i)di \tag{2.6}$$

General detailed proof is available in [7]. Hence, consistent estimator of the discontinuous jump component is the difference of the realized variance and bipower variance measures:

$$RV_{t+1}(\Delta) - BV_{t+1}(\Delta) \to \sum_{t<i\leq t+1} \kappa^2(i) \tag{2.7}$$

However, this work is based on the assumption of absence of jumps because the empirical results presented in [16] do not yield better accuracy for neither of the contracts price volatility and the estimator from (2.1) is less complicated and moreover, the bi-power measure is sensitive to the presence of "zero" returns in the sample, which is probable in examined markets [16]. More detailed description of volatility measurements and prediction on electricity markets is available in [16, 22]

2.2 Recurrent Neural Network with LSTM units

Recurrent Neural networks constitute a special class of neural networks where connections between units (neurons) form a directed cycle. RNN are able to approximate well many different timeseries and dynamic systems [4], however classical recurrent neural networks are hard to train due to the *problem of vanishing gradients* [12], however alternative using *long short-term memory* (LSTM) units was proposed in [17].

A LSTM unit is able to hold information for a long time and consists of one or more memory units and several gates. Original LSTM unit were not able to forget information, thus the *forget gate* was added in [12].

Recurrent neural networks with LSTM units consists of several fully interconnected LSTM units and an output layer, which processes the output from individual LSTM unit into the final prediction. Each LSTM unit is able to hold a value for an arbitrary length of time - the length does not have to specified in advance, which provides an advantage over models with parametric length of the history — AR(n) models and *feed-forward neural*

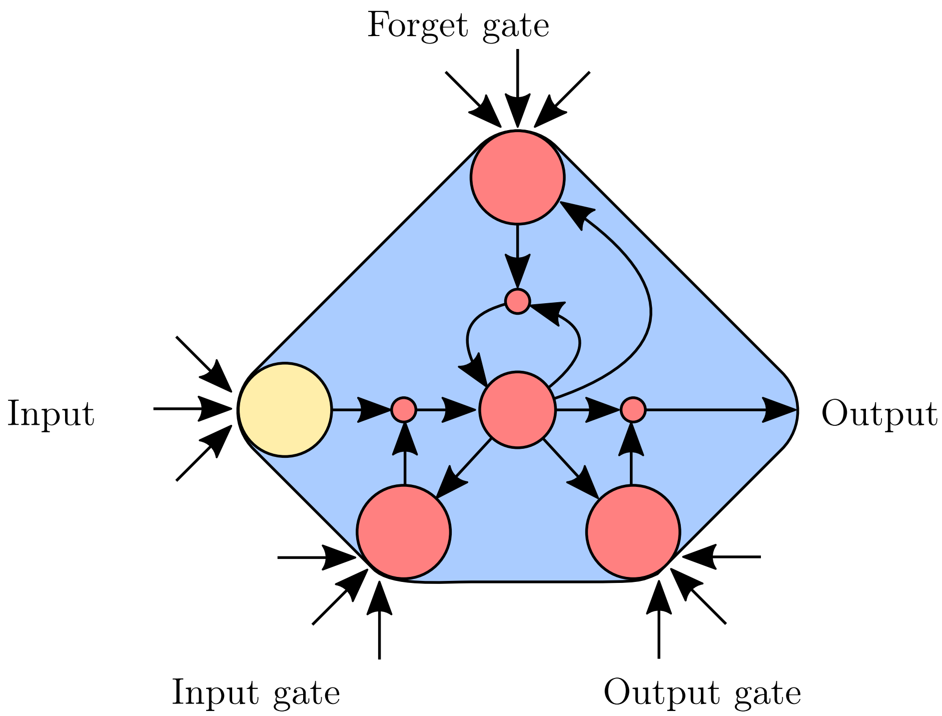


Figure 2.1: Schema of LSTM unit with three gates — input, output and forget.

networks with sliding window models, Moreover, the value stored in LSTM unit is able to be dynamically changed if needed or even to be forgotten using the *forget gate*. The number of LSTM units significantly influence the performance of the neural network.

Each gate in a LSTM units consists of a sigmoid function taking in the weighted sum of inputs, more specifically, the output from gate y is a function of the gate's inputs $y = s(\sum w_i x_i)$ where $s(x)$ is a squashing function [12]. When the output from input gate is close to zero, it also zeroes the input going into the LSTM unit as the gate prevents it from entering another subunit in the LSTM block. When the output from the forget gate is close to zero, the LSTM unit will forget the stored value. The last, output gate determines when the unit should output the value stored in its memory. Detailed schema of the LSTM unit is shown in Figure 2.1.

2.2.1 Used configuration of LSTM models

Three different classes of configuration of LSTM recurrent neural network were used in order to examine thorougly the performance of this class of models. Each of these classes contained 10 models with varying number of LSTM units as the number of LSTM units controls the complexity of the network and also the learning capabilities.

First class consists of n LSTM units with *linear* output layer where $n = 2^i, i = 0, 1, \ldots 9$. This class will be further called *LSTM linear n* model where n is the number of LSTM units. This model takes as an input the volatility x_{t-1} and outputs volatility y_t at time t, however, the model needs to be trained on sequential data as it can remember used values. The linear output layer outputs just a weighted sum of outputs of individual LSTM units, therefore it the output can reach arbitrary value and is not limited to any interval. On the other hand, the linear output layer also limits the ability of LSTM units to reinforcement their prediction each unit has a weighted share in the overall output and if one LSTM unit outputs an extreme value, other LSTM units have to output appropriately large values.

Unlike the first class, the second LSTM class of models uses a sigmoid output layer. The main disadvantage of sigmoid output layer is that the output is bounded to an interval and is not unbounded as in previous model because the sigmoid function outputs only values from interval $[0, 1]$, the volatility data are scaled down to this interval, the network is then trained on the scaled data and the output is then scaled back to original interval, however, as the volatility itself is not bounded, the data were first bounded by 95 % quantile and only then the data were scaled to the interval $[0, 1]$ and fed to the network. This type of LSTM networks has also its advantages — the sigmoid output layer allows the LSTM units to reinforcement outputs of each other, for example, if LSTM unit Aoutputs a high value, the overall output will be close to 1, however, if another LSTM unit B outputs also a high value, the output will be just a bit closer to 1, it will not produce an extreme output value as the linear output layer would provide. This allow more complex combination of outputs from the LSTM units. As in for the previous class, 10 models with different number n of LSTM units were used — for each $n = 2^i, i = 0, 1, \dots 9$.

Third class of LSTM models uses, unlike the previous models, hourly price data. This class uses linear output layer as the first class of LSTM models, however the input values are not the daily realized volatility but a vector of 24 hourly prices for the related day. This type of model work in two stages, firstly, it tries to predict the set of 24 day-ahead hourly prices similarly as in [1, 3, 26, 31, 35], secondly, it computes the realized volatility from the price prediction, this volatility then serves as the output of the model. Instead of using a sequence of 24 hourly data, the network directly uses the whole vector of hourly prices at one time and also produces a vector of 24 hourly prices — the input and outputs are therefore points from 24 dimensional space. However, this type of model is a bit counter intuitive as the volatility is usually wanted for price prediction and not the vice versa.

2.3 Autoregressive models

Another commonly used types of models for predicting volatility are different types of autoregressive models, these model vary in complexity but even the very simple ones are capable of good prediction as in [24]. Autoregressive models, generally, are models where the output variable depends linearly on its own previous values and also on the error term, however, the price volatility of electricity does not really meet such criteria as it depends on many different variables, some are not even observable, and its errors are not independent in time, therefore while this class of model is be used for prediction, many different properties do not not hold as the model assumption are not met, thus such model can be only evaluated in terms of error on out-of-sample data. For the purpose of this work, many different models were used and they are described in following sections.

2.3.1 Basic models

Every used model is based on one of following 8 models:

2.3.1.1 Model *const only*

As the name of the model implies, this model is a very simple model based on a constant only. This model assumes that the volatility is constant:

$$X_t = \beta_0 + \epsilon_t \tag{2.8}$$

Even though it is unlikely that the price volatility of electricity follows such model, this model is useful for comparison with other models.

2.3.1.2 Model *x1*

This is the simplest autoregressive model AR(1):

$$X_t = \beta_0 + \beta_1 X_{t-1} + \epsilon_t \tag{2.9}$$

2.3.1.3 Model *HAR*

This model tries cover both short term variations and long term variation in the price volatility of electricity using the average value from last week and last month together with the value from previous day:

$$X_t = \beta_0 + \beta_1 X_{t-1} + \beta_2 \frac{1}{7} \sum_{i=1}^{7} X_{t-i} + \beta_3 \frac{1}{30} \sum_{i=1}^{30} X_{t-i} \tag{2.10}$$

2.3.1.4 Model *x1-x7*

This model is an autoregressive model AR(7) that has one week history as inputs:

$$X_t = \beta_0 + \sum_{i=1}^{7} \beta_i X_{t-i} \tag{2.11}$$

2.3.1.5 Model *x1,x2,x7*

This model uses restricted set of variables from previous model as these variables have better economical and natural intuition behind them — the term X_{t-7} is there from the reason of weekly periodicity in the data and the other two terms X_{t-1} and X_{t-2} represents short term memory of the volatility.

$$X_t = \beta_0 + \beta_1 X_{t-1} + \beta_2 X_{t-2} + \beta_3 X_{t-7} \tag{2.12}$$

2.3.1.6 Model *x1,x7,x14,x28*

This model uses lagged values with natural economic interpretation as it assumes weekly periodicity and long term memory up to 4 weeks, however, the values are lagged always by week + 1 day, which is not intuitive — but in preliminary experiments, this model achieved better experimental results than the following one.

$$X_t = \beta_0 + \beta_1 X_{t-1} + \beta_2 X_{t-7} + \beta_3 X_{t-14} + \beta_4 X_{t-28} \tag{2.13}$$

2.3.1.7 Model *x1,x6,x13,x27*

This model is similar to the previous one but with better interpretation as the values are directly lagged by one week.

$$X_t = \beta_0 + \beta_1 X_{t-1} + \beta_2 X_{t-6} + \beta_3 X_{t-13} + \beta_4 X_{t-27} \tag{2.14}$$

2.3.1.8 Model *x1-x30*

This model tries to incorporate the monthly history of the realized volatility:

$$X_t = \beta_0 + \sum_{i=1}^{30} \beta_i X_{t-i} \tag{2.15}$$

2.3.2 Dummy variables

Each of the 8 basic models has an extended version which has some dummy variables for controlling different changes and periodicities. Each extension has a special suffix in the name of the model - eg. model *HAR dmy* or *x1,x2,x7 w*.

2.3.2.1 Suffix *w*

The suffix w means that the model has been extended by a dummy variable w which is equal to 1 if the predicted day is on weekend and 0 otherwise, more formally:

$$w(t) = \begin{cases} 1 & \text{if } t \text{ is on weekend} \\ 0 & \text{otherwise} \end{cases} \tag{2.16}$$

2.3.2.2 Suffix *d*

This suffix indicates that the model was extended by dummy variables indicating day in week. The base day is Monday, each other day has a dummy variable.

2.3.2.3 Suffix *m*

When the model has the suffix m in the name, it means that the model has been extended by dummy variables for each month in year with the base month January.

2.3.2.4 Suffix y

The dataset contains years 2010 – 2014, then when the model has this suffix, it means that it have dummy variables for years 2010–2013 with the base year 2014 as it is the most recent.

2.3.3 Modified dataset

Used models are evaluated based on two different error measures — *root mean squared error* (RMSE) and *mean absolute error* (MAE), however, the models were estimated using OLS regression which minimazes only the *root mean squared error* and not the *mean absolute error*, therefore to find some compromise between these two optimization criteria, some models were fitted to a dataset that has an upper bound on the data. Therefore if there are some unexpected spikes in the dataset, they have only much smaller influence in the bounded dataset than in the original one - the influence is still significant though, for example, there are days in the original dataset, for which the realized volatility reaches values over 200, while the 75 % quantile is less than one, that means that if these days are not predicted correctly, they will add about 40,000 to the sum of squared errors. Therefore the optimization procedure minimazing the squared error will results in very high absolute error. Models fitted to the bounded dataset has the suffix *or* in their name.

2.4 GARCH models

One of the common approaches represents the family of generalized autoregressive conditional heteroskedasticity models (GARCH), however, these models are not suitable for electricity intradaily data as there is a very strong periodicity, therefore these models are using the daily price which is computed as a weighted average of hourly prices. GARCH models are very often used for price and volatility prediction on electricity markets,e.g., they were used in [11, 14, 16, 22, 36]. Several different models from GARCH family were used — they are shortly described in following sections.

2.4.1 Standard GARCH model (sGARCH)

This model is is defined as :

$$\sigma_t^2 = \left(\omega + \sum_{j=1}^{m} \zeta_j v_{jt}\right) + \sum_{j=1}^{q} \alpha_j \epsilon_{t-j}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2 \tag{2.17}$$

where σ_{t-j}^2 denotes the conditional variance, ω the intercept and ϵ_t^j denotes the residuals from a mean filtration process [13]. The mean filtration process is further described in [13]. The persistence parameter $\widehat{P}$ is defined as:

$$\widehat{P} = \sum_{j=1}^{q} \alpha_j + \sum_{j=1}^{p} \beta_j \tag{2.18}$$

2.4.2 Integrated GARCH (iGARCH)

This is an special version of standard GARCH model that assumes persistence $\widehat{P} = 1$ [13], this assumption is also imposed in the estimation procedure.

2.4.3 Exponential GARCH model (eGARCH)

This is exponential version of standard GARCH model that defines a relationship for $\log_e(\sigma_t^2)$ instead of just σ_t^2 [13]. This model is defined as:

$$\log_e(\sigma_t^2) = \left(\omega + \sum_{j=1}^{m} \zeta_j v_{jt}\right) + \sum_{j=1}^{q} (\alpha_j z_{t-j} + \omega_j(|z_{t-j}| - E|z_{t-j}|)) + \sum_{j=1}^{p} \beta_j \log_e(\sigma_{t-j}^2) \tag{2.19}$$

where α_j captures the sign effect and ω_j the size effect [13] and

$$E|z_t| = \int_{-\infty}^{\infty} |z| f(z, 0, 1, \ldots) dz \tag{2.20}$$

.

2.4.4 GJR-GARCH (gjrGARCH)

This GARCH models has been extended in a such way that it models positive and negative shocks on the conditional variance asymmetrically by using the indicator function I [13]:

$$\sigma_t^2 = \left(\omega + \sum_{j=1}^{m} \zeta_j v_{jt}\right) + \sum_{j=1}^{q} \left(\alpha_j \epsilon_{t-j}^2 + \gamma_j I_{t-j} \epsilon_{t-j}^2\right) + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2 \tag{2.21}$$

where the function I is defined as:

$$I(\epsilon) = \begin{cases} 1 & \epsilon \leq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.22}$$

and γ_j represents the leverage term [13].

2.4.5 Asymmetric power ARCH (apARCH)

According to [13], the asymmetric power ARCH model allows for both leverage and the Taylor effect. The model is defined in following way:

$$\sigma_t^\delta = \left(\omega + \sum_{j=1}^{m} \zeta_j v_{jt}\right) + \sum_{j=1}^{q} \left(\alpha_j (|\epsilon_{t-j}| - \gamma_j I_{t-j} \epsilon_{t-j})^\delta\right) + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^\delta \tag{2.23}$$

where δ is a Box-Cox transformation of σ_t, $\delta \in \mathbb{R}^+$. The γ_j is the coefficient in the leverage term.

2.4.6 Used parametrization

Each of the models above was used in two different parametrization — both are very simple and they are meant just to provide example as the GARCH(1,1) is usually good enough [15] — but there are many different parametrization that are often much better [15], however the parameter optimization is greatly out of the scope of this work as it is computationally very costly.

Two different mean models were used — ARMA(1,1) and ARMA(1,0) — together with two different GARCH parametrization — GARCH(1,1) and GARCH(1,0) — resulting in total of 16 different models, models eGARCH and iGARCH were used only in GARCH(1,1) parametrizations.

2.4.7 Adjusting the prediction

The final prediction using GARCH models is done in two stages, firstly, the GARCH model is fitted in usual way and the volatility (σ) is calculated, in the second stage, the volatility calculated from GARCH models is used as a variable in OLS regression with target variable the realized volatility because the GARCH models often tend to underestimate the volatility.

$$RV_t = \beta_0 + \beta_1 \sigma_t \tag{2.24}$$

where β_0 and β_1 are estimated coeffiecents through OLS regression, RV_t is the realized volatility at time t and σ_t is the volatility at time t from a GARCH model.

Chapter 3

Data

We have used data from Czech Electricity Day-ahead market for years 2009 – 2014. These data consist of one Excel file per year provided by OTE, a.s. and are available at `http://www.ote-cr.cz/statistika/rocni-zprava`. There are hourly prices in euros per megawatt hour, however, the prices until 31.1.2009 are calculated from the prices in CZK/MWh using the exchange rate provided by the Czech National Bank, since then the data are only in EUR/MWh and the marginal prices in CZK/MWh is only informative and calculated using the price in EUR/MWh.

Each file contains hourly marginal prices, the amount of electricity sold and bought, the amount of electricity exported to or imported from Slovakia. It also contains variables calculated from previously mentioned values - the total price for the electricity (EUR) and also several different daily averages, full description of published information is available at `http://www.ote-cr.cz/`. For the purpose of this work, only the hourly marginal prices were used for calculating the volatility and for prediction of day-ahead hourly prices.

3.1 Hourly prices of electricity

First, we describe the raw data that were used for predicting the electricity prices and for calculating the volatility data. The resulting volatility data that are used through most of the experiment are described in section 3.2.

3.1.1 Data description

The data are containing 52584 samples taken hourly for years 2009 – 2014. The maximum price reached 170 /MWh, while the minimum price reached -150 /MWh, which would be very unexpected in other commodities markets but not as much on the electricity market — the electricity production is a very complex process and the electricity is sold under negative price occasionally in order to maintain the sensitive balance of electricity production and consumption. The average electricity price is 40.69 /MWh, however, as there are many periodicities and seasonalitites present in the data [33, 34], the average electricity price

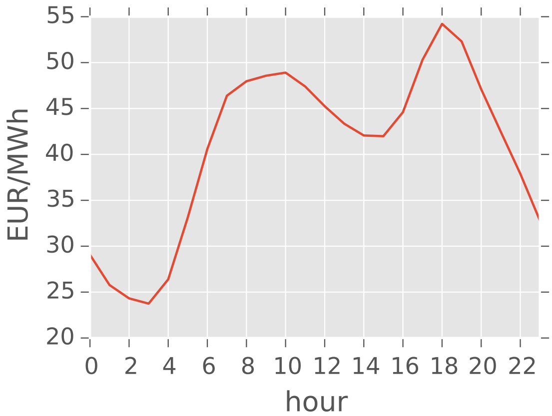


Figure 3.1: Mean hourly prices over years 2009-2014

is not very informative. Basic statistical description of the data is available in Table 3.1, while the individual periodicities are discussed in following section.

count	52584
mean	40.69
std	16.57
min	-150.00
25%	30.58
50%	40.15
75%	51.00
max	170.00

Table 3.1: The basic description of used price data

3.1.2 Periodicity

There are several periodicities (or seasonalities) present in the data — daily, weekly and yearly periodicities [33, 34]. The average daily periodicity is shown in Figure 3.1, however, the average does not show the variance of the data — therefore all daily prices are plotted in Figure A.1, where are also broken down to individual years and quarters thus showing that the trends are a bit different in each year and quarter. The price is also changing over the years as can be seen in Fig. 3.2, where the average daily price is computed for each year in the dataset. The graph shows that the hourly price is not just going up or down but also that the structure of average hourly prices has changed, however, it is impossible to determine from the data whether it was caused by structural changes in the market or just, for example, by different weather in those years. Moreover, the structure of average price is also changing within the year — there are significant differences in quarterly averaged daily periodicities as shown in Fig. 3.3, this has natural interpretation since in first and

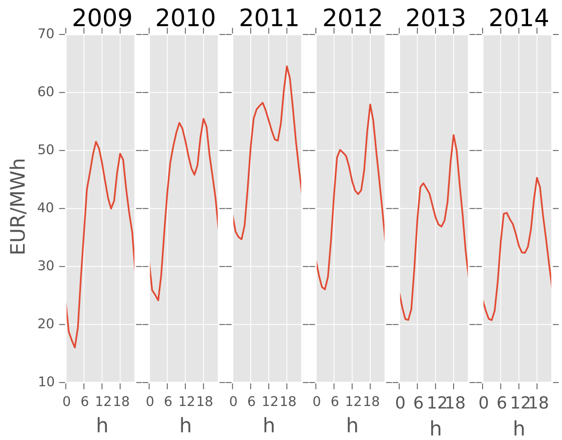


Figure 3.2: Mean hourly prices averaged yearly over years 2009-2014

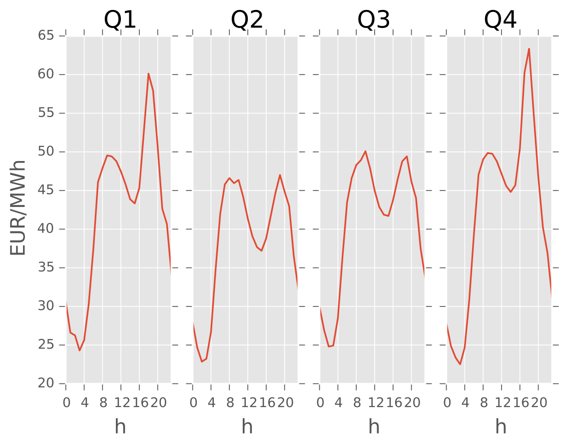


Figure 3.3: Mean hourly prices averaged quarterly over years 2009-2014

fourth quarter the daytime is significantly shorter than in second and third quarters, this results in higher second peak in the graph and also in not as deep local minimum around the noon. Both yearly and quarterly changes in hourly averages prices are also in Fig. 3.4, which shows the differences between different years and different quarters in more condensed format.

3.2 Daily price volatility of electricity

The volatility was calculated from the dataset that was described in previous section 3.1. The realized volatility was calculated for each day using the equation 2.1, however, several modification to the dataset were necessary — as the formula specified in equation 2.1 uses the difference of price transformed by logarithmic transformation and price of electricity

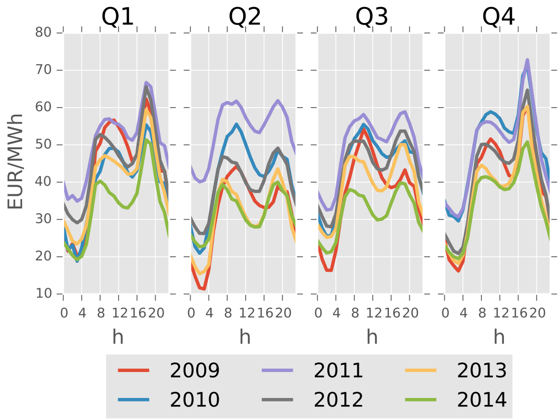


Figure 3.4: Mean hourly prices averaged quarterly and yearly over years 2009-2014

may be negative,thus the logarithm of negative price is undefined. Thus all negative or zero prices were first replaced by the lowest positive value that was present in the dataset (0.01 EUR/MWh). While this modification slightly distorts the dataset, the significance of this change is very small as the price is negative only for 142 hours out ouf the total 52584 hours in the dataset, which represents less than 0.3 % of the dataset.

3.2.1 Data description

The basic data description is provided in Table 3.2, however, it is obvious that while the price volatility is low (< 1) for most of the time as the 75 % quantile is around 0.95, there are periods where the volatility reaches extreme values — the mean is 5.41 and the standard deviation is slightly less than 20, moreover, the maximum value is about 242.

count	2191
mean	5.41
std	19.19
min	0.04
25%	0.26
50%	0.44
75%	0.95
max	242.31

Table 3.2: The basic description of used price volatility data

The occurrence of extreme spikes is also shown in Figure 3.5. The spikes were especially frequent in year 2009 which is one of the reasons why the year 2009 was used only for gaining the past values for the 1st January 2010 and not for the model estimation. Moreover, it would just bring noise into the data as it seems that the markets has changed significantly in 2010 and such spikes are not as frequent as they used to be.

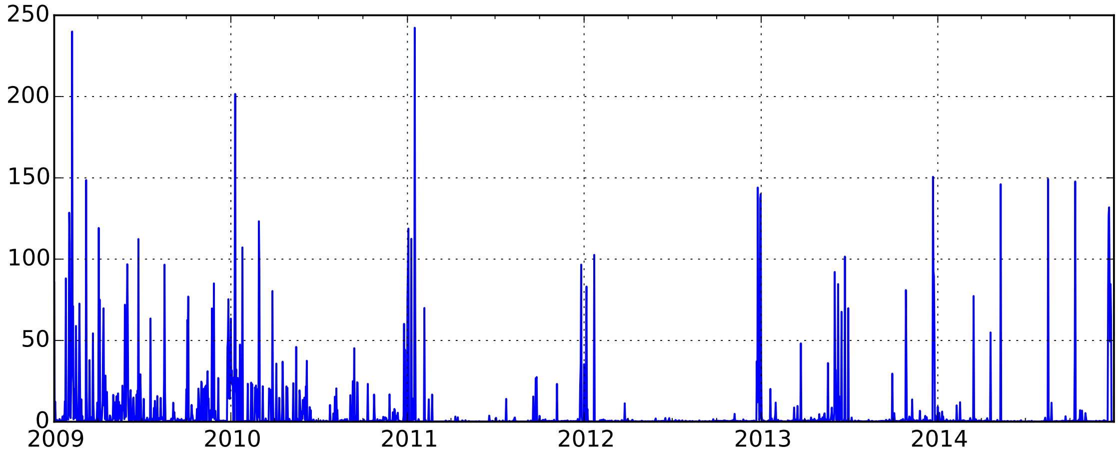


Figure 3.5: Plot of the price volatility dataset over years 2009-2014

3.2.2 Bounded data set

Several used models requires the dataset to be bounded to some interval, however, if the interval would be bounded by maximal value of 242, the most of the data would lie in only 1 % of that interval and that would cause inefficient use of the model. Moreover, the LSTM neural network model with sigmoid output layer require the data to be in the interval $[0, 1]$, therefore the scaling of the interval $[0, 242]$ would produce many numerical inaccuracies due to the computer representation required by some used libraries. From this reasons, there was created a second dataset with values bounded to a smaller interval $[0, q_{90}]$ where q_{90} is the 90 %quantile of the original data. Values higher than the quantile are replaced by the quantile, however, it is important that this is not an outlier removal procedures as there are no outliers in data because the data represents actual price of the electricity without any possible error in measurement — therefore the spikes are not replaced by some mean value but just bounded as the model are still required to try to model such spikes.

There is also another reason why the bounded dataset was produced — parameters for used model are found by minimizing the squared error, however, extreme spikes have also an extreme influence on the fitting procedure often resulting in high mean absolute error. Thus fitting some of the models on the bounded dataset results in a compromise between *mean absolute error* and *root mean squared error*. Another option would be to use different optimization techniques for fitting models but that is out of the scope of this work and might be included in future works.

The original dataset was bounded by the $q_{90} = 10.49$ and the basic summary of the bounded dataset is available in Table 3.3, the bounding influenced only the mean, standard deviation and the maximum value in the dataset summary, the spikes are still present in the dataset, but their influence was slightly limited. The bounded dataset of price volatility is also shown in Figure 3.6, where, in comparison with original dataset depicted in Figure 3.5, spikes are less dominant and non-extreme values of volatility are more significant.

count	2191
mean	1.76
std	3.17
min	0.04
25%	0.26
50%	0.44
75%	0.95
max	10.49

Table 3.3: The basic description of used price volatility data

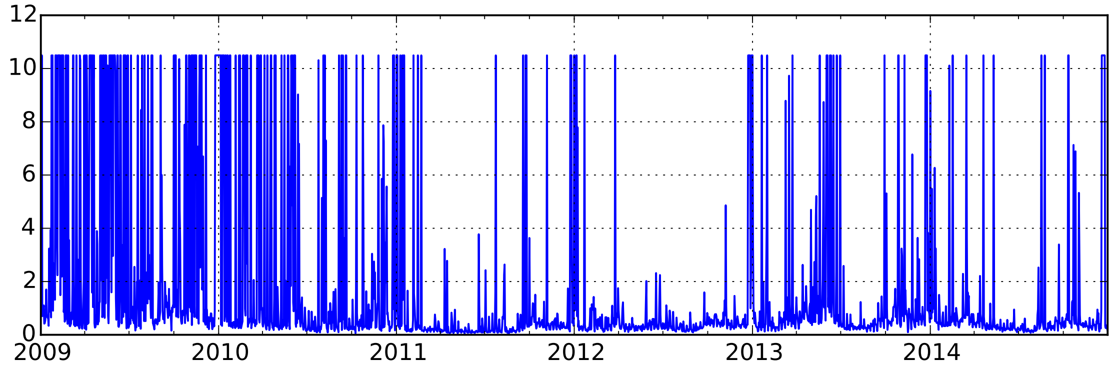


Figure 3.6: Plot of the upper bounded price volatility dataset over years 2009-2014

Chapter 4

Experiments and results

4.1 Implementation

Most of the work was implemented in Python 3, however, several parts used library forecast [19, 20] written in language R [28], which were accessed using the python library *rpy2*.Neural networks model were implemented using machine learning library PyBrain [29] because this library allows the use of recurrent neural networks with LSTM units. However, the PyBrain machine learning library is focused mostly on artificial neural networks, therefore the library StatsModels [30] was used for defining and fitting the linear autoregressive models using ordinary least squares method. Other used libraries included Pandas [25], Scipy [21], Matplotlib [18], Cython [8], IPython [27] and NumPy [32].

4.1.1 Purpose of experiments

The goal of experiments is to compare two distinct approaches for predicting the price volatility of electricity — computational intelligence method using artificial neural network models with LSTM units and more traditional autoregressive models. These two approaches were chosen from several reason, each of them is a representant of a common class used for predicting the volatility and price in general. The class of autoregressive models was chosen because it very simple model while it often provides very good performance [24]. The class of recurrent network models with LSTM units was chosen because such models are capable of modelling even complex models and they are able to hold stored information for an arbitrary length of time.

Experiments compares about 30 different configurations of LSTM models — three different approaches and 10 different network topologies. Also several different configuration of autoregressive models are compared — 8 basic autoregressive models are extended by 5 configuration of dummy variables and trained on two different datasets resulting in 98 different model configuration. Each of the models is evaluated from the perspective of root mean squared error (RMSE), mean absolute error (MAE) and also using the Diebold-Mariano test for both error measurement. However, the purpose of the experiments is not to find the best model for predicting the volatility as it would require comparison of at least several thousands different models — many of proposed models in literature are quite complex and their implementation and comparison is greatly out of the scope of this work.

4.2 Design of experiments

The design of experiments is rather simple for autoregressive models as they were fitted using deterministic methods and they are not influenced by random initialization as some other optimization methods - e.g. evolutionary optimization or gradient descent. However, that cannot be said for artificial neural network models with LSTM units as they are trained using the *resilient backpropagation* (Rprop). Moreover, neural network models in general are prone to overfitting, therefore some validation had to be included in the experimental design.

4.2.1 Training and testing data

As was mentioned above, neural network models are prone to overfitting, i.e., they work very well on the data they were fitted to but they do not work as well on previously unseen data. This is usually caused when the neural network model describes random error or noise instead of the underlying relationship. Other different models may overfit as well, but used autoregressive models are simple enough that it is unlikely they would suffer much from overfitting. In order to get true estimate of the models' performance, the dataset was split into two disjoint dataset — the training dataset (in-sample) and the test dataset (out-of-sample). All models were trained on the training dataset and then their performance was tested on the test dataset. However, these datasets do not form continuous blocks as there would be strong structural differences between those datasets caused by different changes in the data. Rarther, the original dataset was split into many smaller sets where 4 months training period is followed by 1 month test period as shown in Table 4.1

Training	**Test**
2010-01-01 – 2010-04-30	2010-05-01 – 2010-05-31
2010-06-01 – 2010-09-30	2010-10-01 – 2010-10-31
2010-11-01 – 2011-02-28	2011-03-01 – 2011-03-31
2011-04-01 – 2011-07-31	2011-08-01 – 2011-08-31
2011-09-01 – 2011-12-31	2012-01-01 – 2012-01-31
2012-02-01 – 2012-05-31	2012-06-01 – 2012-06-30
2012-07-01 – 2012-10-31	2012-11-01 – 2012-11-30
2012-12-01 – 2013-03-31	2013-04-01 – 2013-04-30
2013-05-01 – 2013-08-31	2013-09-01 – 2013-09-30
2013-10-01 – 2014-01-31	2014-02-01 – 2014-02-28
2014-03-01 – 2014-06-30	2014-07-01 – 2014-07-31
2014-08-01 – 2014-11-30	2014-12-01 – 2014-12-31

Table 4.1: The partitioning of original dataset into the training dataset and test dataset

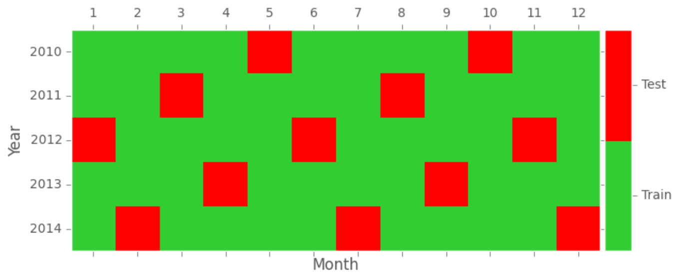


Figure 4.1: Visualization of the partitioning into the training and test dataset

While it might not be clear from the Table 4.1, each month in year is exactly once in the test dataset, which allows to evaluate the models without being biased towards particular time in year. This cannot be unfortunately said for individual years, as the years 2012 and 2014 have each 3 months in the test dataset and only 9 months in the training dataset while years 2010, 2011, and 2013 have each 2 months in the test dataset and 10 months in the training dataset. Still, the distribution of partitioning is very similar for each year and it is not expected to lead to significant biases. The partitioning is also shown in Figure 4.1

4.2.2 Repetitive training of RNN

The training of artificial recurrent neural networks strongly depends on initialization and its optimization may stuck in local minima, it is best practise to run the training cycle several times and then select the best performing model. For the purposes of this work, each LSTM model was trained in 6 times and its performance was evaluated on the training data, therefore the model selection is not biased on the test data. While it would be better to use some form of cross-validation for model selection, it would be computationally very expensive as the very complex neural network models with 512 LSTM units take more than 10 hours to train on the used computer.

Chapter 5

Evaluation of recurrent neural network models

In this chapter, the training results of three different classes of LSTM models described in Section 2.2 from Chapter 2 are provided. The main goal of this section is to evaluated the influence of the complexity — number of LSTM units — on the predictive performance of the model and on the training time as well. Comparison of between classes of LSTM models is provided in Chapter 6 together with comparison with autoregressive and GARCH models.

5.1 LSTM linear models

The artificial recurrent neural network models with LSTM units and linear output layer were the first LSTM models tested on the volatility data as they do not require any limitation of output value unlike the LSTM models with sigmoid output layer.

Figure 5.1 shows the perfomance of the neural networks model with LSTM units and linear output layer with respect to the number of LSTM units. While there is a small variation in the performance, the complexity of the network does not have significant influence on its predictive performance with when measuring the root mean squared error. However, slightly different situation is when comparing models with different complexities using the mean absolute error, where there is a slight progression — models with higher number of LSTM units tend to have lower MAE measure. However, the variance in performance across different initializations is relatively high and it cannot be said that more complex models are necessary better. The predictive performance with respect to the MAE measure is depicted in Figure 5.2.

Moreover, even if there is some slight progression in predictive performance with respect to the complexity of LSTM model with linear output layer, models with high complexity are computationally very costly as is shown in 5.3.

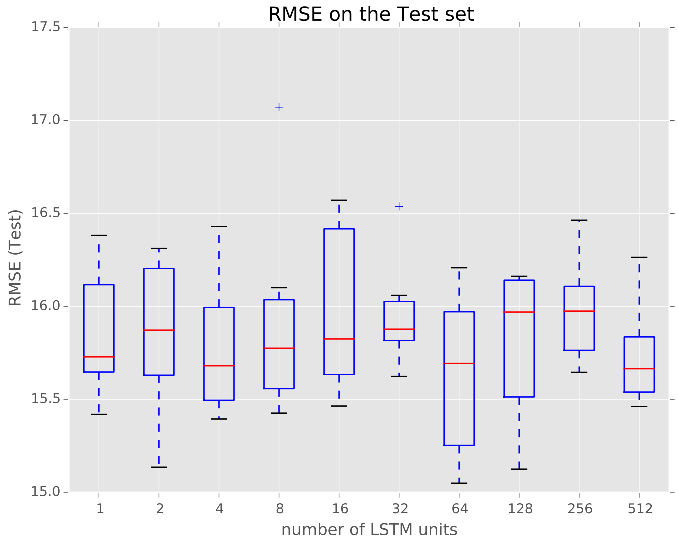


Figure 5.1: Performance of the LSTM linear models on the test set with respect to the RMSE and the number of LSTM units

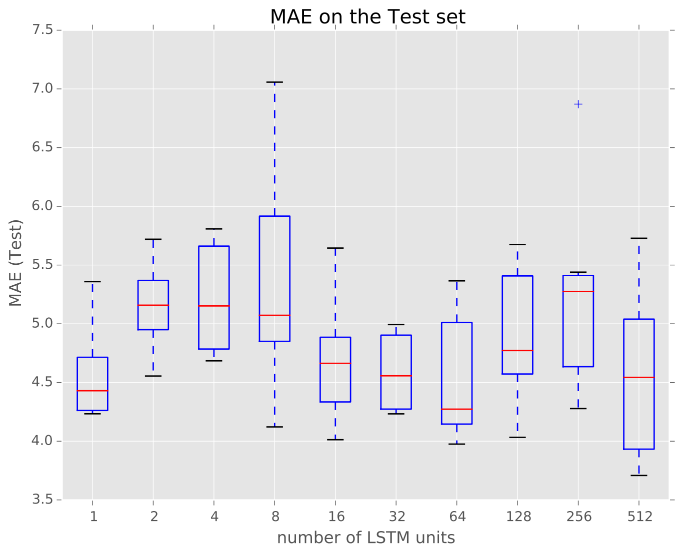


Figure 5.2: Performance of the LSTM linear models on the test set with respect to the MAE and the number of LSTM units

5.2 LSTM sigmoid models

The artificial recurrent neural network models with LSTM units and sigmoid output layer needs to trained on a bounded dataset, thus this class of models is evaluated on both the original dataset as it provides comparison with other models and on the restricted dataset

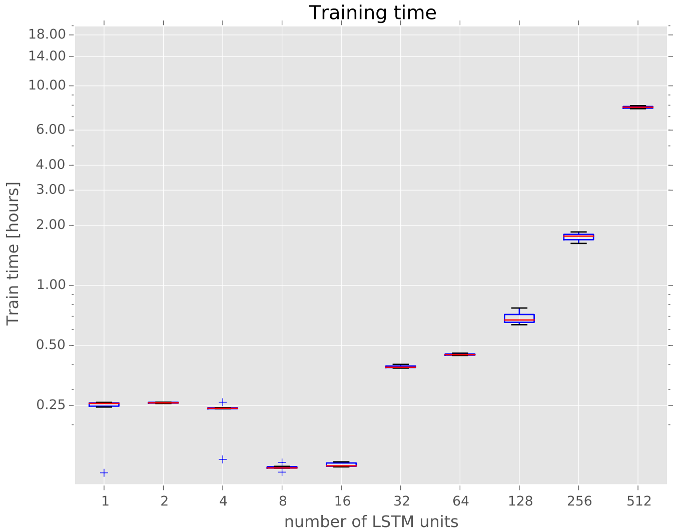


Figure 5.3: Training time of LSTM linear models with respect to the number of LSTM units

as the model optimization procedure was run with respect to this dataset.

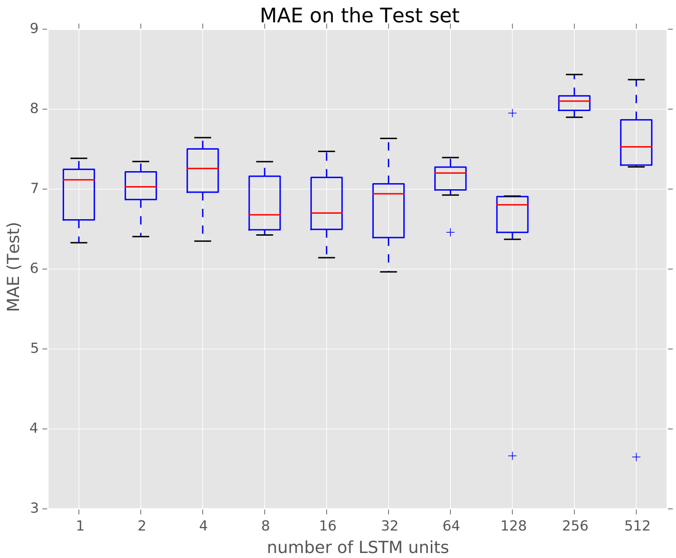


Figure 5.4: Performance of the LSTM linear models on the test set with respect to the MAE and the number of LSTM units

As is shown in Figure 5.4 and Figure 5.5, the performance of the LSTM models is similar on original dataset and bounded dataset, it seems that the MAE measures differs only by a constant. Moreover, it does not seem that there is a significant progression with

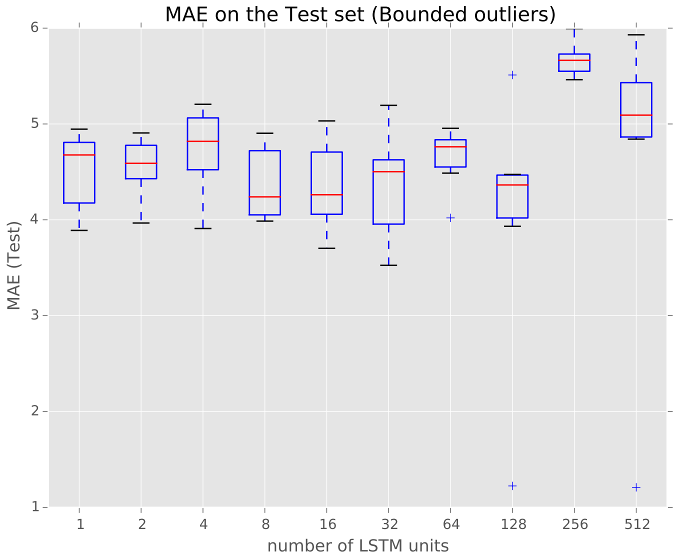


Figure 5.5: Performance of the LSTM linear models on the test set with respect to the MAE and the number of LSTM units

increasing number of LSTM units with the exception for LSTM model with 128 and 512 LSTM units, where the model achieved remarkably low MAE for one random intitialization — this further confirms the need for more random initializations as those models performed well also on the training set as shown in Figures 5.6 and 5.7.

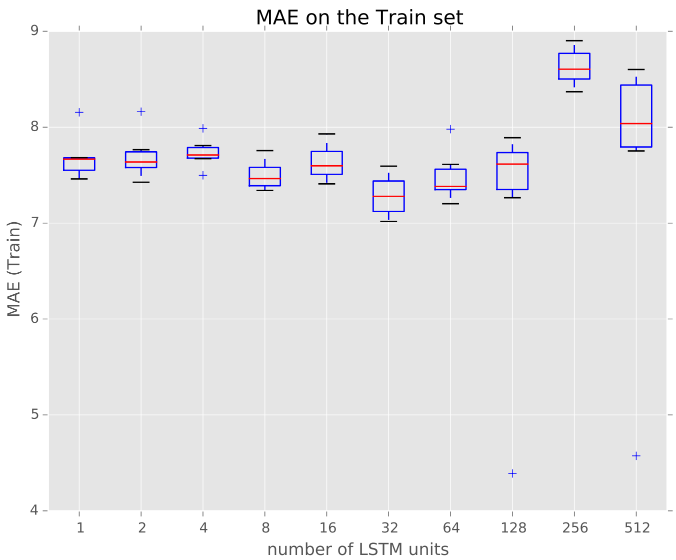


Figure 5.6: Performance of the LSTM linear models on the training set with respect to the MAE and the number of LSTM units

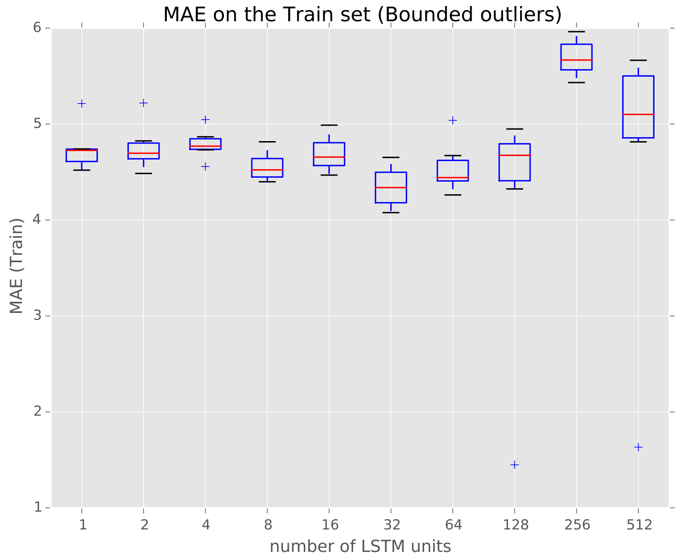


Figure 5.7: Performance of the LSTM linear models on the training set with respect to the MAE and the number of LSTM units

However, the MAE measure is not as informative as the RMSE measure because these models were optimized with respect to the RMSE measure. The training RMSE is shown in Figures 5.8 and 5.9, while the differences in MAE measures between original and bounded datasets were minimal, that cannot be said for the differences in RMSE measures. The model show minimal variance in the RMSE on the bounded training set with exception to two initialization with 128 and 512 LSTM units — actually, the differences in RMSE on the bounded dataset are extremely similar to the differences in MAE on the bouned datasets for both training and test datasets. However, when comparing the RMSE on the bounded and original datasets, the differences are significant — see Figures 5.8 and 5.9 and Figures 5.10 and 5.11. The RMSE on the original dataset shows slightly higher variance and even different distribution of minimal and maximal values.

The training time of the LSTM model with sigmoid output layer is shown in Figure 5.12, while increasing complexity has shown to be not as useful in the LSTM model with linear output layer, here the LSTM sigmoid model schieved several good results for models with higher complexity - namely with 128 and 512 LSTM units. It seems that the LSTM sigmoid model is significantly dependent on the random initialization and it would be useful to try more than 6 repetition for more complex models. However, as depicted in Figure 5.12, the training of more complex models is very costly.

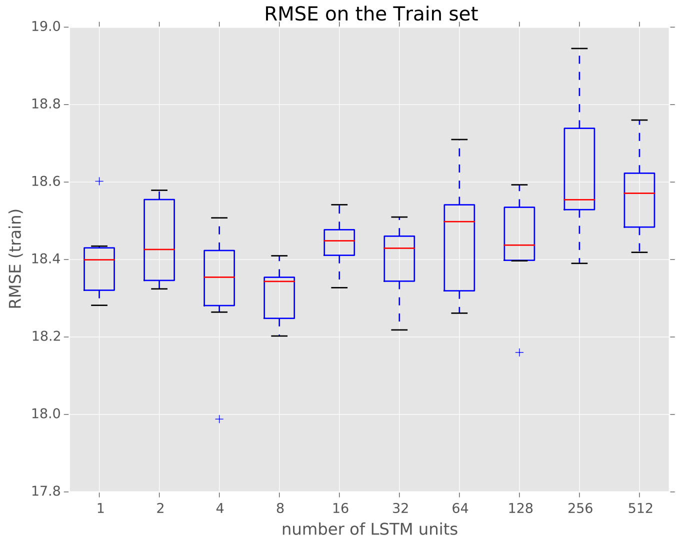


Figure 5.8: Performance of the LSTM linear models on the training set with respect to the RMSE and the number of LSTM units

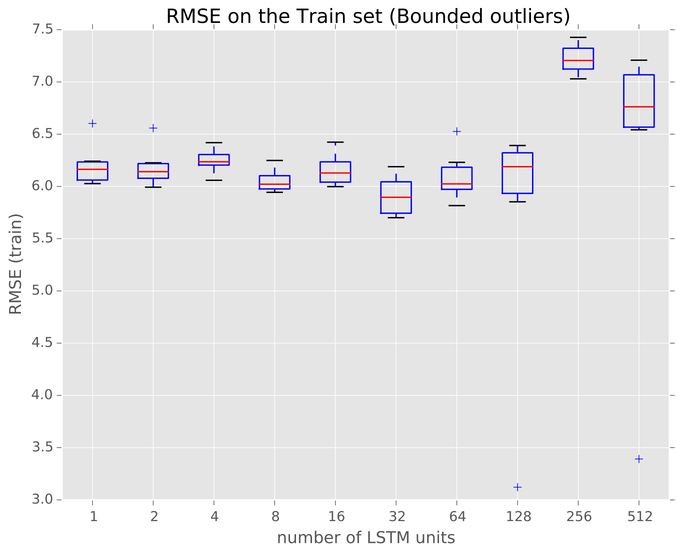


Figure 5.9: Performance of the LSTM linear models on the training set with respect to the RMSE and the number of LSTM units

5.3 LSTM linear models for day-ahead hourly price prediction

This class of LSTM linear models is different from models described above as it works in two stages — first, it predicts the day-ahead hourly prices, second, it computes realized

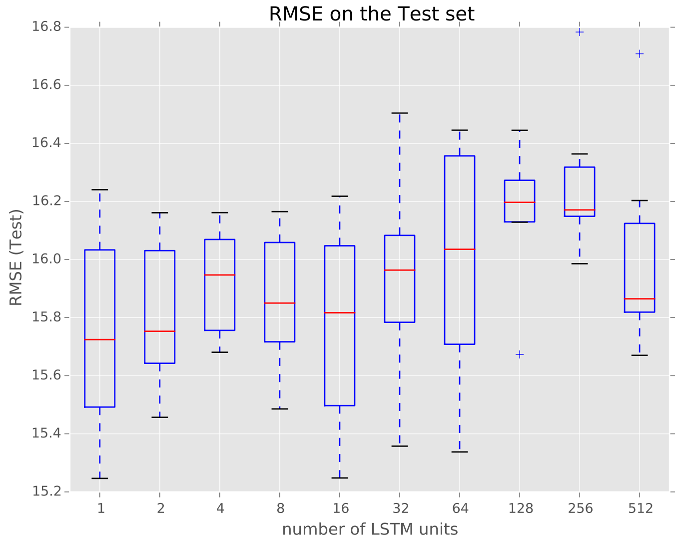


Figure 5.10: Performance of the LSTM linear models on the test set with respect to the RMSE and the number of LSTM units

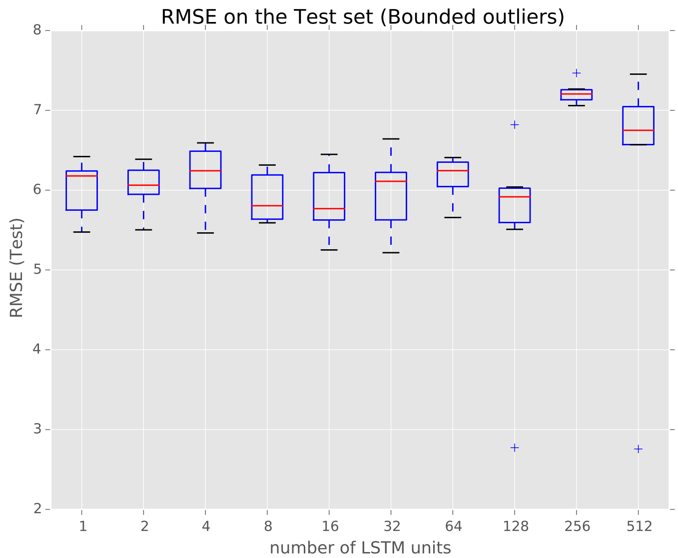


Figure 5.11: Performance of the LSTM linear models on the test set with respect to the RMSE and the number of LSTM units

volatility from the prediction. Thus the experimental results are discussed from two different points of view, the predictive performance of the price prediction and the predictive performance of the volatility prediction.

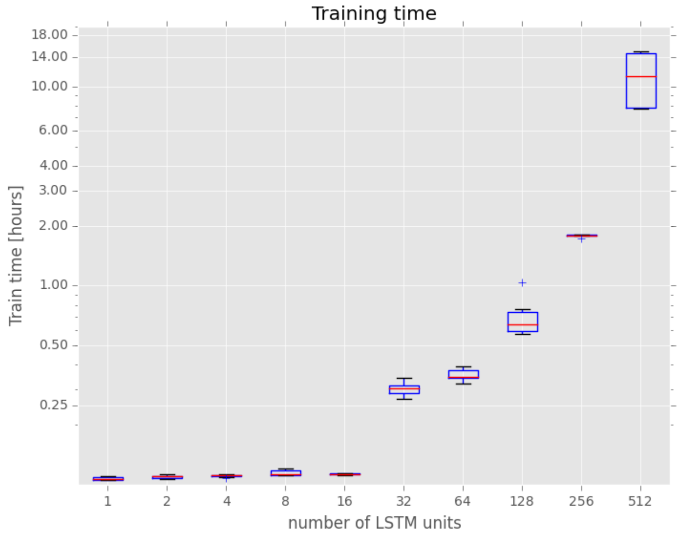


Figure 5.12: Training time of LSTM sigmoid models with respect to the number of LSTM units

5.3.1 Day-ahead price prediction

Day-ahead price prediction is a very complex problem as the electricity price shows many different seasonalities and periodicities, moreover, it is dependent on many external variables, e.g., weather, temperature, exchange rate, and price of generation — which itself depends on types of used power-plants, the price of gas and coal, etc. However, most of the mentioned data is very difficult to obtain, therefore the key assumption of this model is that these variables do not change much on day-to-day basis and thus using past observations allows the model to predict the future day-ahead hourly prices.

The overall RMSE on the test dataset for different topologies is shown in Figure 5.13 — even though the minimum RMSE is slowly decreasing for increasing number of LSTM units, the variance of RMSE is increasing as well — the extreme case is for LSTM with 256 units, where the training failed in one repetition and caused extremely high RMSE error on both test and training set. As shown in Figure 5.14, the RMSE on training dataset is very similar to the RMSE on the test dataset — with a small exception of recurrent neural network model with 512 LSTM units, where the performance on the test dataset is slightly worse.

While it is not perfectly visible in Figures 5.13 and 5.14, the model with 512 LSTM units has generally overfitted to the data — as it is shown in Figures 5.15 and 5.16 showing minimal RMSE per repetition for different topologies of the neural network model with LSTM units and linear output layer. The RMSE is generally decreasing with increasing number of LSTM units on the training set, however, the RMSE is decreasing only up to the model with 256 LSTM units on the test set, while the model with 512 LSTM units has much greater RMSE on the test set, which was probably caused by the fact, that the

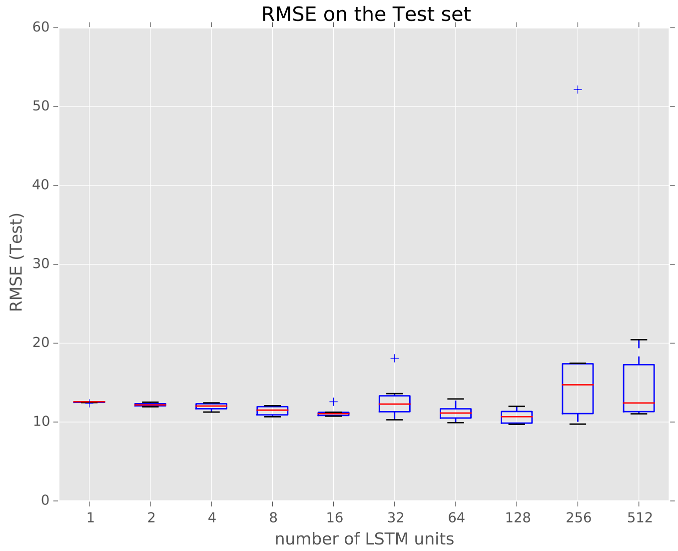


Figure 5.13: Performance of the LSTM linear models for day-ahead price prediction on the test dataset (RMSE)

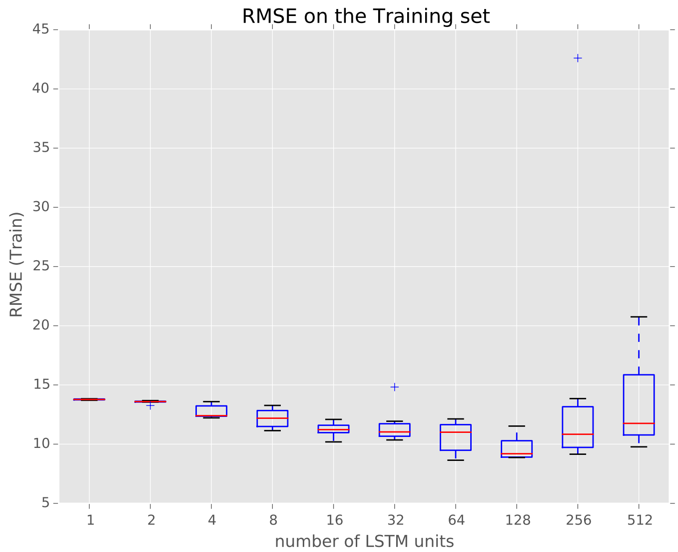


Figure 5.14: Performance of the LSTM linear models for day-ahead price prediction on the training dataset (RMSE)

model with 512 units has overfitted to the training data — it seems that the artificial neural network model has difficulties with generalization and proper training when higher number of LSTM units is present - the LSTM model with 512 LSTM units has relatively high train error and the test error is even higher than would be expected with such train error. The high train error in that case is caused by sub-optimally trained neural network while the test error is caused partly by the same reason and partly by overfitting to the data.

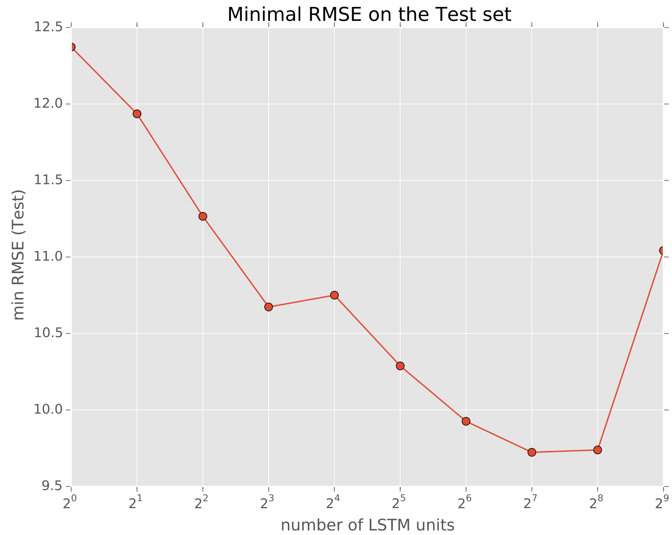


Figure 5.15: Minimal RMSE per repetition for different topologies of LSTM model on the test set

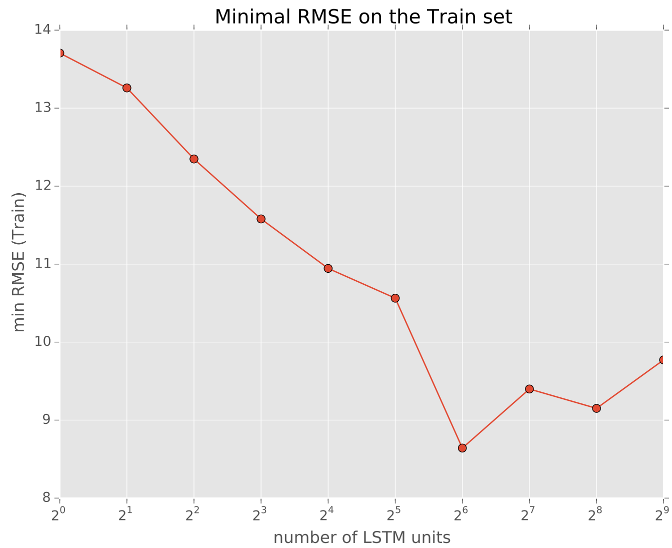


Figure 5.16: Minimal RMSE per repetition for different topologies of LSTM mode on the training set

The training time is shown in Figure 5.17, the very complex neural network model has an extreme training time, which in connection with its poorer performance in comparison with other topologies, makes this model

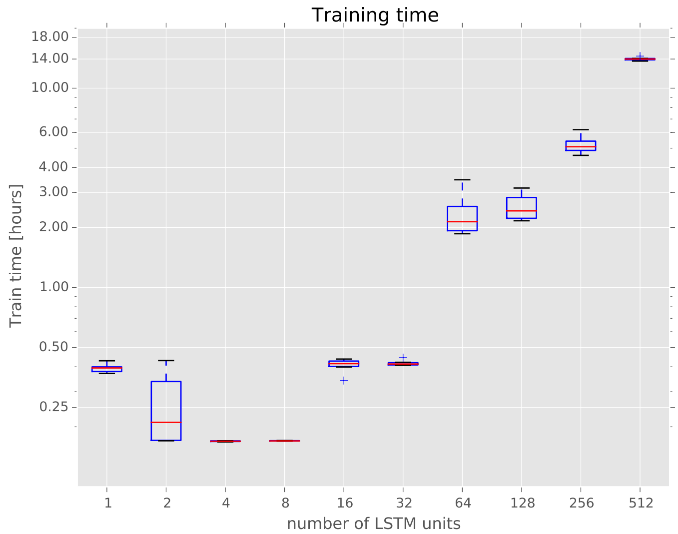


Figure 5.17: Computation costs of LSTM linear model for day-ahead price prediction

5.3.1.1 1 LSTM unit

It appears that the artificial neural network model (ANN) with this topology approximates the mean price for each hour, however it fails to predict weekly and longer periodicities ind the data. The prediction results is shown in Figure B.1 with more details available in Figure B.2.

5.3.1.2 2 LSTM units

This neural network seems to approximate the series with hourly averages as well but it is also able to slightly modify its outputs and does not output only the average as it is able to scale the 'average day' up and down. However, as shown in Figure B.3 and B.4 it is far from perfect — the scaling is usually too subtle.

5.3.1.3 4 LSTM units

The neural network with 8 LSTM units still the 'scaled average day' as the networks with previous topologies, however, it seems to remember another average for high peek days, where the price is much higher in the afternoon and evening than in the regular days. Moreover, it also remembers another output for days with lower prices. Despite these advances, it still fails to cope with the weekly periodicity in the price variance — it predict well the fairly stable low price in the day but it fail to correctly predict the high price as t is much more volatile. Example output can be seen in Figures B.5 and B.6.

5.3.1.4 8 LSTM units

The neural network model with this topology also uses several daily patterns as the previous topologies, however, its range of patterns is much higher and it can also correctly predict good patterns even on the test data - as shown in Figures B.7 and B.8.

5.3.1.5 16 LSTM units

The ANN with 16 LSTM units also mostly uses daily patterns but its pattern range is more flexible than of the previous networks and moreover, it seems as it is able to modify the patterns when needed. An example is shown in Figure B.9.

5.3.1.6 32 LSTM units

In case of a network with this topology, it is not possible anymore to talk about daily patterns as it is able to produce many different outputs while maintaining bot daily and weekly periodicities. Moreover, it is able to approximate the price series nicely with smooth curves — compared to the network with, for example, 512 LSTM units, details are in Figures B.10 and B.11.

5.3.1.7 64 and 128 LSTM units

The neural network models with 64 and 128 LSTM units performed very similarly and, more importantly, very well - the ANN with 64 neural units had the smallest training error while maintaining low test error and the ANN with 128 had the best test score from tested ANNs. Example outputs are available in Figures B.12 and B.13 for an ANN with 64 LSTM units and in Figure B.14 for an ANN with 128 LSTM units.

5.3.1.8 256 and 512 LSTM units

Networks with such topologies are very complex and they showed much bigger variation in both train and test errors, moreover, they took much time to be trained — around 5 hours for the ANN with 256 units and around 14 hours for the ANN with 512 LSTM units. Moreover, their output was much less smooth than the output of less complex networks — the often oscillated around the actual value. In one case, the training of the ANN with

256 LSTM units failed completely, as shown in Figure B.16. Other example outputs are in Figures B.15 and B.17.

5.3.1.9 Results summary

Used neural network models were usually more accurate with increasing number of LSTM units up to the number of 128 LSTM units, then ANNs became a bit less accurate with more complex topologies as they were harder to train and they also overfitted the data. However, experiments for each topology were repeated only 6 times as the training was computationally very costly — all experiments took roughly 156 hours of computations.

5.3.2 Volatility prediction

The realized volatility was computed from the day-ahead price prediction described in previous section. In contrast to previous models LSTM linear and LSTM sigmoid, this model had a slightly different optimization criteria — it optimized the RMSE of the day-ahead price prediction and not the volatility itself, which is calculated as sum of squared differences of logarithmic prices, therefore different behavior was expected and observed. Moreover, days on which volatility spikes has occurred, have usually much less influence here because this model does not take into account the order of hourly prices in the day while the calculation of the realized volatility does. Therefore if there are small fluctuations in the price on the hour-to-hour basis, the realized volatility is expected to be high, however, the RMSE error of price prediction will be similar as if, for example, there were no fluctuation in the price but the prediction would be biased toward higher values — which would produce bigger error in volatility prediction because the volatility without the fluctuations would be relatively low.

As is shown in Figures 5.18 and 5.19, there is no bigger difference in distribution between the RMSE and MAE on the test datasets. The artificial neural network models with LSTM units have fairly low error on the test set even for small networks up to 8 LSTM units, more complex networks behave similarly, however, they show higher variation in the predictive performance — the neural network models with 256 and 512 LSTM units have a very high error for most of the initialization, which is caused by the tendency of such networks to oscillate around the target prices, which do not lead to much higher error for price prediction, but it leads to an extremely high realized volatility. The performance on the training data is very similar to the performance on the test data except for the neural network model with only 1 LSTM unit, where the error is much higher — both MAE and RMSE as is shown in Figures 5.20 and 5.21. This may be caused by the higher occurrence of spikes at the beginning of year 2010 and in January 2011 which are in train data, while the network with only 1 LSTM unit is not flexible enough to different seasonalities in the data, viz Figure B.1.

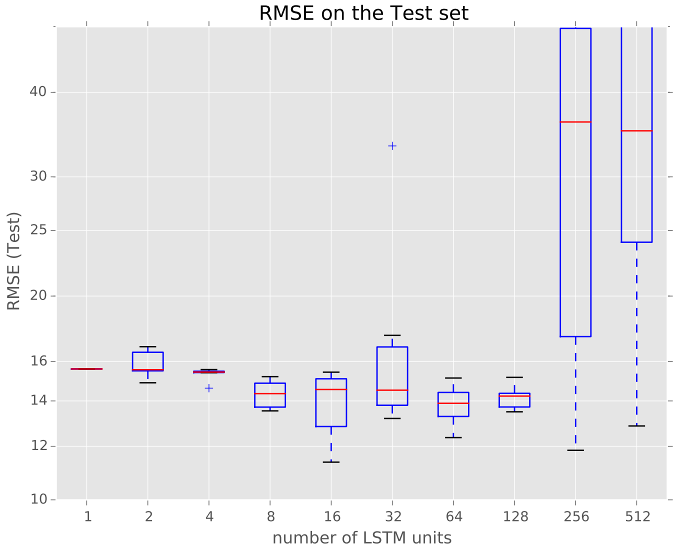


Figure 5.18: Evaluation of volatility prediction of the LSTM linear models for the price prediction on the test set with respect to the RMSE and the number of LSTM units

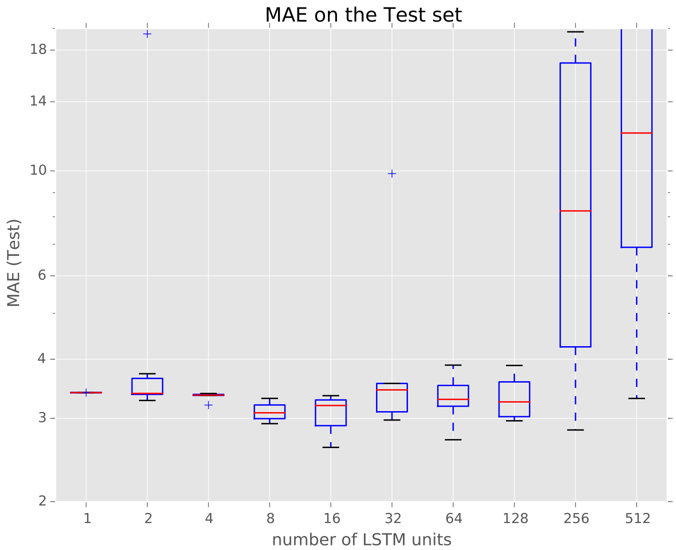


Figure 5.19: Evaluation of volatility prediction of the LSTM linear models for the price prediction on the test set with respect to the MAE and the number of LSTM units

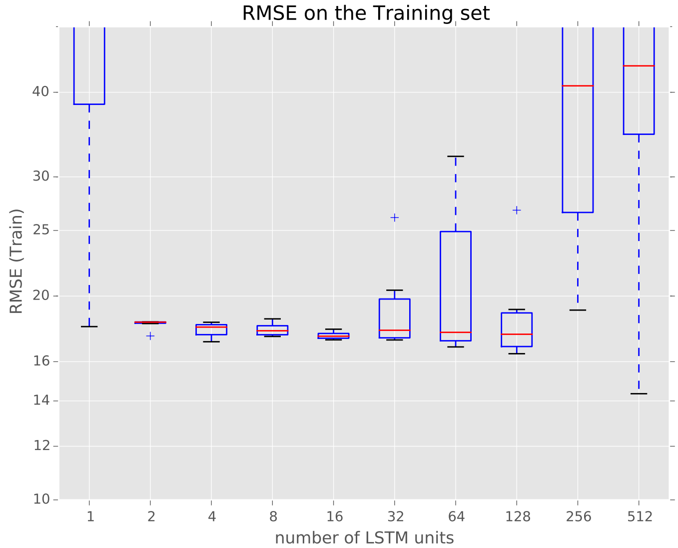


Figure 5.20: Evaluation of volatility prediction of the LSTM linear models for the price prediction on the training set with respect to the RMSE and the number of LSTM units

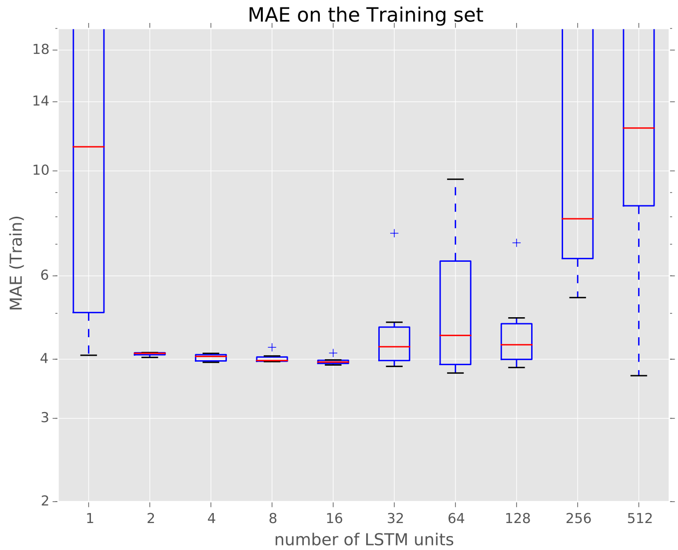


Figure 5.21: Evaluation of volatility prediction of the LSTM linear models for the price prediction on the training set with respect to the MAE and the number of LSTM units

Chapter 6

Model comparison

While the previous chapter compares models only within one class, the goal of this chapter is to provide final comparison of used models. Each model is evaluated in six different categories:

Root mean squared error (RMSE) is one of the most often used error measures as it has natural interpretation — it represents the average Euclidean distance (L_2 distance) of the prediction and true values. Since the euclidean distance represents the straight-line distance in Euclidean space, this measure is often used in Robotics. Moreover, another reason for the popularity of RMSE is that it is easy to take derivative of it, which is often desirable in many optimization procedures.

Mean absolute error (MAE) is also one of the most often used error measure, it is very simple measure and it represents the average Manhattan distance (L_1 distance). While it is less desirable for some optimization techniques because of its derivative, it is often more useful than RMSE optimization as it is more robust to outliers.

Diebold-Mariano test $p = 1$ score (DM1) The Diebold-Mariano test is often used for pairwise comparison of two different forecasts (models), when testing whether two forecast are the same or whether one is superior to the other. The DM1 represents a number of models which are inferior to the evaluated model using Diebold-Mariano test based on absolute error ($p = 1$) on the 5 % significance level. The score is computed over all possible pairings of the model with other models.

Diebold-Mariano test $p = 2$ score (DM2) This score very similar to the DM1 with the exception that the Diebold-Mariano test is now based on the squared error ($p = 2$).

Pareto rank Both RMSE and MAE might be important error measures, however, decreasing RMSE often results in increase in MAE, therefore models that differs in both RMSE and MAE cannot be really compared in terms of efficiency. The Pareto rank is calculated iteratively in following way — first, for the set $\mathbb{S}_0$ of all models, the Pareto optimal set $\mathbb{P}_0$ (frontier) is determined and all models from the set $\mathbb{P}_0$ are assigned rank 0 and are removed from the set $\mathbb{S}_1 = \mathbb{S}_0 \setminus \mathbb{P}_0$, then another iteration is calculated.

Thus the Pareto rank compares the models both with respect to the RMSE and MAE, if there is a model with Pareto rank k, then there are k other models, that that had better performance in at least one of the error measures. However, the Pareto rank does not compare the size of the improvement between models.

These several evaluation measures compares the models from slightly different angles, therefore they provide much better picture as a set of measures than would provide comparison based just on RMSE or MAE. However, there is be single best model because it is unlikely that a model would be best in all these criteria.

6.1 Good models

As the total number of compared models is too high to provide results for all of these models in nicely readable format, only a subset of 'good' models will be discussed in this section. Results for all models are available in Appendix D. The model was selected to be in the *subset of good models* if the model was 15th or better in any criteria described above. The *subset of good models* based on the criteria stated above contains 34 different models — the actual performance of selected models is shown in Table 6.1, models are sorted first by their Pareto rank and then by error measures. The overall model performance is shown in Table 6.2, where are ranks of the models for individual criteria. The Pareto rank is provides a bit distorted results when there is a small cluster of similar points as each of them is in different Pareto frontier, however, it provides very good comparison of the performance of models if they are more or less uniformly distributed — first 10 Pareto frontiers of RMSE and MAE of used models are shown in Figure 6.1.

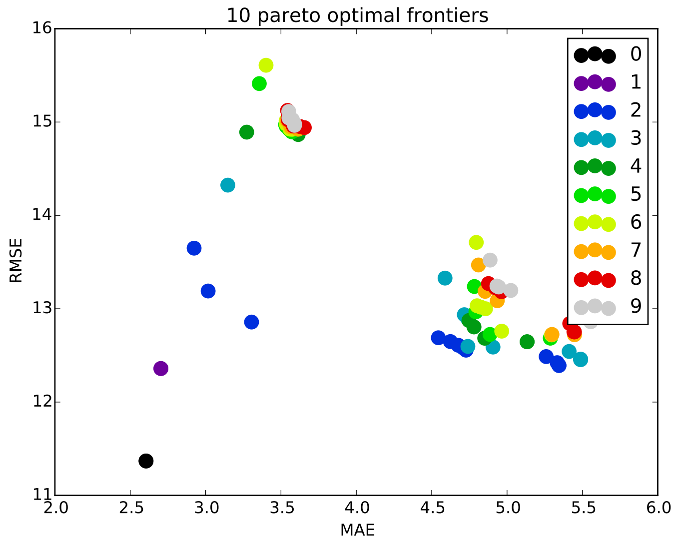


Figure 6.1: Scatter plot of first 10 Pareto frontiers

Model name	RMSE	MAE	DM1	DM2	Pareto
LSTM_lin_fpr_16_r_0	11.37	2.60	131.0	8.0	0
LSTM_lin_fpr_64_r_3	12.36	2.70	140.0	84.0	1
LSTM_lin_fpr_8_r_3	13.65	2.92	138.0	79.0	2
HAR	12.61	4.68	51.0	93.0	2
HAR w	12.56	4.73	42.0	96.0	2
HAR y	12.65	4.62	55.0	93.0	2
LSTM_lin_fpr_512_r_3	12.86	3.30	91.0	32.0	2
x1 d	12.58	4.71	41.0	83.0	2
x1,x6,x13,x27 dm	12.49	5.26	25.0	94.0	2
HAR dm	12.42	5.33	18.0	93.0	2
x1 dm	12.39	5.34	19.0	92.0	2
LSTM_lin_fpr_32_r_3	13.19	3.02	91.0	3.0	2
LSTM_lin_fpr_128_r_1	14.32	3.15	137.0	68.0	3
HAR d	12.60	4.74	41.0	85.0	3
x1 w	12.59	4.91	37.0	89.0	3
HAR dmy	12.45	5.49	13.0	97.0	3
x1,x6,x13,x27 dmy	12.54	5.41	16.0	86.0	3
x1 dmy	12.46	5.49	15.0	86.0	3
LSTM_lin_fpr_2_r_2	14.89	3.27	137.0	30.0	4
x1,x6,x13,x27 y	12.80	4.78	45.0	91.0	4
x1,x6,x13,x27 w	12.68	4.85	41.0	89.0	4
x1	12.65	5.13	19.0	94.0	4
x1-x7 or dm	14.92	3.55	103.0	56.0	5
x1,x6,x13,x27 or d	14.96	3.53	108.0	45.0	5
HAR or d	14.94	3.55	104.0	47.0	5
x1-x7 or d	14.97	3.53	110.0	41.0	5
x1,x6,x13,x27	12.72	4.89	41.0	93.0	5
LSTM_lin_fpr_4_r_0	15.41	3.36	135.0	12.0	5
x1,x2,x7 or d	14.98	3.54	107.0	38.0	6
x1,x7,x14,x28 or d	15.01	3.53	107.0	35.0	6
x1 y	12.76	4.96	33.0	91.0	6
LSTM_lin_fpr_1_r_5	15.61	3.40	135.0	9.0	6
x1-x7 or w	15.01	3.54	104.0	37.0	7
x1,x7,x14,x28 or y	15.12	3.54	101.0	26.0	8

Table 6.1: Evaluation of selected models by actual values — model was selected if it had at least one rank less or equal 15.

Model name	RMSE	MAE	DM1	DM2	Pareto	Average
LSTM_lin_fpr_16_r_0	1.0	1.0	7.0	121.5	0	32.625
LSTM_lin_fpr_64_r_3	2.0	2.0	1.0	17.0	1	5.5
LSTM_lin_fpr_8_r_3	50.0	3.0	2.0	20.0	2	18.75
HAR	13.0	69.0	68.0	6.5	2	39.125
HAR w	9.0	72.0	73.5	2.0	2	39.125
HAR y	15.0	68.0	67.0	6.5	2	39.125
LSTM_lin_fpr_512_r_3	28.0	7.0	50.5	77.5	2	40.75
x1 d	10.0	70.0	78.5	18.0	2	44.125
x1,x6,x13,x27 dm	7.0	103.0	98.0	3.5	2	52.875
HAR dm	4.0	107.0	109.0	6.5	2	56.625
x1 dm	3.0	108.0	106.5	9.0	2	56.625
LSTM_lin_fpr_32_r_3	40.0	4.0	50.5	133.0	2	56.875
LSTM_lin_fpr_128_r_1	57.0	5.0	3.5	23.0	3	22.125
HAR d	12.0	74.0	78.5	16.0	3	45.125
x1 w	11.0	89.0	88.0	12.5	3	50.125
HAR dmy	5.0	118.0	122.0	1.0	3	61.5
x1,x6,x13,x27 dmy	8.0	110.0	113.5	14.5	3	61.5
x1 dmy	6.0	117.0	117.0	14.5	3	63.625
LSTM_lin_fpr_2_r_2	64.0	6.0	3.5	84.0	4	39.375
x1,x6,x13,x27 y	26.0	76.0	71.0	10.5	4	45.875
x1,x6,x13,x27 w	16.0	83.0	78.5	12.5	4	47.5
x1	14.0	101.0	106.5	3.5	4	56.25
x1-x7 or dm	69.0	23.0	14.0	33.0	5	34.75
x1,x6,x13,x27 or d	77.0	12.0	9.0	46.0	5	36.0
HAR or d	73.0	18.0	12.5	41.5	5	36.25
x1-x7 or d	82.0	10.0	8.0	52.5	5	38.125
x1,x6,x13,x27	21.0	87.0	78.5	6.5	5	48.25
LSTM_lin_fpr_4_r_0	122.0	8.0	5.5	111.0	5	61.625
x1,x2,x7 or d	83.0	13.0	10.5	56.0	6	40.625
x1,x7,x14,x28 or d	87.0	11.0	10.5	63.0	6	42.875
x1 y	25.0	96.0	92.0	10.5	6	55.875
LSTM_lin_fpr_1_r_5	128.0	9.0	5.5	117.5	6	65.0
x1-x7 or w	89.0	14.0	12.5	58.0	7	43.375
x1,x7,x14,x28 or y	107.0	15.0	17.5	93.0	8	58.125

Table 6.2: Evaluation of selected models by ranks — model was selected if it had at least one rank less or equal 15.

6.2 Short summary of results for selected models

6.2.1 Main criterion: Pareto rank

From many evaluated models, the LSTM linear model with 16 LSTM units and two stage prediction — first price prediction and second volatility computation — proven to be the best both from perspective of RMSE and MAE. However, this model is closely followed by models from the same class but only with different number of LSTM units — the 2nd and 3rd position have models with 64 and 8 LSTM units. Other topologies were slightly worse but all of them were selected in the subset of models and all of them was in 6th Pareto frontier or better. However, this class of recurrent artificial neural network models with LSTM units was the only class that made it into the selection of *good* models, other classes provided much worse predictions.

The autoregressive models had several successful models — a very good performance had the HAR model with dummy variable for weekends and only slightly worse performance had HAR models without any dummy variables and with the year dummy variable. Surprisingly, even the simplest AR(1) model with had a very good performance, which got slightly better after addition of several dummy variables (*d*, *dm*, *dmy*).

6.2.2 Main criterion: RMSE

When the models are evaluated mainly from the perspective of the root mean squared error (RMSE), the best model is neural network model with 16 LSTM units that firstly predicts the day-ahead hourly prices and then computes volatility (LSTM_lin_fpr_16_r_0), this model is the best in terms of MAE, however, its Diebold-Mariano $p = 2$ test score is very low — this model better only than 8 models from all tested models when using Diebold-Mariano test with = 2 on 5 % level of significance, which is very unexpected as other models in the best 15 when selected by RMSE have the DM2 score at least 80, moreover, the similar model LSTM_lin_fpr_64_r_3, which is 2nd best model by RMSE, have the DM2 score equal to 84. However, this model is the only one that achieved RMSE lower thank 12 and there is relatively huge difference between this model and 2nd model — the difference is almost equal 1 — others 9 models are very similar in terms of RMSE and the differences between consecutive models is less than 0.05. The 15 best models in terms of RMSE are shown in Table 6.3.

Model name	RMSE	MAE	DM1	DM2	Pareto
LSTM_lin_fpr_16_r_0	**11.37**	2.60	131	8	0
LSTM_lin_fpr_64_r_3	**12.36**	2.70	140	84	1
x1 dm	**12.39**	5.34	19	92	2
HAR dm	**12.42**	5.33	18	93	2
HAR dmy	**12.45**	5.49	13	97	3
x1 dmy	**12.46**	5.49	15	86	3
x1,x6,x13,x27 dm	**12.49**	5.26	25	94	2
x1,x6,x13,x27 dmy	**12.54**	5.41	16	86	3
HAR w	**12.56**	4.73	42	96	2
x1 d	**12.58**	4.71	41	83	2
x1 w	**12.59**	4.91	37	89	3
HAR d	**12.60**	4.74	41	85	3
HAR	**12.61**	4.68	51	93	2
x1	**12.65**	5.13	19	94	4
HAR y	**12.65**	4.62	55	93	2

Table 6.3: 15 best model when evaluated by RMSE

6.2.3 Main criterion: MAE

Even though all models were optimalized by procedures optimizing the RMSE, it is also worthy to compare models in terms of MAE — the MAE measure is less sensitive to outliers than RMSE. The 10 best models by MAE are shown in Table 6.4. The best model is again LSTM_lin_fpr_16_r_0, however, this list of models is dominated by all models from the same class — two stage model based on artificial neural network model with LSTM units as 9 models from this class (out of 10) are on the first 9 positions. The only model from this class that is not in the list of 10 best models by MAE is the LSTM_lin_fpr_256, which is probably caused by selection of overfitted model as the models were selected based on the training data performance. The minimum test performance of LSTM_lin_fpr_256 reaches RMSE lower than 12, however, the the RMSE variance for this class of models is very high (viz Figure 5.18) and it is necessary to select these models carefully.

Other models that achieved good performance with respect to the MAE measure were autoregressive models — those that were trained on dataset with bounded extreme values, which is expected — lowering the MAE measure was the reason behind the use of the bounded dataset. However, these models still optimized the RMSE measure, just on the bounded dataset, therefore performance evaluation by MAE measure is less informative than if these models were optimized using different optimization techniques. All 6 autoregressive models from the selection has the MAE measure in interval $[3.53, 3.54]$, moreover, in the best selection of best 50 models, the difference between consecutive autoregressive models is 0.01, the spread is very small. From these two reasons, the MAE measure does not provide much information at this level.

Model name	**RMSE**	**MAE**	**DM1**	**DM2**	**Pareto**
LSTM_lin_fpr_16_r_0	11.37	**2.60**	131	8	0
LSTM_lin_fpr_64_r_3	12.36	**2.70**	140	84	1
LSTM_lin_fpr_8_r_3	13.65	**2.92**	138	79	2
LSTM_lin_fpr_32_r_3	13.19	**3.02**	91	3	2
LSTM_lin_fpr_128_r_1	14.32	**3.15**	137	68	3
LSTM_lin_fpr_2_r_2	14.89	**3.27**	137	30	4
LSTM_lin_fpr_512_r_3	12.86	**3.30**	91	32	2
LSTM_lin_fpr_4_r_0	15.41	**3.36**	135	12	5
LSTM_lin_fpr_1_r_5	15.61	**3.40**	135	9	6
x1-x7 or d	14.97	**3.53**	110	41	5
x1,x7,x14,x28 or d	15.01	**3.53**	107	35	6
x1,x6,x13,x27 or d	14.96	**3.53**	108	45	5
x1,x2,x7 or d	14.98	**3.54**	107	38	6
x1-x7 or w	15.01	**3.54**	104	37	7
x1,x7,x14,x28 or y	15.12	**3.54**	101	26	8

Table 6.4: 15 best model when evaluated by MAE

6.2.4 Main criterion: DM1

This is another measure based on the absolute error, therefore the model evaluation is very similar to MAE evaluation, however, it is still much different — for example, the best model by MAE, LSTM_lin_fpr_16, has the DM1 score only 131, while the model LSTM_lin_fpr_64_r reached the score of 140. However, the LSTM_lin_fpr class of models is still very good — the DM1 score of first 7 models is higher than 130, then there is a wide gap and the 'x1-x7 or d' model is better than only 110 other tested models using Diebold-Mariano test with $p = 1$ and 5 % level of significance as is shown in Table 6.5. The matrix showing all pair-wise Diebold-Mariano tests on 5 % level of significance is shown in Figures C.3 and C.4.

Model name	**RMSE**	**MAE**	**DM1**	**DM2**	**Pareto**
LSTM_lin_fpr_64_r_3	12.36	2.70	**140**	84	1
LSTM_lin_fpr_8_r_3	13.65	2.92	**138**	79	2
LSTM_lin_fpr_2_r_2	14.89	3.27	**137**	30	4
LSTM_lin_fpr_128_r_1	14.32	3.15	**137**	68	3
LSTM_lin_fpr_1_r_5	15.61	3.40	**135**	9	6
LSTM_lin_fpr_4_r_0	15.41	3.36	**135**	12	5
LSTM_lin_fpr_16_r_0	11.37	2.60	**131**	8	0
x1-x7 or d	14.97	3.53	**110**	41	5
x1,x6,x13,x27 or d	14.96	3.53	**108**	45	5
x1,x2,x7 or d	14.98	3.54	**107**	38	6
x1,x7,x14,x28 or d	15.01	3.53	**107**	35	6
HAR or d	14.94	3.55	**104**	47	5
x1-x7 or w	15.01	3.54	**104**	37	7
x1-x7 or dm	14.92	3.55	**103**	56	5
x1,x2,x7 or w	15.02	3.55	**102**	33	8

Table 6.5: 15 best model when evaluated by Diebold-Mariano test score ($p = 1$)

6.2.5 Main criterion: DM2

It is interesting to note that the selection of the best 15 models by DM2 score does not contain any model from the LSTM_lin_fpr that performed very well in evaluations by other criteria — even by RMSE which is also based on the squared error. Three different classes of models show very good performance — 'HAR', 'x1,x6,x13,x27', and 'x1' autoregressive models with different dummy variables. However, there is no clear relationship between the inclusion of particular dummy variables and the performance of the model, e.g, the models with 'dm' dummy variables are not consistently significantly better than models without these dummy variables by comparing just the DM2 score on the 5 % level of significance, however, the models with 'dm' dummy variables seem to be consistently better when comparing by the actual RMSE values, however, the differences are too small to significant when using the Diebold-Mariano test. The selection of 15 best modely by DM2 score is shown in Table 6.6, while all the pairwise comparisons are shown in Figures C.1 and C.2.

Model name	**RMSE**	**MAE**	**DM1**	**DM2**	**Pareto**
HAR dmy	12.45	5.49	13	**97**	3
HAR w	12.56	4.73	42	**96**	2
x1	12.65	5.13	19	**94**	4
x1,x6,x13,x27 dm	12.49	5.26	25	**94**	2
HAR	12.61	4.68	51	**93**	2
HAR dm	12.42	5.33	18	**93**	2
HAR y	12.65	4.62	55	**93**	2
x1,x6,x13,x27	12.72	4.89	41	**93**	5
x1 dm	12.39	5.34	19	**92**	2
x1 y	12.76	4.96	33	**91**	6
x1,x6,x13,x27 y	12.80	4.78	45	**91**	4
x1 w	12.59	4.91	37	**89**	3
x1,x6,x13,x27 w	12.68	4.85	41	**89**	4
x1 dmy	12.46	5.49	15	**86**	3
x1,x6,x13,x27 dmy	12.54	5.41	16	**86**	3

Table 6.6: 15 best model when evaluated by Diebold-Mariano test score ($p = 2$)

6.3 GARCH models

Even though the GARCH models were included in comparison of previous models, their performance was not good enough to be in the selected set of models — however, these models were in a disadvantage as they were able to use only average daily prices instead of full set of 24 hourly prices and they were able to predict only more long-term volatility — short volatility caused by oscillations of hourly prices may not change the average price and therefore would not have an effect on the volatility predicted by GARCH models. To compare different classes of GARCH models, this section will compare only GARCH models.

The performance of GARCH models is shown in Table 6.7 and Table 6.8. The model gjrGARCH has proven to be the best for used data in all used configurations — gjr-GARCH(1,1) ARMA(1,1) and gjrGARCH(1,1) ARMA(1,0) are on the first two positions, followed by sGARCH(1,0) ARMA(1,1) and then again by gjrGARCH models, this time gjrGARCH(1,0) ARMA(1,1) and gjrGARCH(1,0) ARMA(1,0). Most of other GARCH models are similar in performance, however, with the exception of eGARCH models that performs much worse both with respect to the RMSE and MAE.

Model name	RMSE	MAE	DM1	DM2	Pareto
gjrGARCH(1,1) ARMA(1,1)	13.33	4.59	14.0	7.0	0
gjrGARCH(1,1) ARMA(1,0)	13.47	4.81	7.0	3.0	1
sGARCH(1,0) ARMA(1,1)	13.52	4.89	7.0	5.0	2
gjrGARCH(1,0) ARMA(1,1)	13.71	4.80	9.0	1.0	1
gjrGARCH(1,0) ARMA(1,0)	14.10	4.99	7.0	1.0	3
sGARCH(1,1) ARMA(1,1)	14.43	5.11	7.0	3.0	5
iGARCH(1,1) ARMA(1,1)	14.50	5.20	3.0	2.0	6
apARCH(1,0) ARMA(1,0)	14.28	5.09	4.0	0.0	4
apARCH(1,1) ARMA(1,1)	13.76	5.72	0.0	1.0	3
sGARCH(1,1) ARMA(1,0)	15.31	5.53	0.0	1.0	7
sGARCH(1,0) ARMA(1,0)	14.37	5.45	0.0	0.0	5
apARCH(1,1) ARMA(1,0)	13.99	5.69	0.0	0.0	3
apARCH(1,0) ARMA(1,1)	14.07	5.69	0.0	0.0	3
iGARCH(1,1) ARMA(1,0)	15.33	5.54	0.0	0.0	8
eGARCH(1,1) ARMA(1,0)	19.89	6.47	0.0	0.0	9
eGARCH(1,1) ARMA(1,1)	38.60	6.96	0.0	0.0	10

Table 6.7: Evaluation of GARCH models by actual values.

Model name	RMSE	MAE	DM1	DM2	Pareto	Average
gjrGARCH(1,1) ARMA(1,1)	1.0	1.0	1.0	1.0	0	1.0
gjrGARCH(1,1) ARMA(1,0)	2.0	3.0	4.5	3.5	1	3.25
sGARCH(1,0) ARMA(1,1)	3.0	4.0	4.5	2.0	2	3.375
gjrGARCH(1,0) ARMA(1,1)	4.0	2.0	2.0	7.5	1	3.875
gjrGARCH(1,0) ARMA(1,0)	8.0	5.0	4.5	7.5	3	6.25
sGARCH(1,1) ARMA(1,1)	11.0	7.0	4.5	3.5	5	6.5
iGARCH(1,1) ARMA(1,1)	12.0	8.0	8.0	5.0	6	8.25
apARCH(1,0) ARMA(1,0)	9.0	6.0	7.0	13.0	4	8.75
apARCH(1,1) ARMA(1,1)	5.0	14.0	12.5	7.5	3	9.75
sGARCH(1,1) ARMA(1,0)	13.0	10.0	12.5	7.5	7	10.75
sGARCH(1,0) ARMA(1,0)	10.0	9.0	12.5	13.0	5	11.125
apARCH(1,1) ARMA(1,0)	6.0	13.0	12.5	13.0	3	11.125
apARCH(1,0) ARMA(1,1)	7.0	12.0	12.5	13.0	3	11.125
iGARCH(1,1) ARMA(1,0)	14.0	11.0	12.5	13.0	8	12.625
eGARCH(1,1) ARMA(1,0)	15.0	15.0	12.5	13.0	9	13.875
eGARCH(1,1) ARMA(1,1)	16.0	16.0	12.5	13.0	10	14.375

Table 6.8: Evaluation of GARCH models by actual values.

Chapter 7

Conclusion

As are electricity markets getting more and more important, it is getting vital to have good tools for analysing and predicting the behavior of such markets. While such tools are already developed and well researched for stock markets, the electricity markets are different in structure, for example, it is nearly impossible to store electrical energy, therefore these tools are often misleading or even wrong when used on electrical markets. While accurate price and volatility prediction on electricity markets is very difficult, it is possible to obtain results accurate enough using only price and time data without any other exogenous variable. This work compares models representing two main approaches toward volatility prediction in electrical markets — artificial neural networks models and autoregressive models. Each approache is represented by several different classes of models and each class is represented by model with several different configurations. In total, 128 different models are used, evaluated on real data from Czech Electricity Day-ahead market and compared with each other based on multiple criteria. While there several works regarding the volatility prediction on electrical markets have shown in past, to the extent of our knowledge, there is no work comparing the predictive performance of autoregressive models and recurrent artificial neural network models with LSTM units, moreover, the comparison is thorough and takes into account many different configurations of used models.

While this work aimed at comparison of different models for predicting volatility on the electricity markets, the comparison contains focused only on two main approaches — recurrent artificial neural network models with LSTM units and autoregressive models — as these are two main approaches in the literature, however, we would like to compare many different models in future — including but not limiting to *least square support vector machines* (LS-SVM), regression trees, more simple feed-forward artificial neural network models, ordinardy recurrent neural network models, and many more.

Moreover, this work used as for prediction of volatility and electricity prices only historical prices as the goal was to test whether even just this data may provide sufficiently accurate predictions, however, future works will also focus on different types of inputs as they may lead to much better predictions. There are many different variables that can be useful for electricity price prediction, e.g., the electricity load, the amount of electricity sold, the price of coal and gas (as they are used in some types of power-plants), weather (influences both demand — heating or cooling — and supply — solar or wind power-plants), or even the number and types of individual power-plants in the country.

Another limiting factor of this work were resources as fitting of more complex neural network is just computationally very costly, therefore future works with more resources will focus on using even more complex neural networks (e.g., deep-learning) and more complex cross-validation of results when the models will be tested on many different partitioning into test and training datasets which will lead to more accurate and less biased estimates of models' performance. With more resources, the models also will be tested on data from different electricity markets as individual markets may differ in structure and different models may be more suitable for different markets.

Individual classes of models and individual models themselves are described in Chapter 2, different configurations are also described in this chapter. Chapter 3 describes and analyses the data and show several trends in the data as well. Chapter 4 shortly describe the model implementation and then describes the design of experiments. The performance of different artificial neural network models within their classes is also discussed in this chapter. Finally, all models are compared in Chapter 6 with respect to several performance criteria.

Three different classes of models were compared — autoregressive models, GARCH models and artificial recurrent neural network models with LSTM units. Each of these model has several advantages and disadvantages, therefore class of models is recommended for different purposes. GARCH models were not as accurate as other models, however, they use only average daily prices and not the intra-daily data, which may not be always available. Moreover, GARCH models can be used for bot volatility and price prediction. From the family of GARCH models, the GJR-GARCH model (gjrGARCH) has proven to be most suitable, however, even the simple standard GARCH model with ARMA(1,1) mean model has proven to be competitive.

The autoregressive models were also very competitive, especially when compared with dummy variables. Very good results had HAR models as they were able to model both short term volatility and long term volatility, however, even the very simple AR(1) with dummy variables was comparably good, the 'x1 dm' was even better than HAR models in terms of RMSE but only by 0.03. Moreover, the autoregessive models were the best when the main criterion was Diebold-Mariano test on 5 % level of significance.

However, the best results in general were achieved by two stage recurrent artificial neural network model with LSTM units — this model first predicted the set of day-ahead hourly prices and then calculated the volatility. These class of models was significantly the most accurate when using RMSE, MAE, and DM1 criteria. Moreover, the main advantage of this model is that it provides accurate forecasts for both day-ahead hourly prices and daily volatility. However, when compared to autoregressive models, it has also several disadvantages — one of the most important disadvantages is that recurrent neural networks models are very difficult to analyze and interpret as they behave more or less as black-box models. Moreover, some of the complex neural network models are very costly to train, which may be an issue under certain circumstances. Other two neural network models — those directly predicting the volatility, were not as successful as this one, the neural network models with sigmoid output layer and LSTM units were consistently one of the worst models while most of the neural network models with linear output layer and LSTM units were in the middle of evaluated models.

Under most circumstances, the two stage artificial neural network model with LSTM units is recommended for both price and volatility prediction as it achieves very good experimental results and it is very flexible to adapt to changing markets.

Bibliography

[1] N. Amjady. "Day-Ahead Price Forecasting of Electricity Markets by a New Fuzzy Neural Network". In: *IEEE Transactions on Power Systems* 21.2 (May 2006), pp. 887–896. DOI: 10.1109/tpwrs.2006.873409. URL: http://dx.doi.org/10.1109/TPWRS.2006.873409.

[2] N. Amjady and M. Hemmati. "Energy price forecasting - problems and proposals for such predictions". In: *IEEE Power and Energy Magazine* 4.2 (Mar. 2006), pp. 20–29. DOI: 10.1109/mpae.2006.1597990. URL: http://dx.doi.org/10.1109/MPAE.2006.1597990.

[3] Nima Amjady and Ali Daraeepour. "Day-ahead price forecasting of electricity markets by combination of mutual information technique and neural network". In: *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*. Institute of Electrical & Electronics Engineers (IEEE), July 2008. DOI: 10.1109/pes.2008.4596794. URL: http://dx.doi.org/10.1109/PES.2008.4596794.

[4] S. Anbazhagan and N. Kumarappan. "Day-Ahead Deregulated Electricity Market Price Forecasting Using Recurrent Neural Network". In: *IEEE Systems Journal* 7.4 (Dec. 2013), pp. 866–872. DOI: 10.1109/jsyst.2012.2225733. URL: http://dx.doi.org/10.1109/JSYST.2012.2225733.

[5] Torben G. Andersen and Tim Bollerslev. "Answering the Skeptics: Yes, Standard Volatility Models do Provide Accurate Forecasts". In: *International Economic Review* 39.4 (Nov. 1998), p. 885. DOI: 10.2307/2527343. URL: http://dx.doi.org/10.2307/2527343.

[6] Torben Andersen, Tim Bollerslev, Peter Christoffersen, and Francis Diebold. *Volatility Forecasting*. Tech. rep. Mar. 2005. DOI: 10.3386/w11188. URL: http://dx.doi.org/10.3386/w11188.

[7] O. E. Barndorff-Nielsen. "Power and Bipower Variation with Stochastic Volatility and Jumps". In: *Journal of Financial Econometrics* 2.1 (Dec. 2004), pp. 1–37. DOI: 10.1093/jjfinec/nbh001. URL: http://dx.doi.org/10.1093/jjfinec/nbh001.

[8] Stefan Behnel, Robert Bradshaw, Craig Citro, Lisandro Dalcin, Dag Sverre Seljebotn, and Kurt Smith. "Cython: The Best of Both Worlds". In: *Computing in Science & Engineering* 13.2 (Mar. 2011), pp. 31–39. DOI: 10.1109/mcse.2010.118. URL: http://dx.doi.org/10.1109/MCSE.2010.118.

[9] D. Benaouda and F. Murtagh. "Hybrid Wavelet Model for Electricity Pool-price Forecasting in a Deregulated Electricity Market". In: *2006 IEEE International Conference on Engineering of Intelligent Systems*. Institute of Electrical & Electronics Engineers (IEEE), 2006. DOI: 10.1109/iceis.2006.1703198. URL: http://dx.doi.org/10.1109/ICEIS.2006.1703198.

[10] Jaya P. N. Bishwal. "Some New Estimators of Integrated Volatility". In: *Open Journal of Statistics* 01.02 (2011), pp. 74–80. DOI: 10.4236/ojs.2011.12008. URL: http://dx.doi.org/10.4236/ojs.2011.12008.

[11] R.C. Garcia, J. Contreras, M. vanAkkeren, and J.B.C. Garcia. "A GARCH Forecasting Model to Predict Day-Ahead Electricity Prices". In: *IEEE Transactions on Power Systems* 20.2 (May 2005), pp. 867–874. DOI: 10.1109/tpwrs.2005.846044. URL: http://dx.doi.org/10.1109/TPWRS.2005.846044.

[12] Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. "Learning to Forget: Continual Prediction with LSTM". In: *Neural Computation* 12.10 (Oct. 2000), pp. 2451–2471. DOI: 10.1162/089976600300015015. URL: http://dx.doi.org/10.1162/089976600300015015.

[13] Alexios Ghalanos. *Introduction to the rugarch package.* Version 1.3-1). 2014. URL: https://cran.r-project.org/web/packages/rugarch/vignettes/Introduction_to_the_rugarch_package.pdf.

[14] V Gonzalez, J Contreras, and D W Bunn. "Forecasting prices and volatilities using a hybrid fundamental econometric model". In: *IEEE PES General Meeting*. Institute of Electrical & Electronics Engineers (IEEE), July 2010. DOI: 10.1109/pes.2010.5590100. URL: http://dx.doi.org/10.1109/PES.2010.5590100.

[15] P. Reinhard Hansen and A. Lunde. "Does anything beat a GARCH(1, 1)? A comparison based on test for superior predictive ability". In: *2003 IEEE International Conference on Computational Intelligence for Financial Engineering, 2003. Proceedings.* Institute of Electrical & Electronics Engineers (IEEE), 2003. DOI: 10.1109/cifer.2003.1196275. URL: http://dx.doi.org/10.1109/CIFER.2003.1196275.

[16] Erik Haugom, Sjur Westgaard, Per Bjarte Solibakke, and Gudbrand Lien. "Modelling day ahead Nord Pool forward price volatility: Realized volatility versus GARCH models". In: *2010 7th International Conference on the European Energy Market*. Institute of Electrical & Electronics Engineers (IEEE), June 2010. DOI: 10.1109/eem.2010.5558687. URL: http://dx.doi.org/10.1109/EEM.2010.5558687.

[17] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[18] John D. Hunter. "Matplotlib: A 2D Graphics Environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/mcse.2007.55. URL: http://dx.doi.org/10.1109/MCSE.2007.55.

[19] Rob J Hyndman. *forecast: Forecasting functions for time series and linear models.* R package version 6.1. 2015. URL: http://github.com/robjhyndman/forecast.

[20] Rob J Hyndman and Yeasmin Khandakar. "Automatic time series forecasting: the forecast package for R". In: *Journal of Statistical Software* 26.3 (2008), pp. 1–22. URL: http://ideas.repec.org/a/jss/jstsof/27i03.html.

[21] Eric Jones, Travis Oliphant, Pearu Peterson, et al. *SciPy: Open source scientific tools for Python*. [Online; accessed 2015-05-12]. 2001–. URL: http://www.scipy.org/.

[22] Chengjun Li and Ming Zhang. "Application of GARCH Model in the Forecasting of Day-Ahead Electricity Prices". In: *Third International Conference on Natural Computation (ICNC 2007)*. Institute of Electrical & Electronics Engineers (IEEE), 2007. DOI: 10.1109/icnc.2007.252. URL: http://dx.doi.org/10.1109/ICNC.2007.252.

[23] Paras Mandal, Tomonobu Senjyu, Naomitsu Urasaki, Atsushi Yona, Toshihisa Funabashi, and Anurag K. Srivastava. "Price Forecasting for Day-Ahead Electricity Market Using Recursive Neural Network". In: *2007 IEEE Power Engineering Society General Meeting*. Institute of Electrical & Electronics Engineers (IEEE), June 2007. DOI: 10.1109/pes.2007.385970. URL: http://dx.doi.org/10.1109/PES.2007.385970.

[24] Dawit H. Mazengia and Le Anh Tuan. "Forecasting spot electricity market prices using time series models". In: *2008 IEEE International Conference on Sustainable Energy Technologies*. Institute of Electrical & Electronics Engineers (IEEE), Nov. 2008. DOI: 10.1109/icset.2008.4747199. URL: http://dx.doi.org/10.1109/ICSET.2008.4747199.

[25] Wes McKinney. "Data Structures for Statistical Computing in Python". In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 51–56.

[26] F.J. Nogales, J. Contreras, A.J. Conejo, and R. Espinola. "Forecasting next-day electricity prices by time series models". In: *IEEE Transactions on Power Systems* 17.2 (May 2002), pp. 342–348. DOI: 10.1109/tpwrs.2002.1007902. URL: http://dx.doi.org/10.1109/TPWRS.2002.1007902.

[27] Fernando Perez and Brian E. Granger. "IPython: A System for Interactive Scientific Computing". In: *Computing in Science & Engineering* 9.3 (2007), pp. 21–29. DOI: 10.1109/mcse.2007.53. URL: http://dx.doi.org/10.1109/MCSE.2007.53.

[28] R Development Core Team. *R: A Language and Environment for Statistical Computing*. ISBN 3-900051-07-0. R Foundation for Statistical Computing. Vienna, Austria, 2011. URL: http://www.R-project.org/.

[29] Tom Schaul, Justin Bayer, Daan Wierstra, Yi Sun, Martin Felder, Frank Sehnke, Thomas Rückstieß, and Jürgen Schmidhuber. "PyBrain". In: *Journal of Machine Learning Research* 11 (2010), pp. 743–746.

[30] J.S. Seabold and J. Perktold. "Statsmodels: Econometric and Statistical Modeling with Python". In: *Proceedings of the 9th Python in Science Conference*. 2010.

[31] M. Shafie-khah, M. Parsa Moghaddam, and M.K. Sheikh-El-Eslami. "Price forecasting of day-ahead electricity markets using a hybrid forecast method". In: *Energy Conversion and Management* 52.5 (May 2011), pp. 2165–2169. DOI: 10.1016/j.enconman.2010.10.047. URL: http://dx.doi.org/10.1016/j.enconman.2010.10.047.

[32] Stefan van der Walt, S. Chris Colbert, and Gaël Varoquaux. "The NumPy Array: A Structure for Efficient Numerical Computation". In: *Computing in Science & Engineering* 13.2 (Mar. 2011), pp. 22–30. DOI: 10.1109/mcse.2011.37. URL: http://dx.doi.org/10.1109/MCSE.2011.37.

[33] QingQing Wang, ZhaoYang Dong, Xue Li, Junhua Zhao, and Kit Po Wong. "An Effective Approach to Predicting Electricity Market Price Spikes". In: *2007 IEEE Power Engineering Society General Meeting*. Institute of Electrical & Electronics Engineers (IEEE), June 2007. DOI: 10.1109/pes.2007.385852. URL: http://dx.doi.org/10.1109/PES.2007.385852.

[34] Rafał Weron. "Electricity price forecasting: A review of the state-of-the-art with a look into the future". In: *International Journal of Forecasting* 30.4 (Oct. 2014), pp. 1030–1081. DOI: 10.1016/j.ijforecast.2014.08.008. URL: http://dx.doi.org/10.1016/j.ijforecast.2014.08.008.

[35] Lei Wu and Mohammad Shahidehpour. "A hybrid model for integrated day-ahead electricity price and load forecasting in smart grid". In: *IET Generation, Transmission & Distribution* 8.12 (Dec. 2014), pp. 1937–1950. DOI: 10.1049/iet-gtd.2013.0927. URL: http://dx.doi.org/10.1049/iet-gtd.2013.0927.

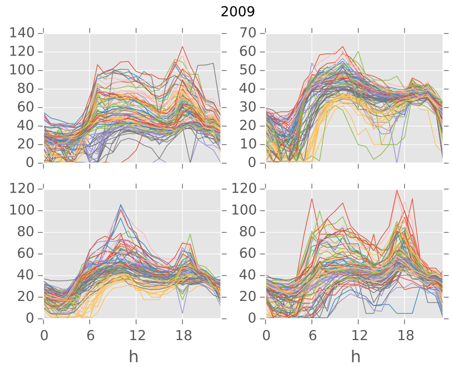
[36] Jianhua Zhang, Jian Han, Rui Wang, and Guolian Hou. "Day-ahead electricity price forecasting based on rolling time series and least square-support vector machine model". In: *2011 Chinese Control and Decision Conference (CCDC)*. Institute of Electrical & Electronics Engineers (IEEE), May 2011. DOI: 10.1109/ccdc.2011.5968342. URL: http://dx.doi.org/10.1109/CCDC.2011.5968342.

[37] Jun Hua Zhao, Zhao Yang Dong, Xue Li, and Kit Po Wong. "A Framework for Electricity Price Spike Analysis With Advanced Data Mining Methods". In: *IEEE Transactions on Power Systems* 22.1 (Feb. 2007), pp. 376–385. DOI: 10.1109/tpwrs.2006.889139. URL: http://dx.doi.org/10.1109/TPWRS.2006.889139.

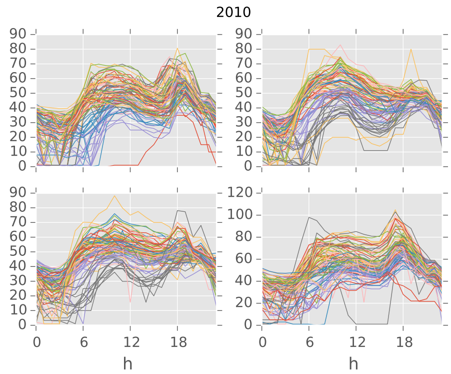
Appendices

Appendix A

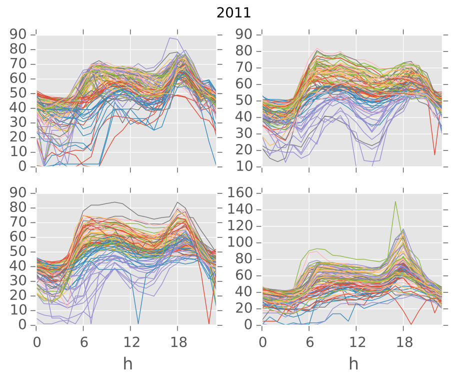
Price dataset plots



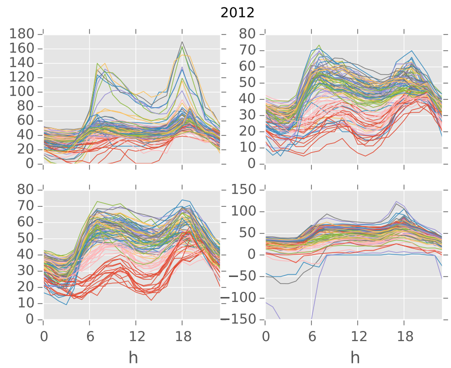
(a) Daily prices for year 2009



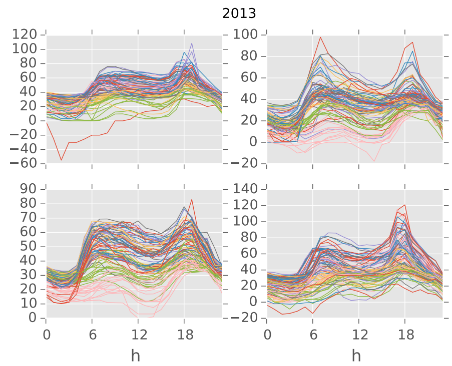
(b) Daily prices for year 2010



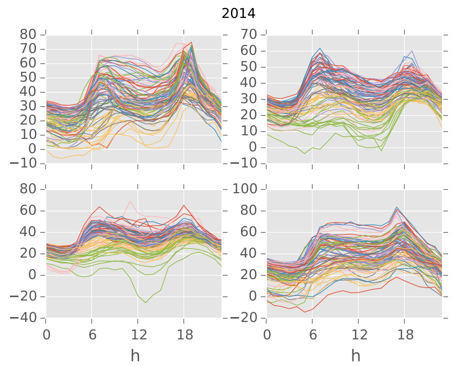
(c) Daily prices for year 2011



(d) Daily prices for year 2012



(e) Daily prices for year 2013



(f) Daily prices for year 2014

Figure A.1: These figures shows the daily prices for every day in given year and quarter. They show that the price follows different patterns in different parts of the year and also that the prices often behaves chaotically and unexpectedly.The order of quarters in subfigures is Q1,Q2,Q3,Q4 from right to left, up to down.

Appendix B

Day-ahead price prediction by LSTM linear models

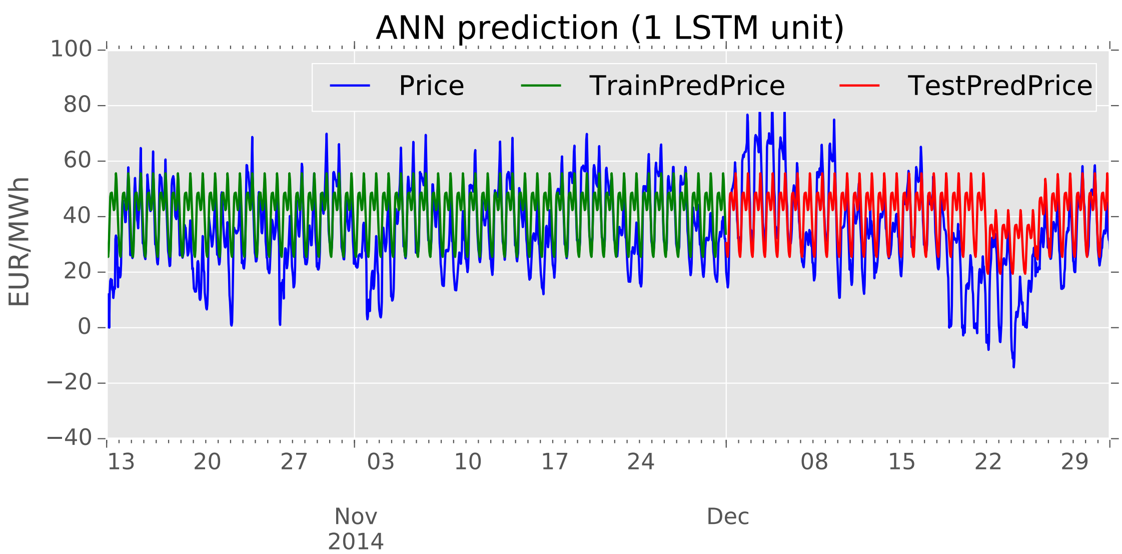


Figure B.1: Example output of an ANN with 1 LSTM unit

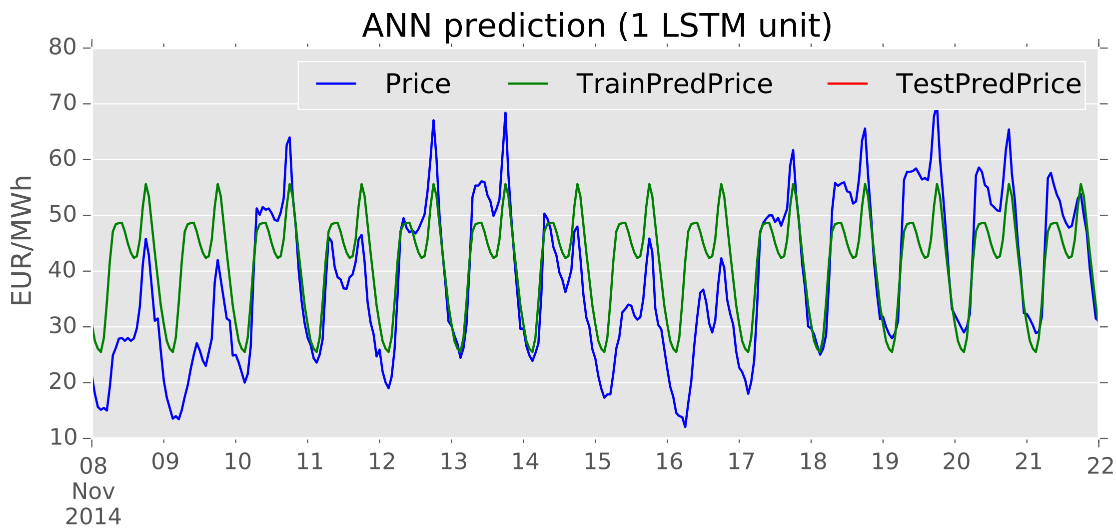


Figure B.2: Detail of output of an ANN with 1 LSTM unit

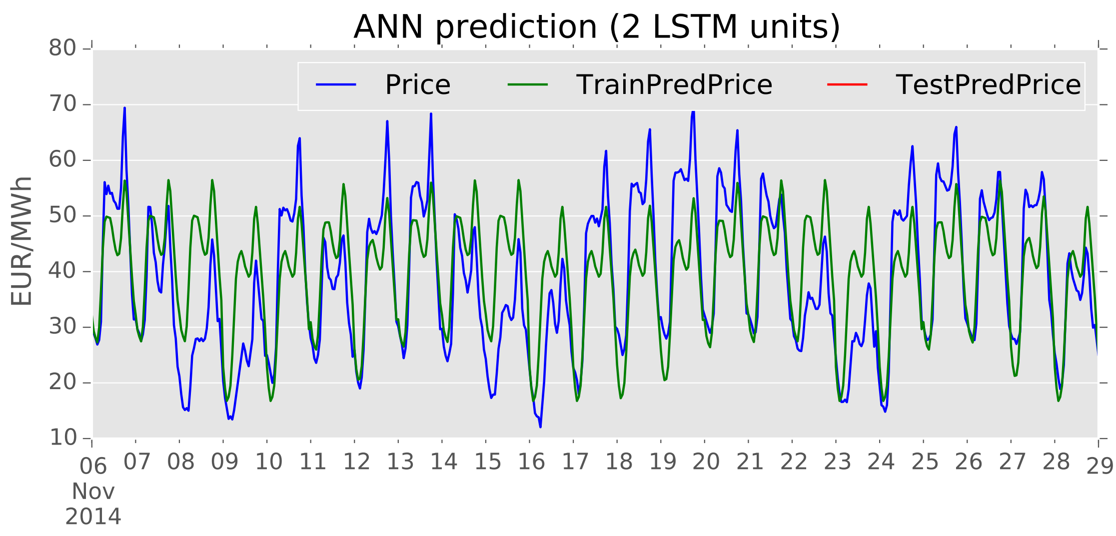


Figure B.3: Example output of an ANN with 2 LSTMs unit

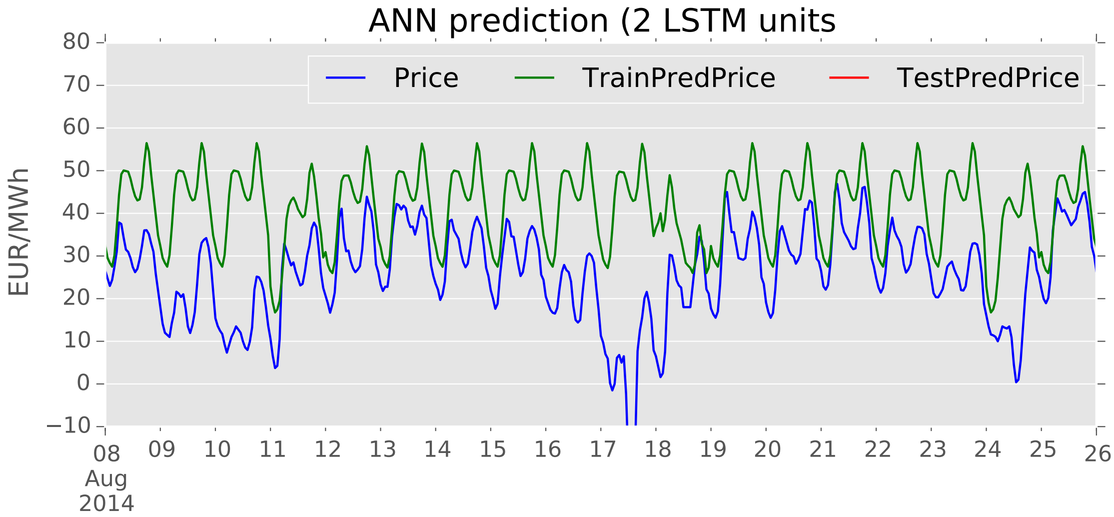


Figure B.4: Detail of output of an ANN with 2 LSTMs unit

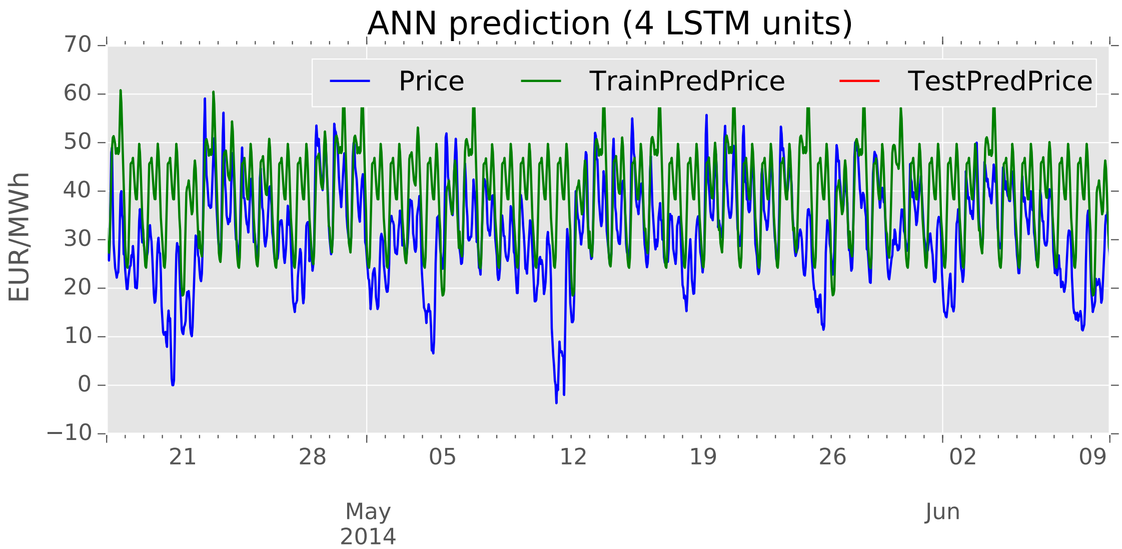


Figure B.5: Example output of an ANN with 4 LSTMs unit

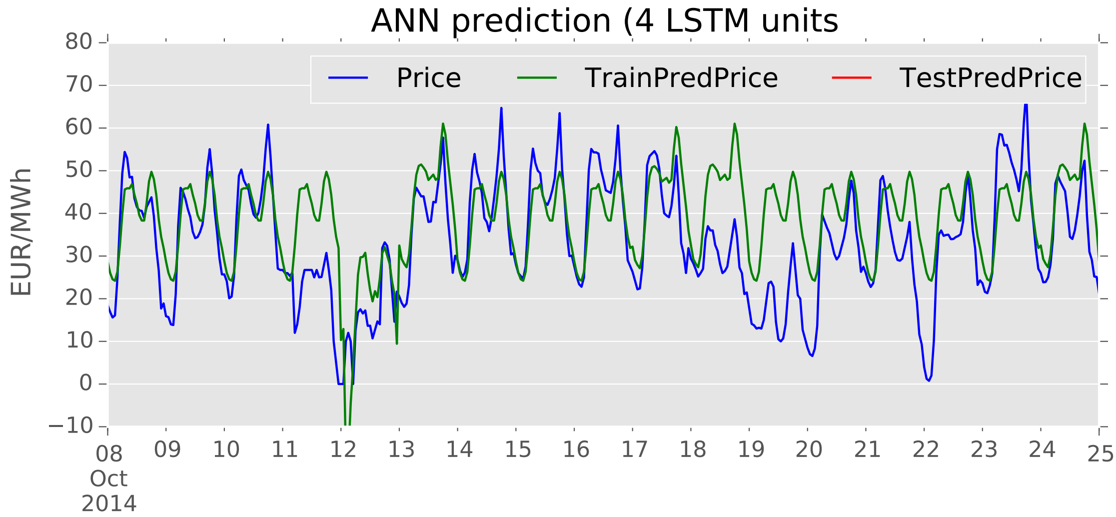


Figure B.6: Detail of output of an ANN with 4 LSTMs unit

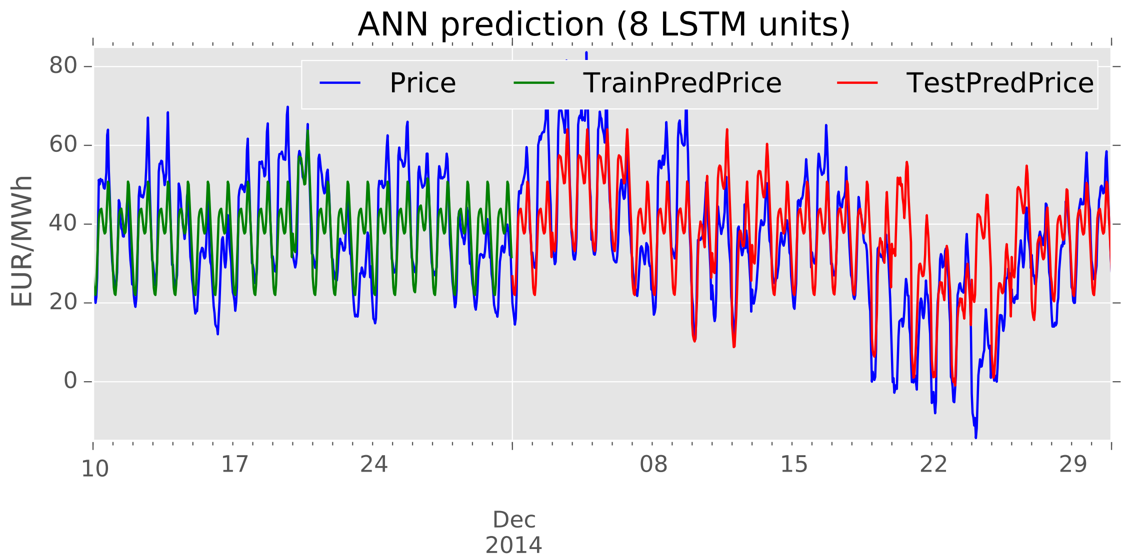


Figure B.7: Example output of an ANN with 8 LSTMs unit

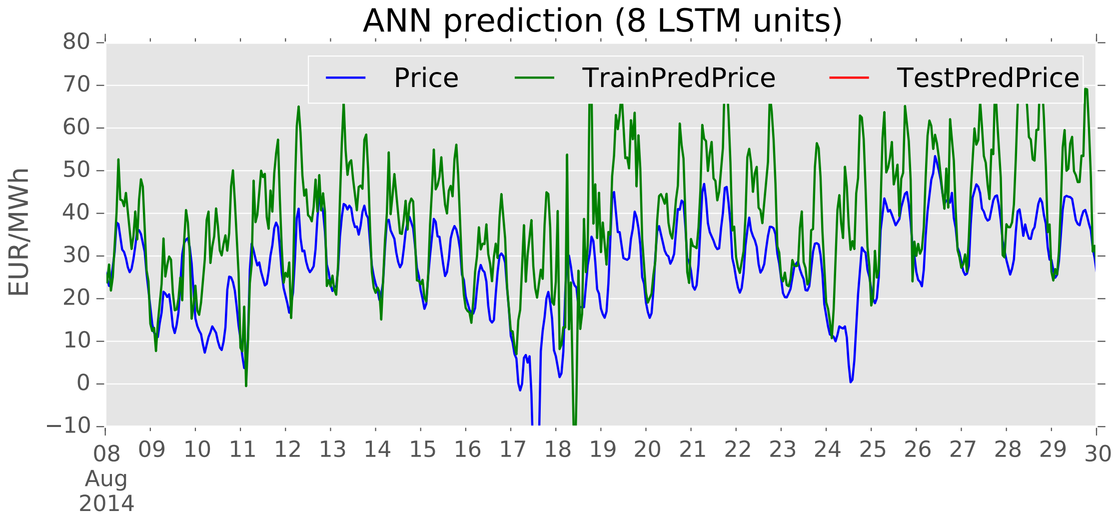


Figure B.8: Detail of output of an ANN with 8 LSTMs unit

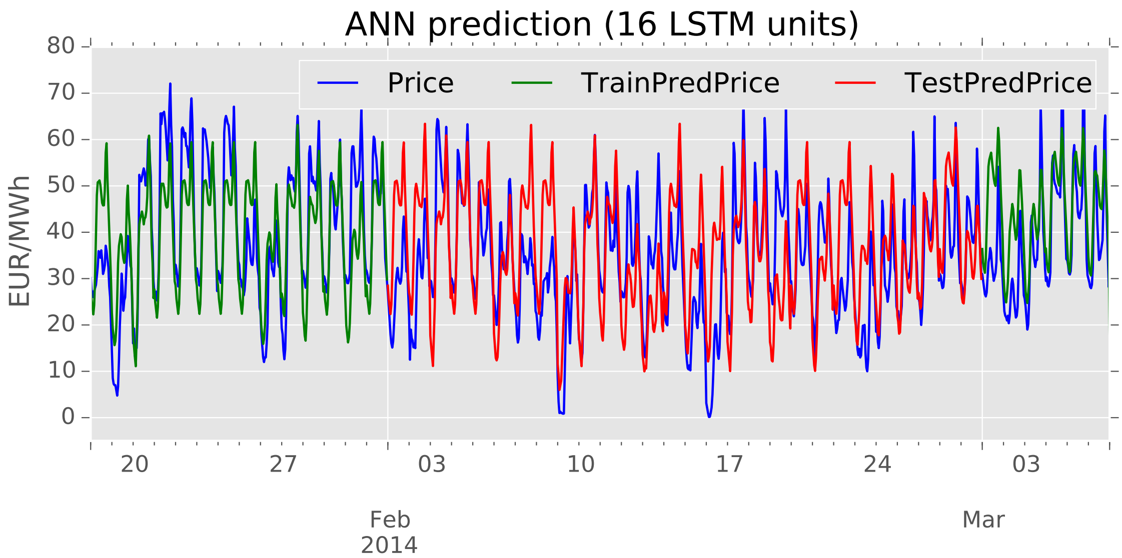


Figure B.9: Example output of an ANN with 16 LSTMs unit

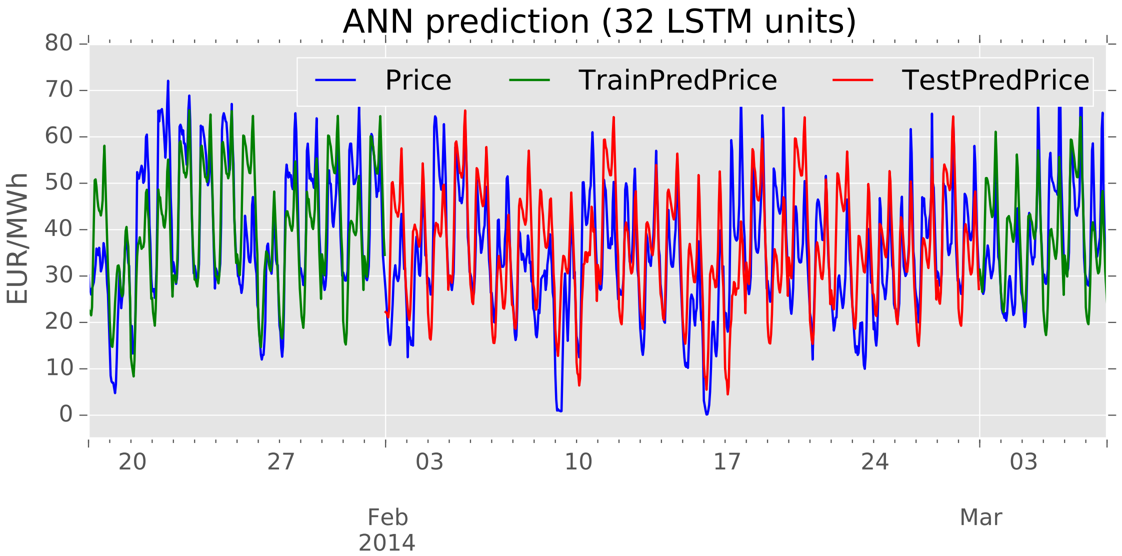


Figure B.10: Example output of an ANN with 32 LSTMs unit

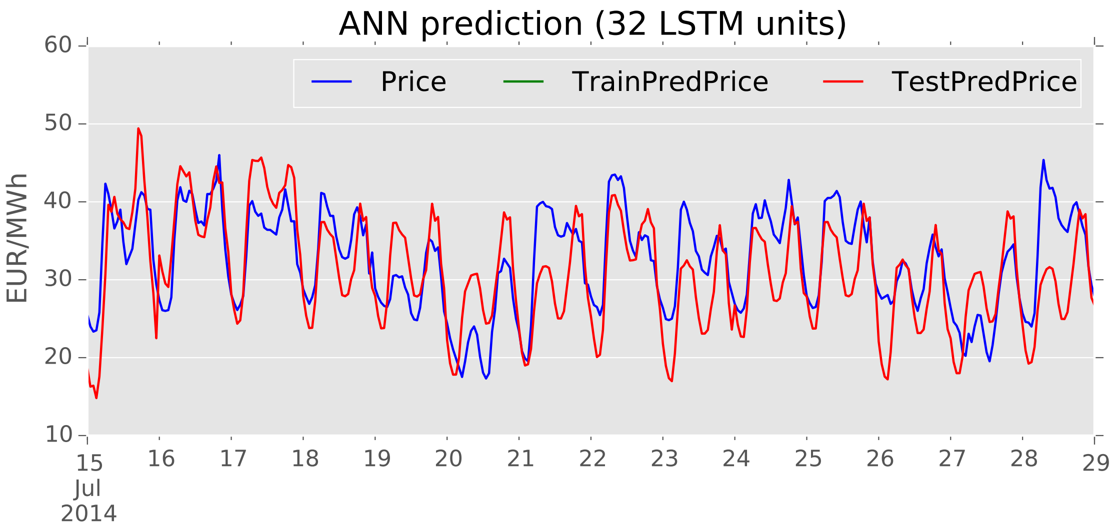


Figure B.11: Detail of output of an ANN with 32 LSTMs unit

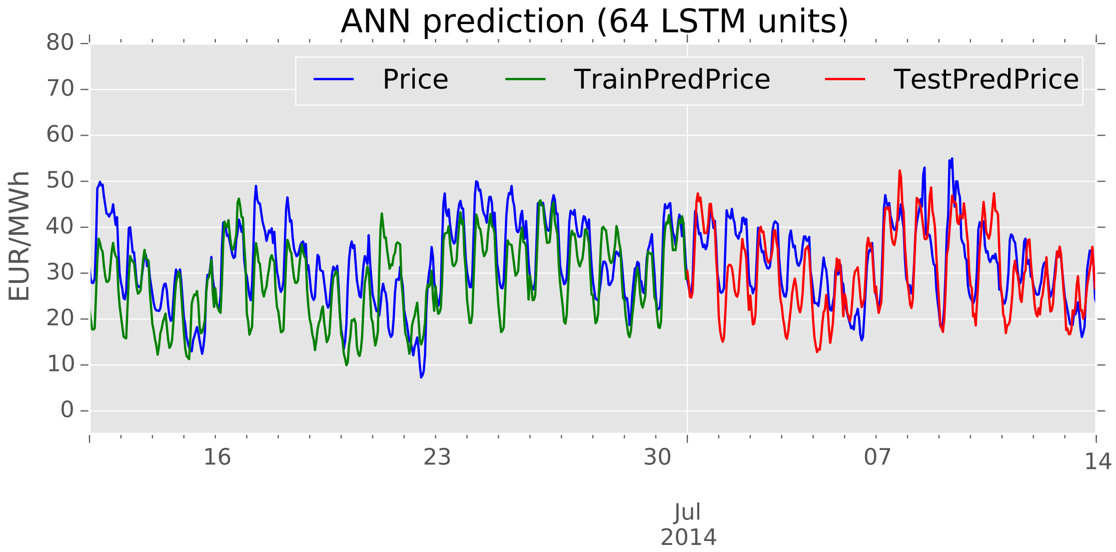


Figure B.12: Example output of an ANN with 64 LSTMs unit

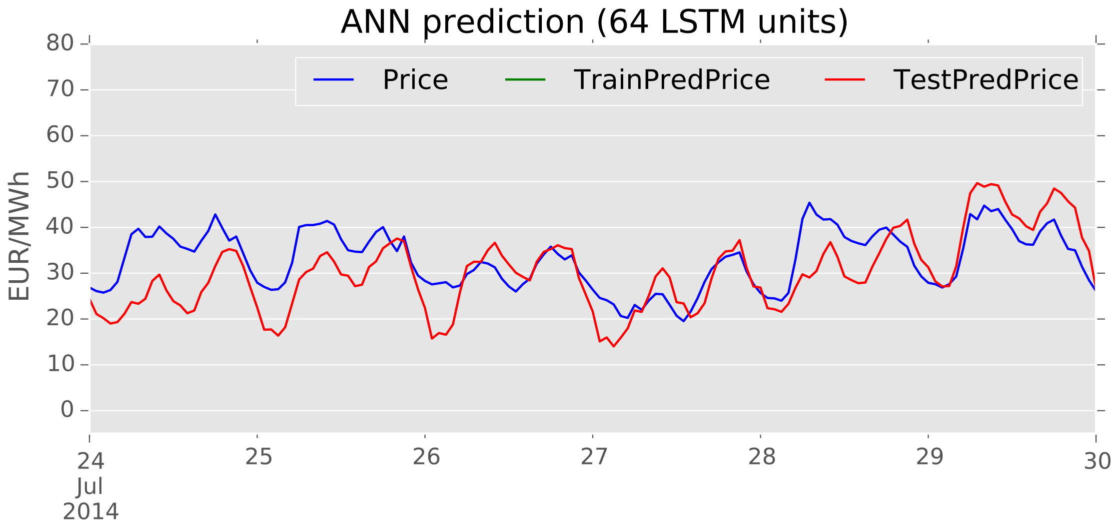


Figure B.13: Detail of output of an ANN with 64 LSTMs unit

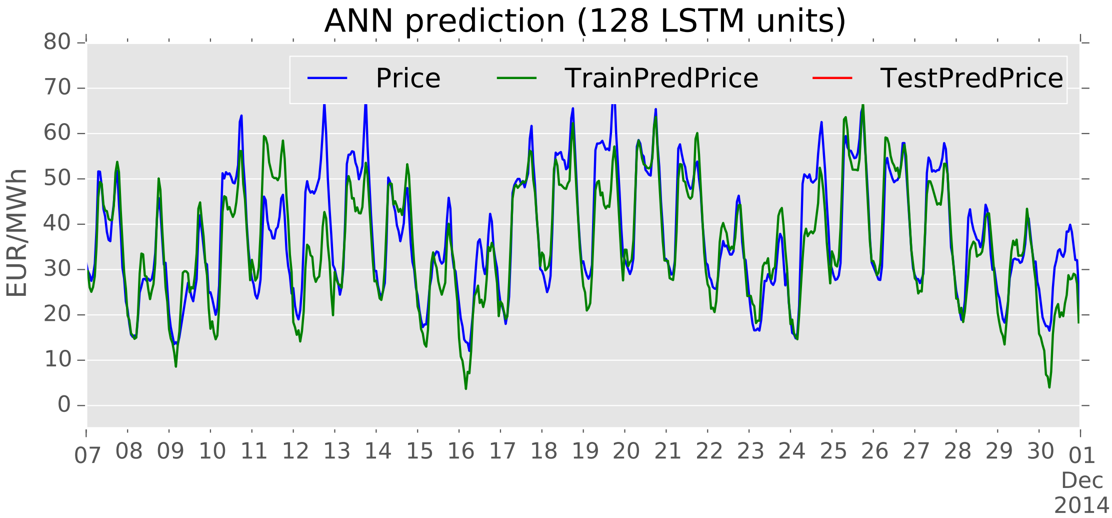


Figure B.14: Example output of an ANN with 128 LSTMs unit

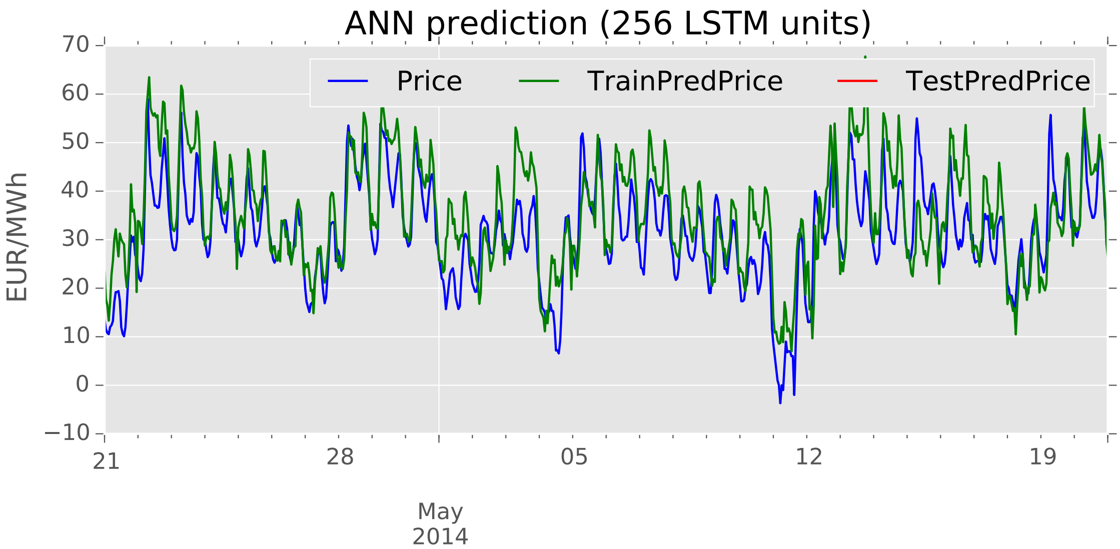


Figure B.15: Example output of an ANN with 256 LSTMs unit

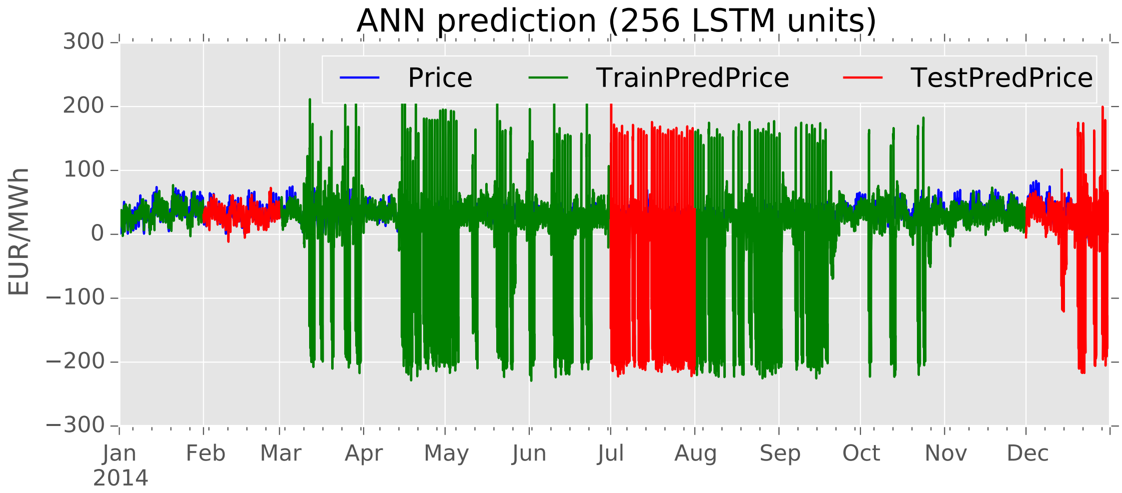


Figure B.16: Example output of a wrongly trained ANN with 256 LSTMs unit

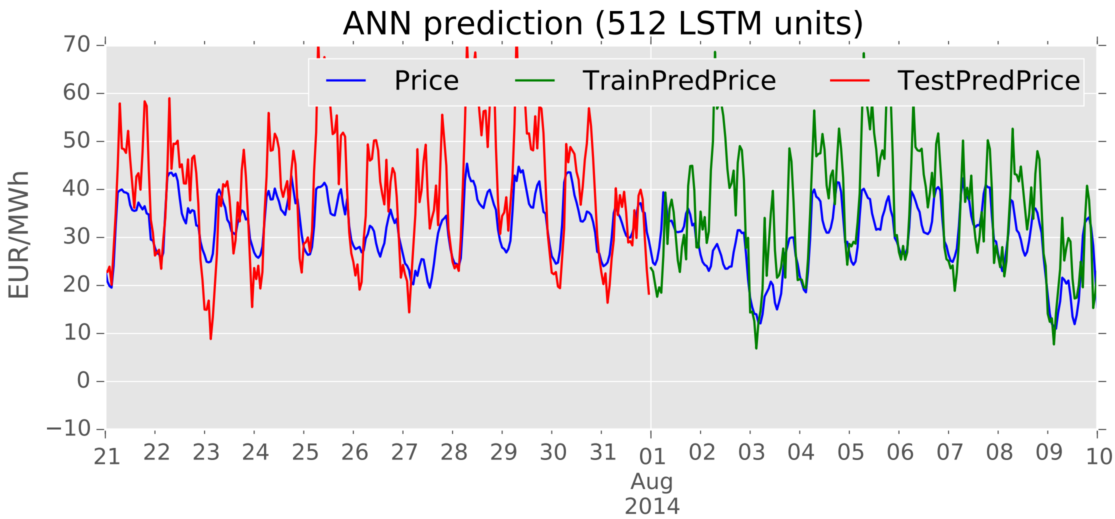


Figure B.17: Example output of an ANN with 512 LSTMs unit

Appendix C

Diebold-Mariano tests comparison

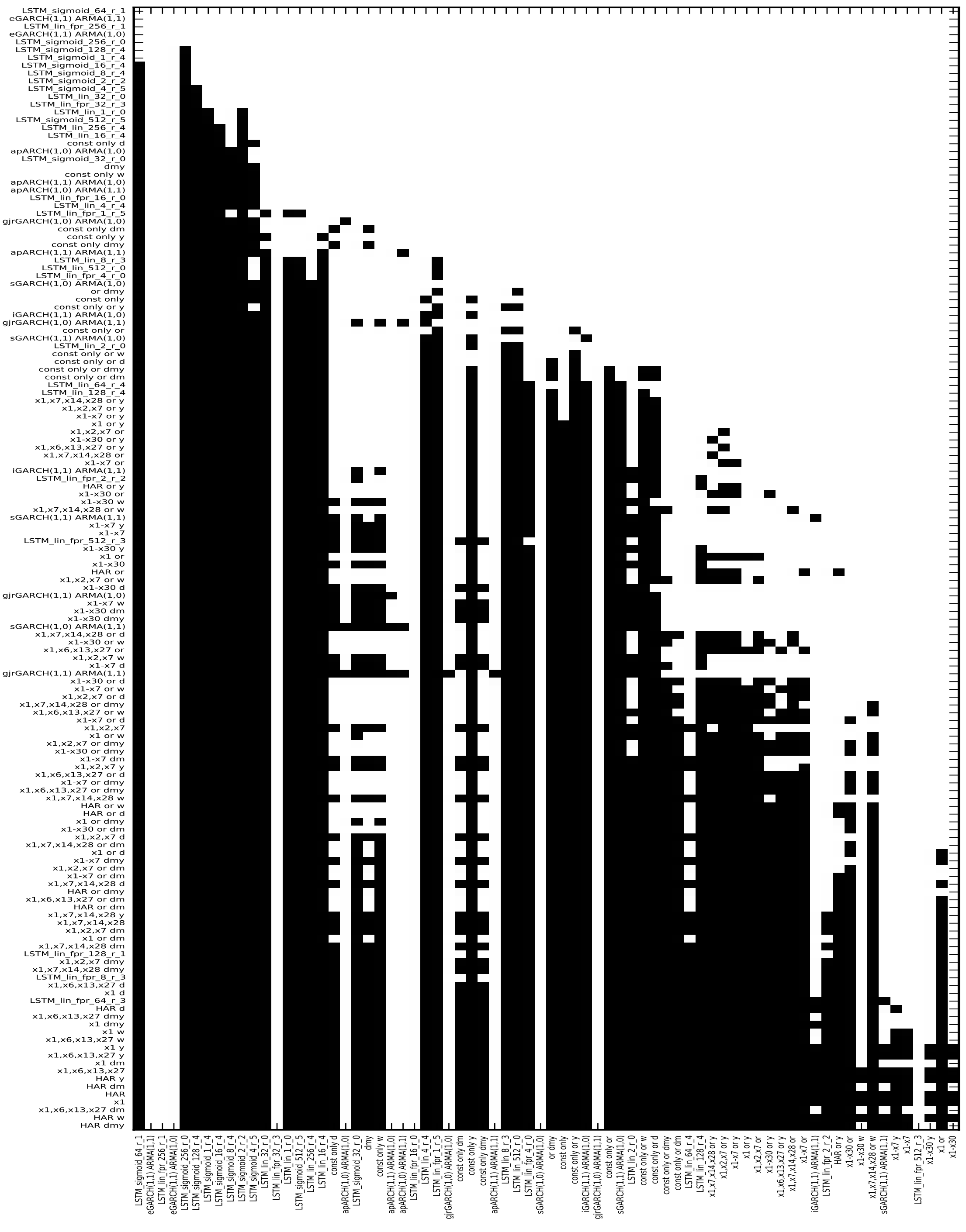


Figure C.1: Matrix of models sorted by the number of dominations using the Diebold-Mariano test with power 2 (squared error) — point $a_{i,j}$ is black if and only if model i dominates model j using the test on 5 % level of significance.

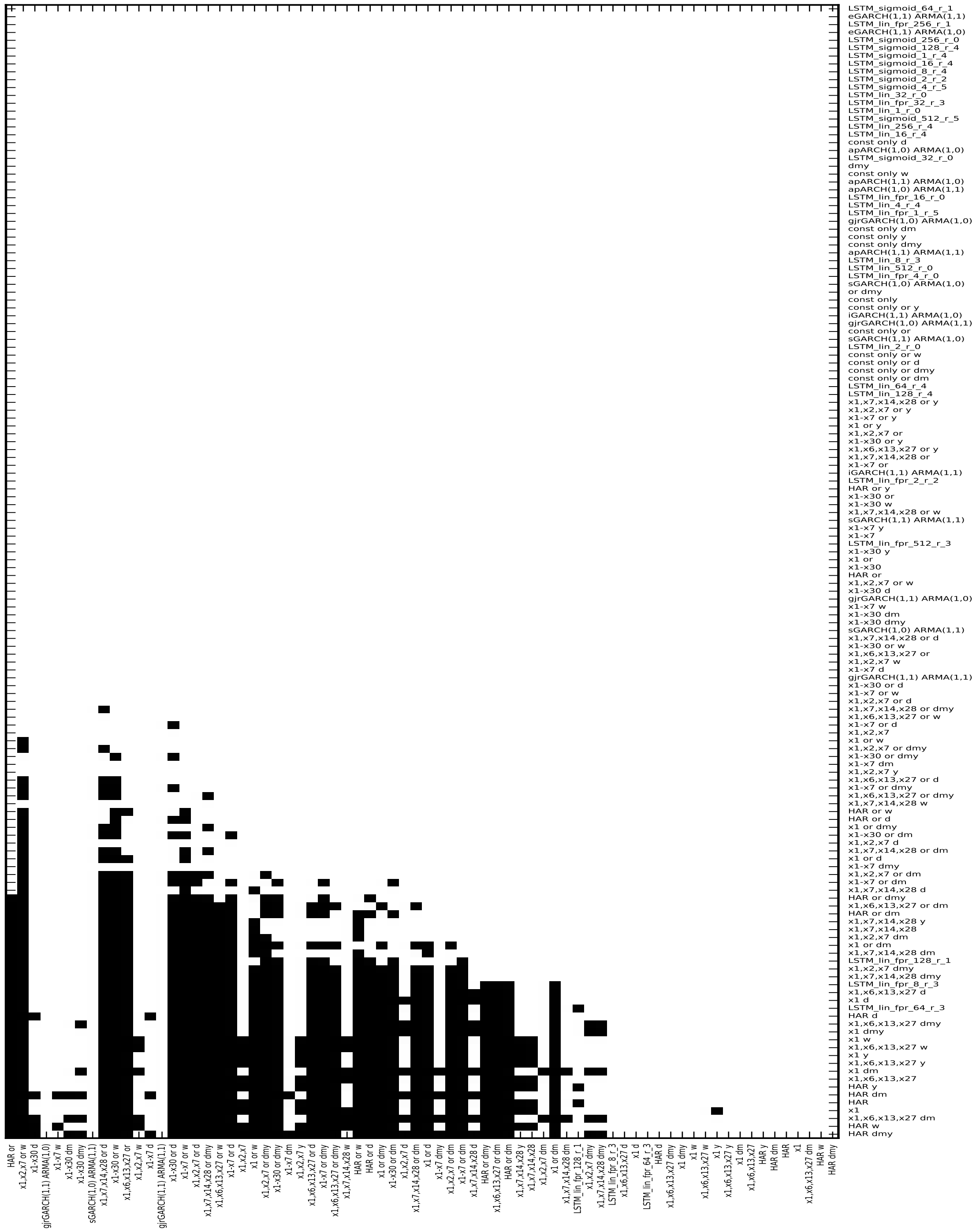


Figure C.2: Continuation of Figure C.1

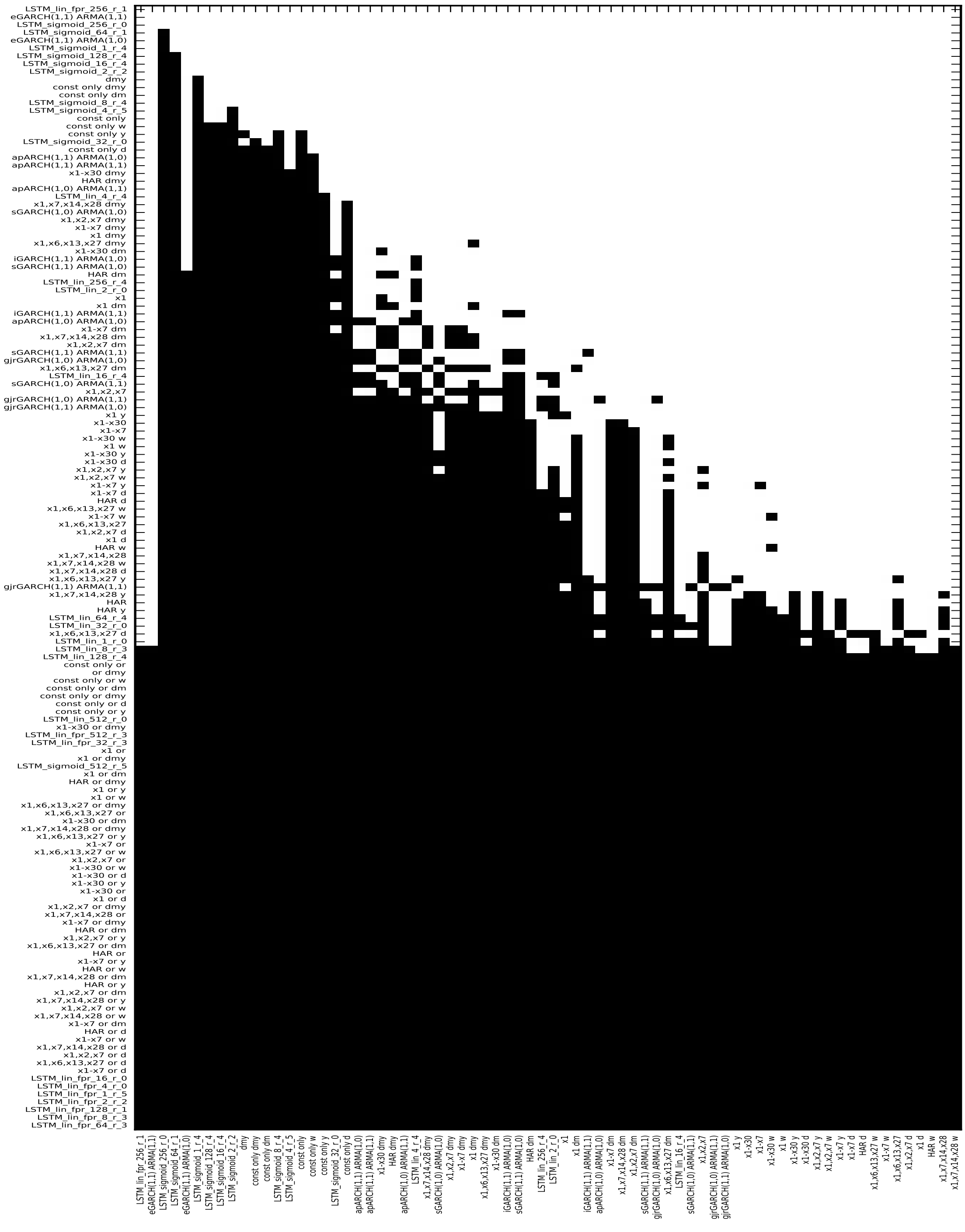


Figure C.3: Matrix of models sorted by the number of dominations using the Diebold-Mariano test with power 1 (absolute error) — point $a_{i,j}$ is black if and only if model i dominates model j using the test on 5 % level of significance.

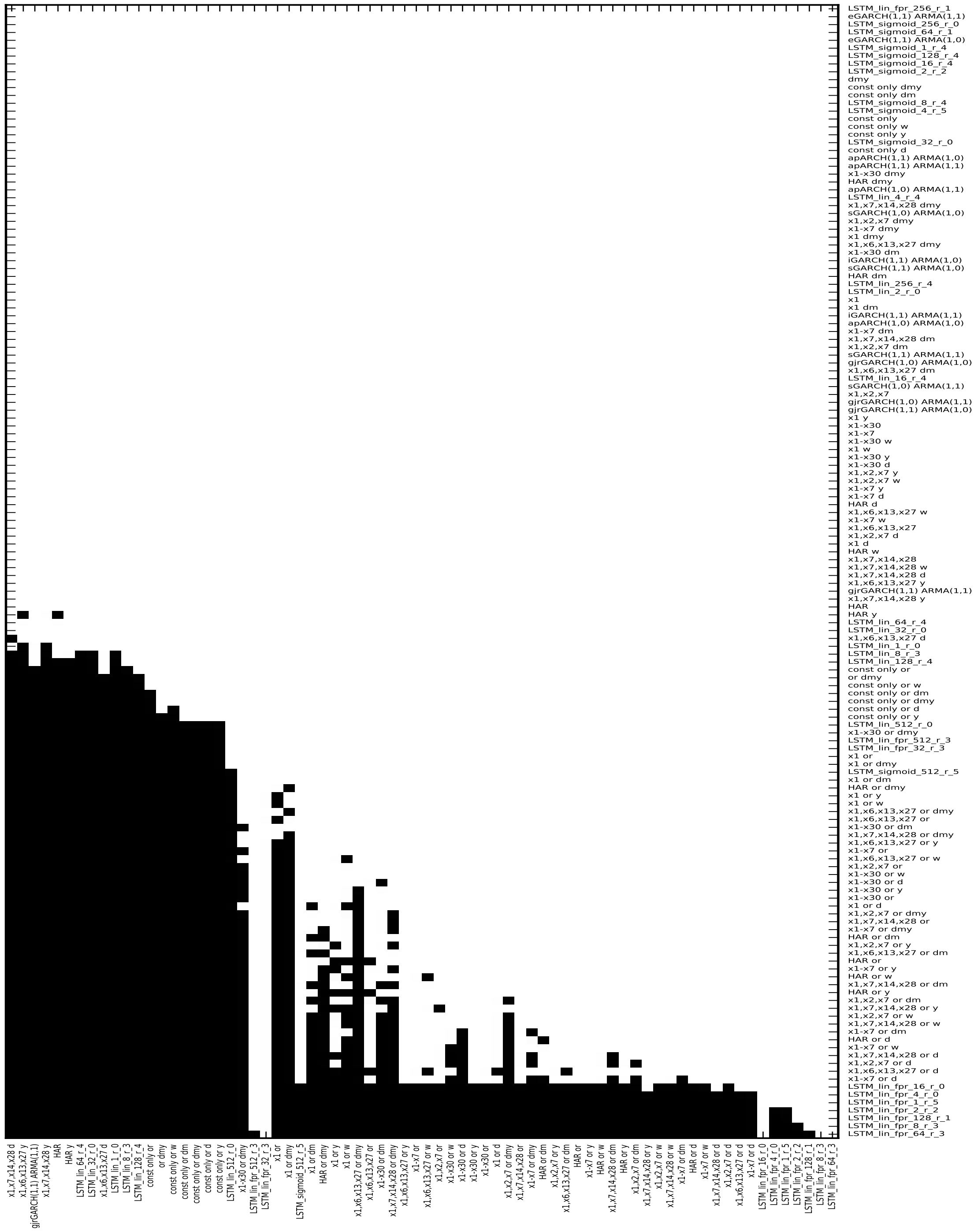


Figure C.4: Continuation of Figure C.3

Appendix D

Results for all models

Table D.1: Comparison of performances of all tested models. Models are sorted by average rank.

Model name	RMSE	MAE	DM1	DM2	Pareto
LSTM_lin_fpr_64_r_3	12.36	2.70	140.0	84.0	1
LSTM_lin_fpr_8_r_3	13.65	2.92	138.0	79.0	2
LSTM_lin_fpr_128_r_1	14.32	3.15	137.0	68.0	3
LSTM_lin_fpr_16_r_0	11.37	2.60	131.0	8.0	0
x1-x7 or dm	14.92	3.55	103.0	56.0	5
x1,x6,x13,x27 or d	14.96	3.53	108.0	45.0	5
HAR or d	14.94	3.55	104.0	47.0	5
x1,x2,x7 or dm	14.92	3.56	101.0	53.0	6
HAR or dm	14.90	3.57	98.0	64.0	5
x1-x7 or d	14.97	3.53	110.0	41.0	5
x1,x6,x13,x27 or dm	14.90	3.57	98.0	64.0	5
HAR	12.61	4.68	51.0	93.0	2
HAR w	12.56	4.73	42.0	96.0	2
HAR y	12.65	4.62	55.0	93.0	2
LSTM_lin_fpr_2_r_2	14.89	3.27	137.0	30.0	4
x1,x7,x14,x28 or dm	14.95	3.57	100.0	49.0	7
x1,x2,x7 or d	14.98	3.54	107.0	38.0	6
LSTM_lin_fpr_512_r_3	12.86	3.30	91.0	32.0	2
x1 or d	14.92	3.57	97.0	50.0	6
x1,x6,x13,x27 d	12.69	4.54	59.0	82.0	2
HAR or w	14.97	3.56	99.0	47.0	6
x1,x7,x14,x28 or d	15.01	3.53	107.0	35.0	6
x1-x7 or w	15.01	3.54	104.0	37.0	7
x1 d	12.58	4.71	41.0	83.0	2

Continued on next page

Table D.1 – *Continued from previous page*

Model name	RMSE	MAE	DM1	DM2	Pareto
x1 or dm	14.87	3.61	92.0	67.0	4
HAR d	12.60	4.74	41.0	85.0	3
x1,x6,x13,x27 y	12.80	4.78	45.0	91.0	4
x1-x7 or dmy	14.96	3.58	98.0	46.0	8
HAR or dmy	14.92	3.61	93.0	59.0	7
x1,x6,x13,x27 w	12.68	4.85	41.0	89.0	4
x1,x6,x13,x27	12.72	4.89	41.0	93.0	5
x1,x2,x7 or w	15.02	3.55	102.0	33.0	8
x1-x30 or dm	14.94	3.60	93.0	48.0	7
x1,x2,x7 or dmy	14.96	3.59	97.0	43.0	9
x1 w	12.59	4.91	37.0	89.0	3
x1,x7,x14,x28 or w	15.05	3.55	102.0	31.0	8
x1,x7,x14,x28 d	12.94	4.72	42.0	57.0	3
HAR or	15.02	3.56	99.0	33.0	8
x1,x6,x13,x27 or dmy	14.95	3.63	93.0	46.0	8
x1,x6,x13,x27 dm	12.49	5.26	25.0	94.0	2
x1 or dmy	14.94	3.65	91.0	48.0	8
x1,x6,x13,x27 or w	14.98	3.59	95.0	39.0	9
x1-x30 or d	14.99	3.58	96.0	37.0	9
x1 or w	14.96	3.62	93.0	42.0	8
x1,x7,x14,x28 y	13.24	4.78	48.0	65.0	5
HAR or y	15.04	3.55	100.0	30.0	9
x1,x2,x7 d	12.87	4.75	41.0	49.0	4
x1 y	12.76	4.96	33.0	91.0	6
x1,x7,x14,x28	13.19	4.85	42.0	65.0	7
x1,x7,x14,x28 or dmy	14.99	3.61	93.0	39.0	10
x1	12.65	5.13	19.0	94.0	4
x1-x30 or w	15.02	3.58	95.0	35.0	9
HAR dm	12.42	5.33	18.0	93.0	2
x1 dm	12.39	5.34	19.0	92.0	2
LSTM_lin_fpr_32_r_3	13.19	3.02	91.0	3.0	2
x1-x30 or dmy	14.97	3.63	91.0	43.0	10
x1,x7,x14,x28 or y	15.12	3.54	101.0	26.0	8
x1,x7,x14,x28 w	13.03	4.80	42.0	46.0	6
x1-x30 or	15.05	3.56	96.0	31.0	10
x1-x7 or y	15.11	3.55	99.0	26.0	9
x1,x6,x13,x27 or	15.03	3.60	93.0	35.0	10
x1,x7,x14,x28 or	15.09	3.55	97.0	28.0	10
gjrGARCH(1,1) ARMA(1,1)	13.33	4.59	47.0	37.0	3

Continued on next page

Table D.1 – *Continued from previous page*

Model name	**RMSE**	**MAE**	**DM1**	**DM2**	**Pareto**
x1,x2,x7 or y	15.12	3.55	98.0	26.0	11
HAR dmy	12.45	5.49	13.0	97.0	3
x1,x6,x13,x27 dmy	12.54	5.41	16.0	86.0	3
x1-x7 d	12.90	4.74	40.0	36.0	3
LSTM_lin_fpr_4_r_0	15.41	3.36	135.0	12.0	5
x1,x2,x7 dm	12.69	5.29	23.0	66.0	5
x1-x30 or y	15.08	3.57	96.0	28.0	11
x1-x7 or	15.08	3.56	95.0	29.0	10
x1,x7,x14,x28 dm	12.72	5.30	23.0	67.0	6
x1 dmy	12.46	5.49	15.0	86.0	3
x1-x7 w	12.97	4.79	41.0	34.0	5
x1,x2,x7 or	15.09	3.57	95.0	28.0	12
LSTM_lin_fpr_1_r_5	15.61	3.40	135.0	9.0	6
x1 or	15.01	3.67	91.0	33.0	11
x1,x6,x13,x27 or y	15.08	3.58	94.0	28.0	12
x1,x2,x7 w	13.00	4.86	39.0	35.0	6
x1,x2,x7 y	13.24	4.93	38.0	44.0	9
x1,x2,x7 dmy	12.72	5.45	15.0	77.0	7
x1-x30 d	13.02	4.83	38.0	33.0	6
x1,x7,x14,x28 dmy	12.75	5.45	15.0	77.0	8
x1 or y	15.08	3.61	93.0	27.0	13
x1-x7 dm	12.72	5.30	21.0	43.0	7
x1,x2,x7	13.20	5.03	28.0	41.0	9
x1-x7 dmy	12.75	5.44	15.0	50.0	8
x1-x7 y	13.27	4.88	40.0	32.0	8
x1-x30 y	13.23	4.92	37.0	33.0	8
gjrGARCH(1,1) ARMA(1,0)	13.47	4.81	32.0	33.0	7
x1-x30	13.18	4.96	34.0	33.0	8
sGARCH(1,0) ARMA(1,1)	13.52	4.89	27.0	35.0	9
x1-x30 w	13.09	4.94	37.0	31.0	7
x1-x7	13.23	4.94	35.0	32.0	9
x1-x30 dm	12.84	5.42	16.0	34.0	8
const only or dm	15.18	3.86	85.0	23.0	14
const only or dmy	15.24	3.86	85.0	23.0	15
LSTM_lin_128_r_4	15.12	4.03	80.0	25.0	14
LSTM_lin_64_r_4	15.05	4.37	57.0	23.0	12
const only or d	15.32	3.86	86.0	19.0	14
gjrGARCH(1,0) ARMA(1,1)	13.71	4.80	31.0	17.0	6
sGARCH(1,1) ARMA(1,1)	14.43	5.11	24.0	32.0	12

Continued on next page

Table D.1 – *Continued from previous page*

Model name	**RMSE**	**MAE**	**DM1**	**DM2**	**Pareto**
const only or w	15.37	3.92	84.0	18.0	16
x1-x30 dmy	12.86	5.56	13.0	34.0	9
const only or y	15.45	3.84	87.0	15.0	14
LSTM_lin_512_r_0	15.52	3.74	91.0	12.0	14
or dmy	15.39	3.91	84.0	15.0	16
const only or	15.40	3.95	82.0	17.0	17
iGARCH(1,1) ARMA(1,1)	14.50	5.20	20.0	30.0	13
LSTM_sigmoid_512_r_5	15.67	3.65	92.0	5.0	14
LSTM_lin_8_r_3	15.51	4.12	74.0	12.0	18
gjrGARCH(1,0) ARMA(1,0)	14.10	4.99	24.0	9.0	10
apARCH(1,0) ARMA(1,0)	14.28	5.09	21.0	7.0	11
LSTM_lin_1_r_0	15.70	4.24	62.0	5.0	19
LSTM_lin_32_r_0	15.81	4.29	58.0	3.0	20
sGARCH(1,0) ARMA(1,0)	14.37	5.45	15.0	13.0	12
apARCH(1,1) ARMA(1,1)	13.76	5.72	12.0	11.0	10
apARCH(1,0) ARMA(1,1)	14.07	5.69	13.0	8.0	10
apARCH(1,1) ARMA(1,0)	13.99	5.69	12.0	8.0	10
LSTM_lin_2_r_0	15.13	5.41	18.0	18.0	15
const only dm	14.65	6.45	3.0	10.0	14
const only dmy	14.72	6.53	3.0	10.0	15
LSTM_lin_16_r_4	15.68	4.91	26.0	6.0	19
sGARCH(1,1) ARMA(1,0)	15.31	5.53	17.0	17.0	16
iGARCH(1,1) ARMA(1,0)	15.33	5.54	17.0	16.0	17
LSTM_lin_256_r_4	15.65	5.44	18.0	6.0	19
const only	15.26	6.40	4.0	15.0	17
const only w	15.24	6.24	6.0	8.0	16
LSTM_lin_4_r_4	15.49	5.81	14.0	8.0	18
const only d	15.28	6.00	11.0	7.0	16
const only y	15.42	6.09	9.0	10.0	19
dmy	15.16	6.45	3.0	8.0	16
LSTM_sigmoid_32_r_0	15.36	5.96	9.0	7.0	18
LSTM_sigmoid_4_r_5	15.83	6.35	4.0	3.0	21
LSTM_sigmoid_8_r_4	15.92	6.53	3.0	2.0	22
LSTM_sigmoid_16_r_4	16.02	6.72	2.0	2.0	23
LSTM_sigmoid_2_r_2	16.12	6.88	2.0	2.0	24
LSTM_sigmoid_128_r_4	16.26	6.72	2.0	1.0	23
LSTM_sigmoid_1_r_4	16.12	7.15	1.0	1.0	24
eGARCH(1,1) ARMA(1,0)	19.89	6.47	1.0	0.0	22
LSTM_sigmoid_64_r_1	16.45	7.39	1.0	0.0	25

Continued on next page

Table D.1 – *Continued from previous page*

Model name	RMSE	MAE	DM1	DM2	Pareto
eGARCH(1,1) ARMA(1,1)	38.60	6.96	0.0	0.0	25
LSTM_sigmoid_256_r_0	16.78	8.43	0.0	0.0	26
LSTM_lin_fpr_256_r_1	43.04	7.85	0.0	0.0	26

Table D.2: Comparison of performances of all tested models. Models are sorted by average rank.

Model name	RMSE	MAE	DM1	DM2	Pareto	Average
LSTM_lin_fpr_64_r_3	2.0	2.0	1.0	17.0	1	5.5
LSTM_lin_fpr_8_r_3	50.0	3.0	2.0	20.0	2	18.75
LSTM_lin_fpr_128_r_1	57.0	5.0	3.5	23.0	3	22.125
LSTM_lin_fpr_16_r_0	1.0	1.0	7.0	121.5	0	32.625
x1-x7 or dm	69.0	23.0	14.0	33.0	5	34.75
x1,x6,x13,x27 or d	77.0	12.0	9.0	46.0	5	36.0
HAR or d	73.0	18.0	12.5	41.5	5	36.25
x1,x2,x7 or dm	70.0	26.0	17.5	34.0	6	36.875
HAR or dm	66.0	31.0	25.5	29.5	5	38.0
x1-x7 or d	82.0	10.0	8.0	52.5	5	38.125
x1,x6,x13,x27 or dm	65.0	33.0	25.5	29.5	5	38.25
HAR	13.0	69.0	68.0	6.5	2	39.125
HAR w	9.0	72.0	73.5	2.0	2	39.125
HAR y	15.0	68.0	67.0	6.5	2	39.125
LSTM_lin_fpr_2_r_2	64.0	6.0	3.5	84.0	4	39.375
x1,x7,x14,x28 or dm	74.0	29.0	19.5	37.5	7	40.0
x1,x2,x7 or d	83.0	13.0	10.5	56.0	6	40.625
LSTM_lin_fpr_512_r_3	28.0	7.0	50.5	77.5	2	40.75
x1 or d	67.0	34.0	29.0	35.5	6	41.375
x1,x6,x13,x27 d	18.0	66.0	64.0	19.0	2	41.75
HAR or w	81.0	24.0	22.0	41.5	6	42.125
x1,x7,x14,x28 or d	87.0	11.0	10.5	63.0	6	42.875
x1-x7 or w	89.0	14.0	12.5	58.0	7	43.375
x1 d	10.0	70.0	78.5	18.0	2	44.125
x1 or dm	63.0	44.0	46.5	24.5	4	44.5
HAR d	12.0	74.0	78.5	16.0	3	45.125
x1,x6,x13,x27 y	26.0	76.0	71.0	10.5	4	45.875
x1-x7 or dmy	78.0	36.0	25.5	44.0	8	45.875
HAR or dmy	68.0	45.0	42.0	31.0	7	46.5

Continued on next page

Table D.2 – *Continued from previous page*

Model name	RMSE	MAE	DM1	DM2	Pareto	Average
x1,x6,x13,x27 w	16.0	83.0	78.5	12.5	4	47.5
x1,x6,x13,x27	21.0	87.0	78.5	6.5	5	48.25
x1,x2,x7 or w	91.0	17.0	15.5	72.0	8	48.875
x1-x30 or dm	72.0	42.0	42.0	39.5	7	48.875
x1,x2,x7 or dmy	79.0	40.0	29.0	49.0	9	49.25
x1 w	11.0	89.0	88.0	12.5	3	50.125
x1,x7,x14,x28 or w	95.0	16.0	15.5	81.0	8	51.875
x1,x7,x14,x28 d	32.0	71.0	73.5	32.0	3	52.125
HAR or	90.0	25.0	22.0	72.0	8	52.25
x1,x6,x13,x27 or dmy	75.0	48.0	42.0	44.0	8	52.25
x1,x6,x13,x27 dm	7.0	103.0	98.0	3.5	2	52.875
x1 or dmy	71.0	51.0	50.5	39.5	8	53.0
x1,x6,x13,x27 or w	84.0	39.0	35.5	54.5	9	53.25
x1-x30 or d	86.0	38.0	32.0	58.0	9	53.5
x1 or w	76.0	47.0	42.0	51.0	8	54.0
x1,x7,x14,x28 y	44.0	77.0	69.0	27.5	5	54.375
HAR or y	94.0	20.0	19.5	84.0	9	54.375
x1,x2,x7 d	30.0	75.0	78.5	37.5	4	55.25
x1 y	25.0	96.0	92.0	10.5	6	55.875
x1,x7,x14,x28	39.0	84.0	73.5	27.5	7	56.0
x1,x7,x14,x28 or dmy	85.0	43.0	42.0	54.5	10	56.125
x1	14.0	101.0	106.5	3.5	4	56.25
x1-x30 or w	92.0	35.0	35.5	63.0	9	56.375
HAR dm	4.0	107.0	109.0	6.5	2	56.625
x1 dm	3.0	108.0	106.5	9.0	2	56.625
LSTM_lin_fpr_32_r_3	40.0	4.0	50.5	133.0	2	56.875
x1-x30 or dmy	80.0	49.0	50.5	49.0	10	57.125
x1,x7,x14,x28 or y	107.0	15.0	17.5	93.0	8	58.125
x1,x7,x14,x28 w	36.0	80.0	73.5	44.0	6	58.375
x1-x30 or	97.0	28.0	32.0	81.0	10	59.5
x1-x7 or y	104.0	19.0	22.0	93.0	9	59.5
x1,x6,x13,x27 or	93.0	41.0	42.0	63.0	10	59.75
x1,x7,x14,x28 or	103.0	21.0	29.0	88.5	10	60.375
gjrGARCH(1,1) ARMA(1,1)	47.0	67.0	70.0	58.0	3	60.5
x1,x2,x7 or y	105.0	22.0	25.5	93.0	11	61.375
HAR dmy	5.0	118.0	122.0	1.0	3	61.5
x1,x6,x13,x27 dmy	8.0	110.0	113.5	14.5	3	61.5
x1-x7 d	31.0	73.0	82.5	60.0	3	61.625
LSTM_lin_fpr_4_r_0	122.0	8.0	5.5	111.0	5	61.625

Continued on next page

Table D.2 – *Continued from previous page*

Model name	**RMSE**	**MAE**	**DM1**	**DM2**	**Pareto**	**Average**
x1,x2,x7 dm	17.0	104.0	101.5	26.0	5	62.125
x1-x30 or y	98.0	30.0	32.0	88.5	11	62.125
x1-x7 or	100.0	27.0	35.5	86.0	10	62.125
x1,x7,x14,x28 dm	19.0	105.0	101.5	24.5	6	62.5
x1 dmy	6.0	117.0	117.0	14.5	3	63.625
x1-x7 w	33.0	78.0	78.5	67.0	5	64.125
x1,x2,x7 or	102.0	32.0	35.5	88.5	12	64.5
LSTM_lin_fpr_1_r_5	128.0	9.0	5.5	117.5	6	65.0
x1 or	88.0	52.0	50.5	72.0	11	65.625
x1,x6,x13,x27 or y	99.0	37.0	38.0	88.5	12	65.625
x1,x2,x7 w	34.0	85.0	84.0	63.0	6	66.5
x1,x2,x7 y	45.0	92.0	85.5	47.0	9	67.375
x1,x2,x7 dmy	20.0	115.0	117.0	21.5	7	68.375
x1-x30 d	35.0	82.0	85.5	72.0	6	68.625
x1,x7,x14,x28 dmy	23.0	114.0	117.0	21.5	8	68.875
x1 or y	101.0	46.0	42.0	91.0	13	70.0
x1-x7 dm	22.0	106.0	103.5	49.0	7	70.125
x1,x2,x7	41.0	98.0	95.0	52.5	9	71.625
x1-x7 dmy	24.0	113.0	117.0	35.5	8	72.375
x1-x7 y	46.0	86.0	82.5	77.5	8	73.0
x1-x30 y	42.0	91.0	88.0	72.0	8	73.25
gjrGARCH(1,1) ARMA(1,0)	48.0	81.0	93.0	72.0	7	73.5
x1-x30	38.0	95.0	91.0	72.0	8	74.0
sGARCH(1,0) ARMA(1,1)	49.0	88.0	96.0	63.0	9	74.0
x1-x30 w	37.0	93.0	88.0	81.0	7	74.75
x1-x7	43.0	94.0	90.0	77.5	9	76.125
x1-x30 dm	27.0	111.0	113.5	67.0	8	79.625
const only or dm	110.0	56.0	56.5	97.0	14	79.875
const only or dmy	111.0	57.0	56.5	97.0	15	80.375
LSTM_lin_128_r_4	106.0	61.0	61.0	95.0	14	80.75
LSTM_lin_64_r_4	96.0	65.0	66.0	97.0	12	81.0
const only or d	116.0	55.0	55.0	99.0	14	81.25
gjrGARCH(1,0) ARMA(1,1)	51.0	79.0	94.0	103.0	6	81.75
sGARCH(1,1) ARMA(1,1)	59.0	100.0	99.5	77.5	12	84.0
const only or w	119.0	59.0	58.5	100.5	16	84.25
x1-x30 dmy	29.0	121.0	122.0	67.0	9	84.75
const only or y	124.0	54.0	54.0	107.0	14	84.75
LSTM_lin_512_r_0	127.0	53.0	50.5	111.0	14	85.375
or dmy	120.0	58.0	58.5	107.0	16	85.875

Continued on next page

Table D.2 – *Continued from previous page*

Model name	RMSE	MAE	DM1	DM2	Pareto	Average
const only or	121.0	60.0	60.0	103.0	17	86.0
iGARCH(1,1) ARMA(1,1)	60.0	102.0	105.0	84.0	13	87.75
LSTM_sigmoid_512_r_5	130.0	50.0	46.5	130.5	14	89.25
LSTM_lin_8_r_3	126.0	62.0	62.0	111.0	18	90.25
gjrGARCH(1,0) ARMA(1,0)	55.0	97.0	99.5	117.5	10	92.25
apARCH(1,0) ARMA(1,0)	56.0	99.0	103.5	126.0	11	96.125
LSTM_lin_1_r_0	132.0	63.0	63.0	130.5	19	97.125
LSTM_lin_32_r_0	133.0	64.0	65.0	133.0	20	98.75
sGARCH(1,0) ARMA(1,0)	58.0	116.0	117.0	109.0	12	100.0
apARCH(1,1) ARMA(1,1)	52.0	124.0	124.5	113.0	10	103.375
apARCH(1,0) ARMA(1,1)	54.0	122.0	122.0	121.5	10	104.875
apARCH(1,1) ARMA(1,0)	53.0	123.0	124.5	121.5	10	105.5
LSTM_lin_2_r_0	108.0	109.0	109.0	100.5	15	106.625
const only dm	61.0	133.0	133.5	115.0	14	110.625
const only dmy	62.0	135.0	133.5	115.0	15	111.375
LSTM_lin_16_r_4	131.0	90.0	97.0	128.5	19	111.625
sGARCH(1,1) ARMA(1,0)	115.0	119.0	111.5	103.0	16	112.125
iGARCH(1,1) ARMA(1,0)	117.0	120.0	111.5	105.0	17	113.375
LSTM_lin_256_r_4	129.0	112.0	109.0	128.5	19	119.625
const only	113.0	131.0	130.5	107.0	17	120.375
const only w	112.0	129.0	129.0	121.5	16	122.875
LSTM_lin_4_r_4	125.0	125.0	120.0	121.5	18	122.875
const only d	114.0	127.0	126.0	126.0	16	123.25
const only y	123.0	128.0	127.5	115.0	19	123.375
dmy	109.0	132.0	133.5	121.5	16	124.0
LSTM_sigmoid_32_r_0	118.0	126.0	127.5	126.0	18	124.375
LSTM_sigmoid_4_r_5	134.0	130.0	130.5	133.0	21	131.875
LSTM_sigmoid_8_r_4	135.0	136.0	133.5	136.0	22	135.125
LSTM_sigmoid_16_r_4	136.0	138.0	137.0	136.0	23	136.75
LSTM_sigmoid_2_r_2	138.0	139.0	137.0	136.0	24	137.5
LSTM_sigmoid_128_r_4	139.0	137.0	137.0	138.5	23	137.875
LSTM_sigmoid_1_r_4	137.0	141.0	140.0	138.5	24	139.125
eGARCH(1,1) ARMA(1,0)	142.0	134.0	140.0	142.0	22	139.5
LSTM_sigmoid_64_r_1	140.0	142.0	140.0	142.0	25	141.0
eGARCH(1,1) ARMA(1,1)	143.0	140.0	143.0	142.0	25	142.0
LSTM_sigmoid_256_r_0	141.0	144.0	143.0	142.0	26	142.5
LSTM_lin_fpr_256_r_1	144.0	143.0	143.0	142.0	26	143.0

Appendix E

Performance of selected models by individual criteria.

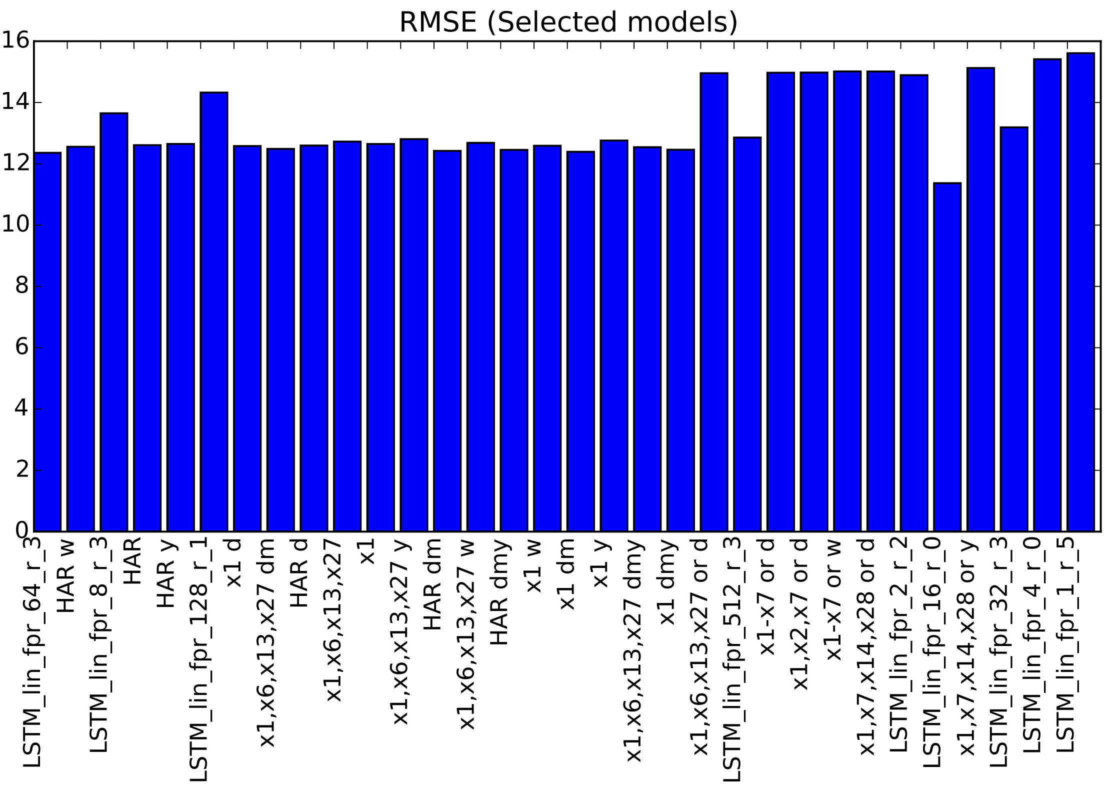


Figure E.1: Evaluation of selected models by RMSE — model was selected if it had at least one rank less or equal 15.

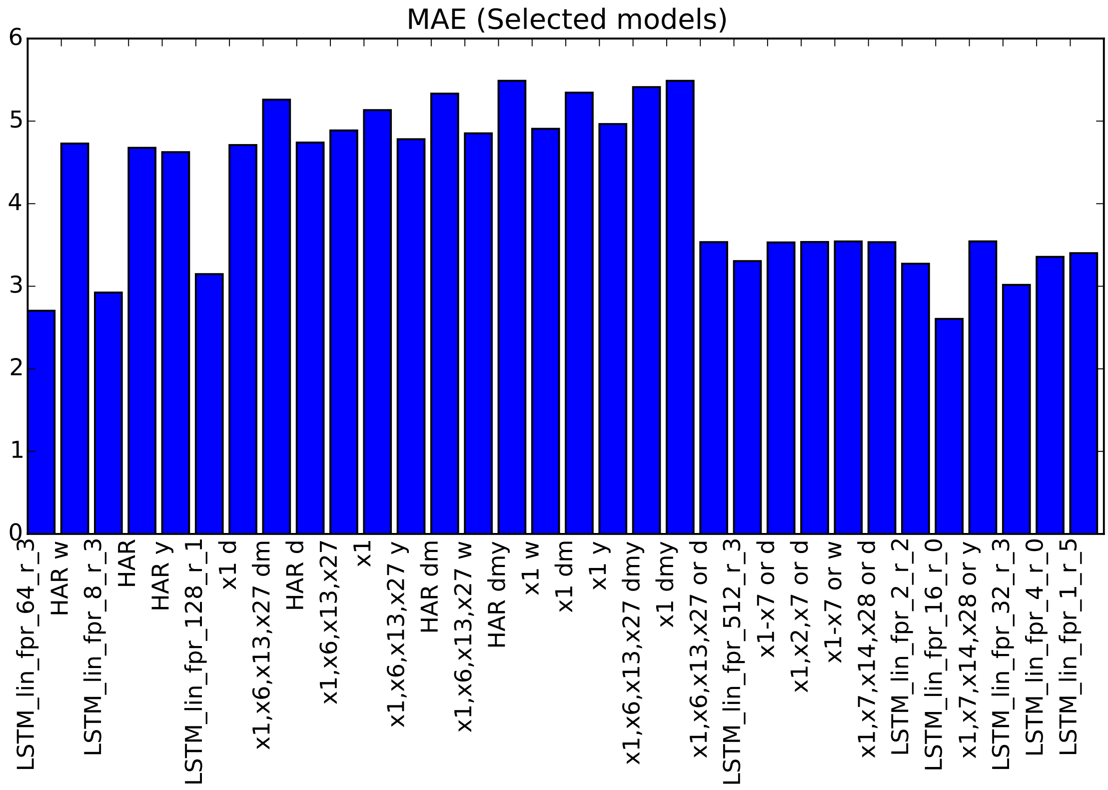


Figure E.2: Evaluation of selected models by MAE — model was selected if it had at least one rank less or equal 15.

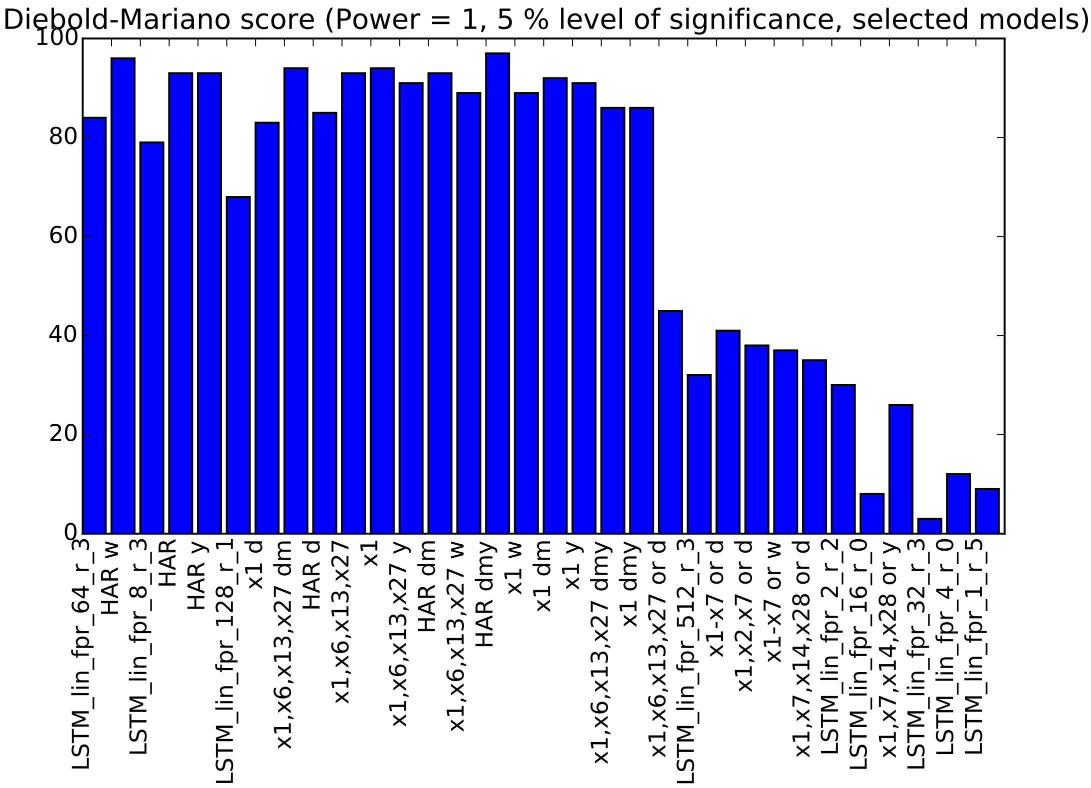


Figure E.3: Evaluation of selected models by Diebold-Mariano score with $p = 1$ and 5 % level of significance — model was selected if it had at least one rank less or equal 15.

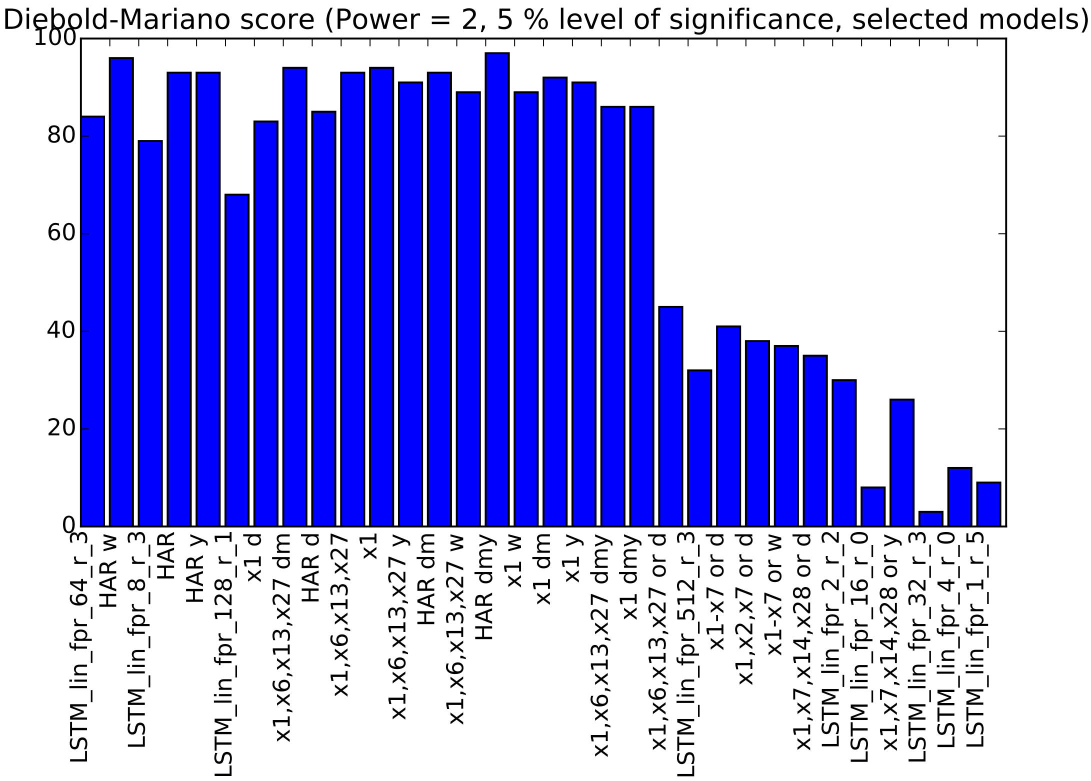


Figure E.4: Evaluation of selected models by Diebold-Mariano score with $p = 2$ and 5 % level of significance — model was selected if it had at least one rank less or equal 15.

Appendix F

Content of the DVD

The attached DVD contains the electronic version of this work, all graphs from this work, notebooks with the code and original data.

- `/Notebooks/` contains IPython notebooks with all used codes. Modify paths before use.
  - `/Notebooks/IES_Electricity_Volatility_Estimation-Model_comparison.ipynb` contains the code for model comparison, before use, all forecasts have to be saved in Pickle files.
  - `/Notebooks/IES_Electricity_Data_Visualisation_and_ANN_experiments.ipynb`
  - `/Notebooks/IES_Electricity_ANN_Experiments_visualisation.ipynb`
  - `/Notebooks/IES_Electricity_Volatility_Estimation.ipynb`
  - `/Notebooks/IES_Electricity_Volatility_Estimation-GARCH.ipynb`
  - `/Notebooks/IES_Electricity_Volatility_Estimation-Linear_regression.ipynb`
  - `/Notebooks/IES_Electricity_Volatility_Estimation-Sigmoid.ipynb`
  - `/Notebooks/IES_Electricity_Volatility_Forecast_from_Price_Prediction.ipynb`
- `/Data/` contains datasets
- `/Images/` contains all graphs and figures from this work
- `/Pickles/` contains saved model and forecast in Python Pickle containers
- `/Kunc_Vladimir_BP_2015.pdf` the electronic version of this work