CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Transportation Sciences

Department of Applied Mathematics

POWER PURCHASE STRATEGY OF RETAIL CUSTOMERS UTILIZING ADVANCED CLASSIFICATION METHODS

Doctoral Thesis

May 2022 Lenka Jonáková

Candidate:	Ing. Lenka Jonáková
Study programme:	Engineering Informatics
Field of study:	Engineering Informatics in Transportation and Telecommunications
Contact:	Czech Technical University in Prague Faculty of Transportation Sciences Department of Applied Mathematics Na Florenci 25, 110 00, Prague 1
Email:	jonaklen@cvut.cz
Supervisor:	doc. Ing. Ivan Nagy, CSc.
Contact:	Czech Technical University in Prague Faculty of Transportation Sciences Department of Applied Mathematics Na Florenci 25, 110 00, Prague 1

STATEMENT OF THE THESIS ORIGINALITY

I honestly declare that this doctoral thesis *Power Purchase Strategy of Retail Customers Utilizing Advanced Classification Methods* was written only by me under the professional guidance of my supervisor and with the use of literature and other information sources, which are all cited in the text and listed at the end of the thesis.

May 25, 2022

……

Ing. Lenka Jonáková

ACKNOWLEDGEMENTS

I would like to sincerely thank my supervisor doc. Ing. Ivan Nagy, CSc. for his immense support and expert advice, which he has been very kindly providing to me since my bachelor studies. His human, tolerant, and caring approach towards students is highly exceptional, inspiring, and admirable, and I was truly privileged to have the opportunity to work under his supervision. His authority in the area of mathematical modelling indeed significantly changed not only the course of my studies, but also my professional life.

I would also like to express my greatest gratitude to my family and my partner for their boundless understanding and support throughout my studies. Their encouragement and affection gave me the extra push and strength I needed on my journey.

ABSTRACT

This thesis reflects a unique task with significant business potential, on the edge of the wholesale and retail power market, i.e., progressive purchase of power derivatives by retail customers. The main emphasis is on the estimation of the oversold and overbought market utilizing various classification methods, and subsequent simulation of the progressive power purchase. For this purpose, the Czech power baseload yearly futures are used as a reference contract. Continuous price fixing, which is a very popular and commonly used strategy ensuring an average profit-loss result, is used as a benchmark to evaluate benefits of the investigated methods.

Due to the significant lack of publications in this area, the main contribution of this thesis is the comprehensive examination of methods in the context of the task, the thorough comparison and evaluation of their benefits, and the proposal of the most suitable solution. Ten well-established techniques are exploited for the purposes of data classification, namely, relative strength index, k-nearest neighbor, naive Bayes, support vector classifier, random forest, AdaBoost, 1-, 2- and 3-layer feed forward neural network, and long short-term memory.

Even though all the examined models exceeded the defined benchmark, long short-term memory proved its exceptional qualities among the other methods in terms of consistent prediction performance and generalization abilities. Nevertheless, its weaknesses such as high requirements for programming capacity, long training time, sensitivity to initialization of parameters as well as limited possibility of results interpretation should be taken into account. As a result, a solution combining low maintenance and simplicity of relative strength index and high accuracy of long short-term memory was proposed to make the price fixing procedure more practical and efficient. Considering an average auctioned volume in the order of tens of thousands of MWhs, the estimated average savings when employing the proposed solution are estimated to reach value in the order of tens to hundreds of thousands of EUR per one auction in comparison to the defined benchmark.

Key words: Czech power futures, retail market, progressive purchase, technical analysis, k-nearest neighbor, naive Bayes, support vector classifier, ensemble methods, neural network

ABSTRAKT

Tato disertační práce se zaměřuje na úlohu s významným obchodním potenciálem, která je definována na rozhraní velkoobchodního a maloobchodního trhu s elektřinou. Jedná se o postupný nákup dlouhodobých kontraktů na dodávku elektřiny koncovými zákazníky. Hlavní důraz je kladen na odhad přeprodaného a překoupeného trhu s využitím různých klasifikačních metod a následnou simulaci postupného nákupu. Jako reference je použit roční kontrakt na dodávku elektřiny v základním pásmu v České republice. Analyzované metody jsou porovnány s již existující, velmi populární a hojně využívanou strategií nákupu, která naceňuje daný kontrakt dle průměru závěrečných cen.

Vzhledem k nedostatečnému množství publikací adresujících tuto problematiku je hlavním přínosem této práce podrobná analýza metod v kontextu specifikované úlohy, posouzení a porovnání jejich přínosů, a návrh vhodného řešení. Pro účely klasifikace dat je využito deset etablovaných technik; jedná se o index relativní síly, algoritmus k-nejbližších sousedů, naivní Bayes, metoda podpůrných vektorů, náhodný les, AdaBoost, 1-, 2- a 3-vrstvá dopředná neuronová síť a long short-term memory.

Přestože všechny zkoumané modely dosáhly lepšího výsledku oproti strategii využívající průměru závěrečných cen, long short-term memory prokázala v porovnání s ostatními metodami zvláště výjimečné kvality, především z hlediska konzistence přesnosti predikce a generalizačních schopností. Je však třeba uvážit také slabiny tohoto přístupu, jako jsou například vysoké požadavky na výpočetní výkon systému, pomalé učení modelu, citlivost na inicializaci parametrů, stejně tak jako obtížná interpretace výsledků. Z důvodu zachování co největší praktičnosti a efektivity řešení byl navržen přístup kombinující nenáročný provoz a jednoduchost výpočtu indexu relativní síly a značnou přesnost algoritmu long short-term memory. Předpokládáme-li průměrný poptávaný objem v řádu desítek tisíc MWh, odhadované průměrné úspory při použití navržené metody se pohybují v řádu desítek až stovek tisíc EUR na jednu aukci oproti běžně využívané strategii nákupu na základě průměru závěrečných cen.

Klíčová slova: české energetické futures, maloobchodní trh, postupný nákup, technická analýza, k-nejbližších sousedů, naivní Bayes, metoda podpůrných vektorů, náhodný les, AdaBoost, neuronová síť

LIST OF ACRONYMS

AdaBoost	Adaptive Boosting
Adam	Adaptive Learning Rate Optimization Algorithm
ANN	Artificial Neural Networks
Bagging	Bootstrap Aggregating
Bbl	Barrel of Crude Oil
CEGH	Central European Gas Hub AG
CDS	Clean Dark Spread
CLS	Clean Lignite Spread
CNN	Convolutional Neural Network
CSS	Clean Spark Spread
CZ VTP	Czech Virtual Trading Point
ECX	European Climate Exchange
EU ETS	European Union Emissions Trading System
EUA	European Union Allowance
GRU	Gated Recurrent Unit
HHV	High Heating Value
ICE	The Intercontinental Exchange
KNN	K-Nearest Neighbor
LR	Learning Rate
LSTM	Long Short-term Memory
MA	Moving Average
MAE	Mean Absolute Error
MSCI	Morgan Stanley Capital International
MSE	Mean Square Error
mtCO2	One Metric Ton of Carbon Dioxide or Carbon-equivalent Greenhouse Gas
NN	Neural Network
PCA	Principal Component Analysis
PXE	Power Exchange Central Europe
RBF	Radial Basis Function
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RSI	Relative Strength Index
S&P	Standard and Poor's 500
SO	Stochastic Oscillator
SVC	Support Vector Classifier
SVM	Support Vector Machine
TRPC Coal API2	Rotterdam Coal Futures
TTF	Title Transfer Facility
YTD	Year To Date

CONTENT

1 INTRODUCTION

At the beginning of the nineties, the European energy sector went through a period of deregulation within which the government monopolies were eliminated. In contrast to the prior arrangement, in which the power producers also assumed the role of suppliers, liberalization enabled the entry of other subjects into the market. The sector became attractive to smaller power producers as well as to traders, who filled the blank space in the supplier chain. The increase in competition has been accompanied not only by the utilization of new technologies and the decrease in price, but also by the development of the power derivatives market [1].

The power market can be divided into wholesale and retail markets. The wholesale market is intended exclusively for power producers and traders, not for end-consumers. Therefore, the trading is exempt from any taxation as well as from any state-regulated fees. On the contrary, the main purpose of the retail market is the power supply to the end-consumers, and the state-regulation is applied here. Despite the considerable differences, retail prices can be derived from wholesale prices to a great extent [1].

This thesis reflects a unique task with significant business potential, on the edge of the wholesale and retail market, that is, purchase of power derivatives by retail customers. Due to the increased demand for the complexity of services from retail consumers, suppliers started to incorporate a specific requirement for progressive purchase into the bilateral power delivery agreements. This mechanism enables end-consumers to buy the demanded volume in many tranches for a price which is derived directly from the wholesale price, and, in this way, to diversify the price risk. Some of the consumers take a step further and use this opportunity to speculate on the development of wholesale prices. The so-called progressive purchase, in different forms, is becoming increasingly popular in Central Europe. Consequently, this methodology was also adopted by some of the regulated exchange platforms in the region, such as Power Exchange Central Europe, a.s., (PXE) [2] and Czech Moravian Commodity Exchange Kladno (CMCEK) [3]. The popularity of the method can be documented in figures from PXE; approximately one quarter of all power consumers have chosen the progressive purchase approach during the last three years. It corresponds to 88 % of the total volume traded on the PXE power retail market, indicating the considerable desirability of this procedure among clients with high consumption [2].

In Western Europe, the tendency during the last years seemed to be heading more intensively toward digitalization initiatives, e.g., real-time management of smart grids, where supply and consumption are priced against the spot market. Although progressive purchase does not offer the same level of pricing efficiency, it is a publicly recognized and very easily implemented solution to risk diversification without any additional costs for hardware or software equipment. Therefore, the business potential of this approach is believed to be significant and worth further research.

Due to recent events, price risks in the European energy market have strongly escalated, and the key question is, how the system would be coping with potential further energy shortages, and more importantly, whether the situation would be manageable without significant regulatory measures. For the purposes of this study, we assume that the liberal market conditions are met, and the pricing mechanisms are fairly efficient.

2 DEFINITION OF THE TASK

For the purposes of the study, we will consider the following representative scenario: The retail customer demands a contract for a yearly electricity supply. Based on the delivery profile, the customer is offered a margin by the supplier defined in relative or absolute terms, i.e., the final price equals the margin multiplied or added to the wholesale price, respectively. Prior to contract confirmation, the customer can choose which wholesale contract will be used as a reference for price fixing. The customer has the possibility to purchase the demanded power volume in n tranches and can fix the price k-times in one day, i.e., he is able to fix the price for the k/n portion of the whole delivery in one day. The final price is equal to the average of all fixed prices. In case the end-customer does not fix the price in the predefined number of steps, the fixing proceeds automatically at the furthest possible date(s).

Even though the definition of the task as well as initial assumptions may seem highly complex, essentially, after the contract confirmation, the customer role is limited to providing supplier with purchase instructions and to speculate in this way on the wholesale market. Therefore, the main goal of this thesis can be simplified and narrowed down to the estimation of buying signals. An analysis will be exploited for the Czech power yearly baseload futures, with delivery in the front year, which are used by end customers as reference contracts most frequently.

It is important to emphasize that contrary to speculative power traders, who can flexibly increase or decrease their risk exposure by managing their open position, retail customers do not have such a possibility, and thus, improvement in the efficiency of estimating trading signals in this business area has a significant potential from the risk management as well as economic perspective.

Considering the input data are believed to include non-stationarity, non-linearity, and noise, price signals will be estimated with the use of different types of machine learning algorithms, i.e., one-, two- and three-layer feed-forward neural network with supervised learning, support vector classifier, random forest and AdaBoost. Assuming potential autocorrelation dependencies within the time-series, the long short-term memory neural network will be further exploited. Although these machine learning methods usually offer an exceptional performance in terms of prediction accuracy, the training process is slow and the interpretation of causal relationships within the models is very challenging. The threat of overfitting as well as of non-sufficient model robustness is thus more tangible. Therefore, also simpler techniques, such as k-nearest neighbor and Bayesian approach, specifically naive Bayes, which allow deeper model understanding, higher flexibility in terms of model adjustment as well as easier results interpretation, will also be used for the data classification. Furthermore, technical analysis will be utilized, specifically Relative Strength Index, which is well-established indicator among traders.

2.1 Case Study

To clarify some of the specifics of the defined task, a practical example of different hedging strategies for the progressive power purchase will be presented in this chapter.

Given a progressive purchase of a power supply, which is fixed against the wholesale reference yearly baseload contract with delivery in 2019 within one year before its delivery, let us assume the following three price fixing scenarios:

1. Optimal four-step price fixing
2. Evenly distributed four-step price fixing
3. Continuous price fixing (i.e., fixing against everyday settlement price)

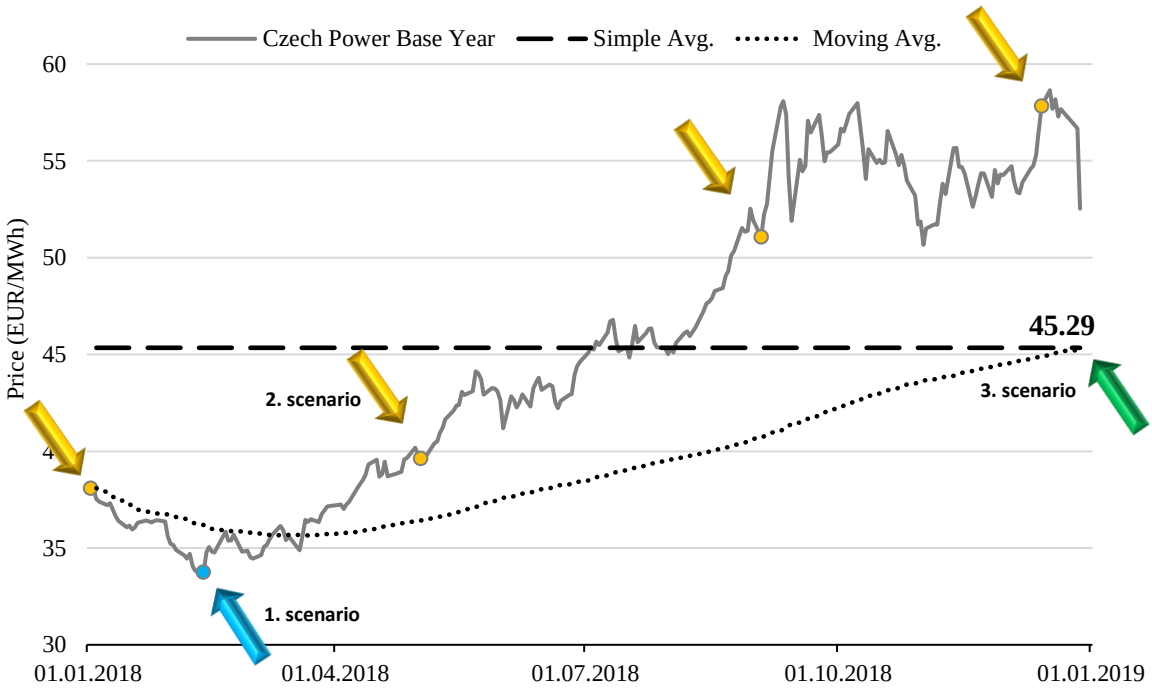

Figure 2.1: Price development of Czech power with baseload delivery in 2019 (case study of price fixing scenarios)

The optimal way to fix the price would be to proceed with all four fixing steps on 12.02.2018 for the yearly minimum price of 33.75 EUR/MWh. Given the 253 data samples and assuming uniform random selection of the buying signals, the probability of randomly choosing the optimal result is in the order of tenths of a percent.

Considering the distribution of trading days, an evenly distributed four-step fixing on 02.01.2018 (settlement price: 38.09 EUR/MWh), 02.05.2018 (settlement price: 39.62 EUR/MWh), 03.09.2018 (settlement price: 51.06 EUR/MWh) and on 14.12.2018 (settlement price: 57.82 EUR/MWh) would lead to the final price 46.65 EUR/MWh. This procedure presents a partial effect of price risk diversification.

Continuous price fixing can be represented as a simple cumulative moving average, that is, the average of all settlement prices available from the very beginning of the respective year. Fixing against the everyday settlement price provides the second-best result, i.e., 45.34 EUR/MWh. This approach represents a very popular method of price fixing, which ensures on average profit-loss result. Officials of cities, municipalities and other important subjects responsible for power purchase are often exposed to significant public pressure and do not want to take the responsibility for any estimation of buying signals. Therefore, risk diversification strategies and algorithms that can be easily automated, such as this one, seem to be highly demanded.

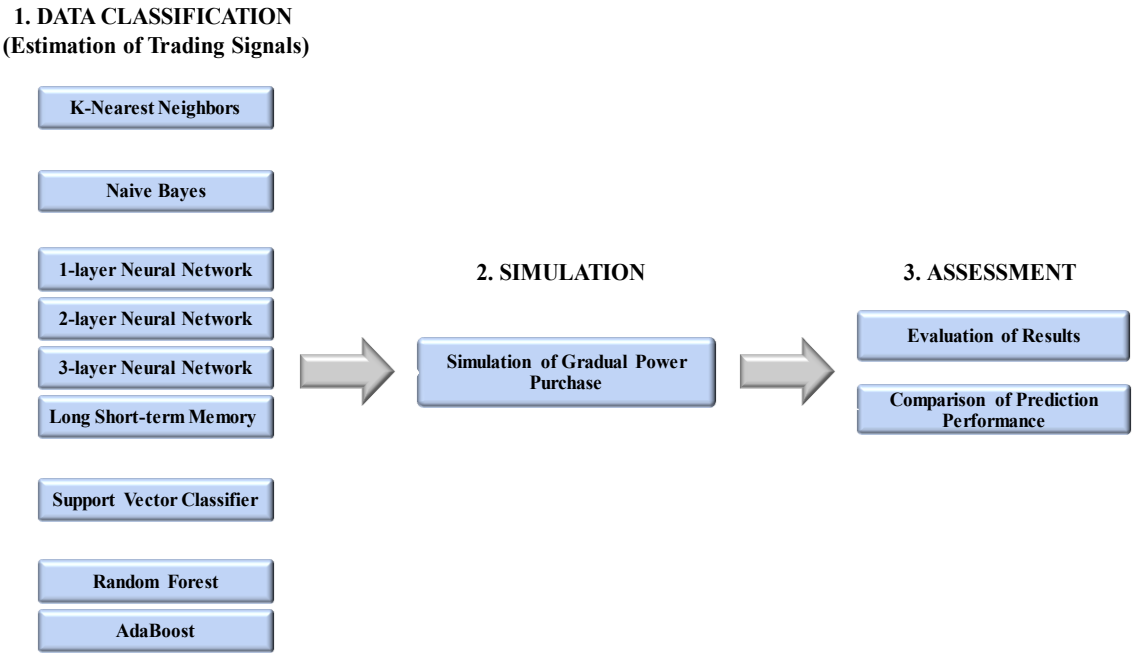

Figure 2.2: Graphical representation of workflow

2.2 Benchmark

The continuous price fixing presented in the previous chapter (see the third scenario), i.e., fixing against everyday settlement price, will be considered a benchmark for the purposes of further analysis and evaluation of the investigated methods.

2.3 Goal of the Thesis

The main goal of this dissertation thesis is to estimate oversold and overbought market conditions with use of various classification techniques in the context of the highly challenging task of hedging of power price risk by retail customers. The Czech power baseload yearly futures are used as the reference contract for this purpose.

For the purposes of data classification, ten well-established techniques are being exploited, namely relative strength index, k-nearest neighbor, naive Bayes, support vector classifier, random forest, AdaBoost, 1-, 2- and 3-layer feed forward neural network, and long short-term memory.

After the models' training and data classification, the predicted trading signals are utilized for the simulation of the progressive power purchase. Continuous price fixing, which is a very popular and commonly used method ensuring average profit-loss results, is used as a benchmark to evaluate the benefits of the exploited methods.

The prediction performance of the different models is thereafter compared and evaluated against the established benchmark. This step is perceived to have the largest practical impact, and therefore, is assumed to be the most important part of the thesis contribution. For further details, see Figure 2.2 that shows the completed workflow.

3 DATA

The data matrix consists of daily settlement prices from 02.01.2007 to 29.12.2021 and comprises 3904 samples. Within our research, the following fundamental as well as technical indicators will be examined.

3.1 Fundamental Data

- Price of the Czech Base Power Front Year (EUR/MWh)
  Power supply of 1 MW for a period of one year (delivery 24/7), with a place of delivery in the Czech Republic.
- Price of the TRPC Coal API2 Front Year (EUR/Tonne)
  European API2 thermal coal yearly futures.
- Price of the ICE Brent Front Month (EUR/Bbl)
  Monthly financial futures based on the ICE daily settlement price for Brent futures.
- Price of the TTF Gas Front Year (EUR/MWh)
  Yearly gas futures with physical delivery in a virtual trading point the Title Transfer Facility.

  Considering that the task focuses primarily on the Czech power market, the CEGH or CZ VTP gas price might seem more reasonable to be utilized. However, the liquidity in these hubs is much lower, and thus TTF contract is used instead as an approximation.
- Price of the ICE ECX EUA Front Year (EUR/Tonne)
  Entitlement to emit one tonne of carbon dioxide equivalent gas.
- Clean Spark Spread (EUR/MWh)
  Spark spread is a margin of a gas-fired power plant from selling a unit of electricity, which can be expressed as the difference between the cost of feedstock gas and the equivalent price of electricity on a High Heating Value (HHV) basis.

$$spark\ spread = baseload\ power\ price - gas\ price \div fuel\ efficiency \quad (3.1.1)$$

$$clean\ spark\ spread = spark\ spread - emissions\ price \cdot emissions\ intensity\ factor \div fuel\ efficiency \quad (3.1.2)$$

  Countries that are covered by the European Union Emissions Trading Scheme have to include into their financial balance also the cost of carbon dioxide emission allowances.

  For the purposes of this study, the emission intensity factor is considered 0.18404 mtCO2/MWh and gas plant efficiency is assumed to be 50 % HHV [4].

- Clean Dark Spread (EUR/MWh)
  Correspondingly to the spark spread, dark spread is defined as a difference between the cost of feedstock coal and the equivalent price of unit of electricity produced.

$$dark\ spread = baseload\ power\ price - coal\ price \div energy\ conversion\ factor \div fuel\ efficiency \quad (3.1.3)$$

$$clean\ dark\ spread = dark\ spread - emissions\ price \cdot emissions\ intensity\ factor \div fuel\ efficiency \quad (3.1.4)$$

  where coal-to-power energy conversion is 6.978, the emission intensity factor is assumed 0.34056 mtCO2/MWh, and coal plant efficiency is considered to be 35 % LHV [4].

- Clean Lignite Spread (EUR/MWh)
  Compared to natural gas and hard coal, lignite power production is the most emissions intensive. Assuming an average net thermal efficiency of 38% (efficiency varies in range of 34%-43%), lignite-fired power plant emits approximately 1093 gCO2/kWh (range 1221-966 respectively), which implies the emission intensity factor 0.4534 mtCO2/MWh. It is about 10% more of emission load than in case of hard coal and about three times more than in case of gas-fired power plant.

  The greatest part of variable costs of lignite power production is the cost of emission allowances. Therefore, price of lignite is usually neglected in the calculation of clean lignite spread [5].

$$clean\ lignite\ spread = baseload\ power\ price - emissions\ price \cdot emissions\ intensity\ factor \div fuel\ efficiency \quad (3.1.5)$$

- S&P Index (EUR)
  Stock market index of 500 of the largest publicly traded companies in the United States.

The specific contracts and trading platforms were selected with respect to their liquidity to ensure as efficient pricing procedures as possible. The fundamental data are further discussed in Chapter 4.

3.2 Technical Data

- Relative Strength Index (RSI)
- 14-day Moving Average
- 14-day Volatility
- Difference from the YTD Maximum Price
- Difference from the YTD Minimum Price

3.3 Data Pre-processing

Standardization was used as the data pre-processing technique in this study, during which the distribution of values of each feature is transformed so that its mean equals to zero and its standard deviation is one.

$$x' = \frac{x - \mu}{\sigma} \tag{3.3.1}$$

where μ is the mean and σ is the standard deviation of the training samples [6].

The statistics are estimated on samples in the training set and stored to be used later to transform the testing dataset during prediction.

Furthermore, a robust standardization was examined for the training of neural networks, which are known to be exceptionally sensitive to outliers. Robust standardization is very similar to the standard scaling mentioned above, but instead of mean and variance, it utilizes median and quartiles, specifically in range between 25^{th} and 75^{th} quantile. In this way, the scaler ignores the most distant data points [7]. However, robust standardization in this case did not prove to offer any additional benefits.

3.4 Output Specification

The price of the Czech base power is classified into ten categories by dividing the interval of all the settlement prices within the respective year into 10 equally large sections (1^{st} category representing very strong buy signal, 2^{nd} strong buy signal, …, 10^{th} being very strong sell signal), and is used in this form as the model output for the purposes of model training.

Today´s model output, i.e., estimated trading signal, encompasses information about the short-term condition in the market, and is derived from current values of the input variables. Contrary to a prediction of future absolute price values, prediction of the actual trading signal is believed to increase model robustness, while preserving an added value for a market participant in a form of trend indication, which allows to enter profitable trading position.

3.5 Source of Data

The Thomson Reuters Eikon software provided by Refinitiv, which is a platform designed for financial professionals aggregating different types of market information, was used as a primary source of the input data mentioned above. However, the data can also be aggregated from other, publicly available sources, mainly from the webpages of the relevant exchanges, such as EEX, ICE and Powernext. Moreover, other publicly available platforms, such as TradingView.com, can also be exploited.

4 DATA ANALYSIS

As anticipated, the prices of long-term contracts reflect the long-term market situation. They are mainly influenced by macro-economic events, infrastructure growth, which can be very difficult to quantify, and furthermore by the prices of power resources, which are present in the process of power production. Thus, it is important to identify the energy mix of power production in the relevant area.

According to the national energy mix of the Czech Republic, brown coal covers about 40% of the overall energy production. Therefore, the price of coal is one of the most important factors in the process of modelling power prices. As presented in Figure 4.1, the correlation between power and coal prices has been significant. Another variable closely related to the price of coal and considerably influencing power pricing is the price of emission allowances. The low price of emission allowances reduces the benefits of using less carbon-heavy technologies, and instead favours less expensive production from coal power plants [8]. Therefore, the power prices rise with the increase in price of emission allowances, and vice versa. The share of renewable energy resources on the total power production significantly differs from year to year, not only because of changes in weather fundamentals, but also due to high investments in this sector, and the abrupt development of new solar and wind power farms. In recent years, production from renewable power sources has covered from 4% to 12% of the overall energy production of the Czech Republic [9].

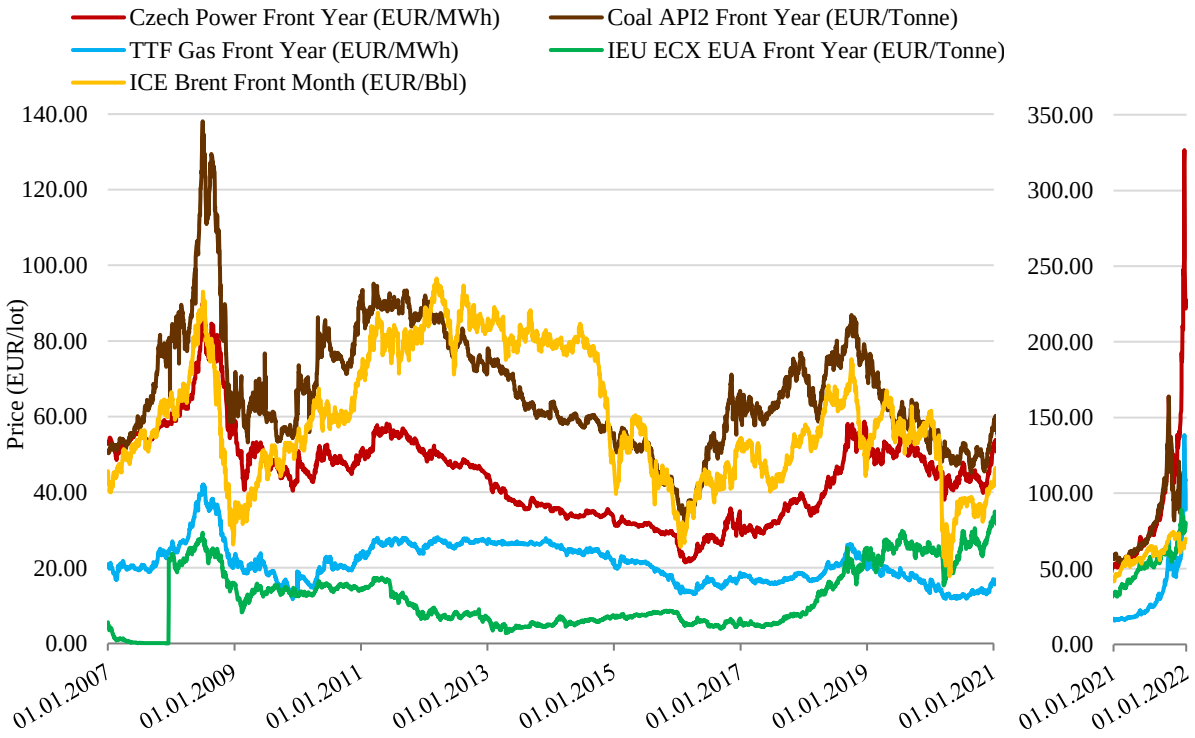

Figure 4.1: Development of commodity prices

The third place is occupied by natural gas, whose share in production is about 8 % [9]. As can be observed in Figure 4.1, the correlation between power and gas prices is very strong, however, the volatility differs significantly. Historically, the volatility of gas prices was much lower because, contrary to power, gas is a storable commodity and thus the trading risks could be reduced. However, with the gradual departure from fossil fuels as well as nuclear power production on the European Union level, many states including the Czech Republic became much more dependent on natural gas, which was supposed to serve as a transitional resource on the path toward further decarbonization.

Nuclear sources, which make up about 40 % of the production, should also be considered [9]. Although the operating costs of nuclear power plants are very low, coal power plants have a perceivable competitive advantage in the areas where the access to cheap resources is possible. This situation occurs not only in the Czech Republic, where coal mining fully covers the domestic consumption, but also in the United States, South Africa, Australia, India and China [10].

Figure 4.2: Development of price of Czech base front year power contract

Last but not least, an important source of information is the price of oil. Although oil covers only a negligible portion of the total power production of the Czech Republic [9], due to its crucial influence on the global economy, the oil market is an important indicator of macroeconomic events. Because of its efficiency, oil price usually reacts to events much earlier than in the case of other energy commodities, such as power or gas, and thus, usually allows to track significant changes in the price trend in the very beginning. Historically, an apparent dependency was observed between the prices of oil and gas, and consequently power, as presented in Figure 4.1. However, during the last years the correlation has been disrupted as a cause of political interventions in this sector.

The effect of the global economy on Czech power prices is demonstrated in Figure 4.2. As can be observed, at the beginning of year 2009 prices reacted to the global financial crisis with a sharp and significant downtrend. In 2011, prices responded to other market uncertainty caused by the Fukushima nuclear disaster, which is perceived as an essential turning point for environmental movement, leading to a decision of gradual phase-out of nuclear power plants. After that we witnessed five years of price decrease, primarily caused by the decrease in the price of fossil fuels and by the significant support of renewable energy resources, whose prices were artificially suppressed due to the subsidies provided. However, at the beginning of 2016 the long-term trend changed, and prices started to increase due to the outage of nuclear power plants in France. The results of the Brexit referendum that took place in June 2016, causing further market uncertainty, provided additional bullish impulse. The increasing trend in power prices continued and was further supported by ambitious plans of environmental initiatives, which shaped the current form of the EU Emission Trading System (EU ETS). One of the most prominent recent events was the adoption of the "Fit for 55" package, which was proposed by the European Commission in July 2021, binding to reduce greenhouse gas emissions in energy, land use, transport and taxation by at least 55% by 2030 [11]. The effect of the global economy can also be captured by changes in price of oil, or by changes in price of stock market indices, such as S&P, MSCI or Dow Jones. For these purposes, the S&P index was chosen as the representative and is further examined in this study.

As depicted in Figures 4.1, 4.2 and 4.3, the year 2021 fully revealed weaknesses of the energy system, which were demonstrated by exerting tremendous political pressure through a threat of disruption in fossil fuel deliveries by the Russian Federation, resulting in an abrupt increase in power and gas prices to an unprecedented level. This pressure escalated in February 2022 when Russia invaded Ukraine, causing the greatest humanitarian crisis in Europe since the Second World War.

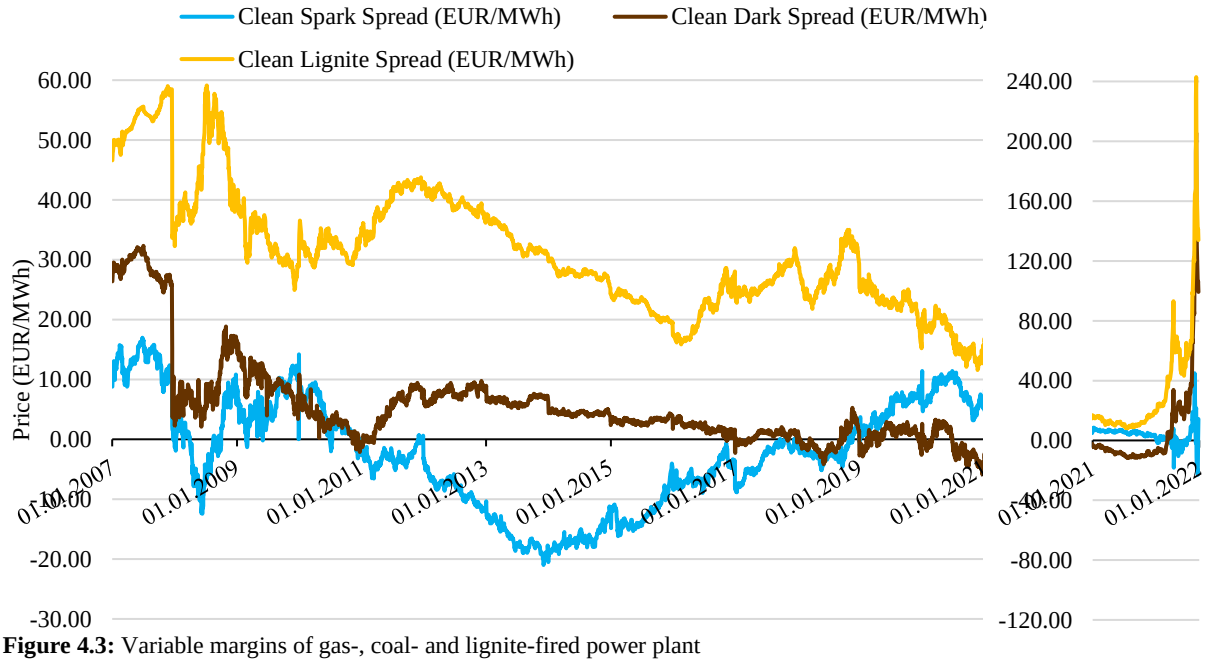

Figure 4.3: Variable margins of gas-, coal- and lignite-fired power plant

The ongoing immense uncertainty on the energy market present from the end of 2021 is having a direct impact not only on the rapid increase in inflation but also on the deepening economic recession in Europe.

To successfully estimate oversold or alternatively overbought market conditions, the concept of power pricing based on the cost of utilized technologies has to be introduced. As indicated in Chapter 3.1, three main indicators are recognized in the context of power production margin, that is, clean spark spread, clean dark spread, and clean lignite spread. These margins often act as price anchors, used by power producers not only to estimate potential profit, but more importantly to determine their hedging strategies. Whereas production from lignite was always highly profitable during the period examined, margins related to other technologies such as production from coal and gas were not always positive, see Figure 4.3. As can be observed, the profitability of gas and coal power production frequently competed with each other, making these two technologies the most prominent for the estimation of the price infection points.

Table 4.1: Correlations among the analysed variables (2016-2020)

	Czech Power Front Year	Coal API2 Front Year	ICE Brent Front Month	TTF Gas Front Year	EUA Front Year	Clean Spark Spread	Clean Dark Spread	Clean Lignite Spread	S&P	RSI	14-day Moving Average	14-day Volatility	Difference from YTD Minimum	Difference from YTD Maximum
Czech Power Front Year	1.00													
Coal API2 Front Year	0.50	1.00												
ICE Brent Front Month	0.51	0.77	1.00											
TTF Gas Front Year	0.49	0.86	0.82	1.00										
EUA Front Year	0.87	0.06	0.19	0.08	1.00									
Clean Spark Spread	0.69	-0.08	-0.11	-0.28	0.84	1.00								
Clean Dark Spread	-0.23	-0.21	-0.19	-0.01	-0.36	-0.18	1.00							
Clean Lignite Spread	0.20	0.87	0.63	0.80	-0.30	-0.33	0.26	1.00						
S&P	0.75	0.07	0.16	-0.07	0.88	0.84	-0.48	-0.31	1.00					
RSI	0.06	0.20	0.13	0.18	-0.02	-0.08	-0.09	0.16	-0.04	1.00				
14-day Moving Average	0.99	0.48	0.49	0.47	0.87	0.70	-0.23	0.19	0.74	-0.05	1.00			
14-day Volatility	0.45	0.24	0.13	0.30	0.38	0.23	-0.11	0.11	0.22	0.01	0.45	1.00		
Difference from YTD Minimum	0.40	0.59	0.34	0.62	0.11	-0.05	0.04	0.55	0.02	0.40	0.35	0.28	1.00	
Difference from YTD Maximum	-0.19	0.36	0.25	0.36	-0.38	-0.49	-0.09	0.40	-0.39	0.53	-0.25	-0.11	0.27	1.00

Presumably, there are other technologies playing a role in the power price settlement process, such as extremely cheap production from renewable resources, or on the contrary, very expensive power production from oil. Nevertheless, it is reasonable to assume that the impact of the first mentioned starts to manifest rather shortly before

delivery when the weather conditions are tangible and has only limited impact on the pricing of long-term contracts. On the other hand, in the event of extreme scarcity of resources, it is possible to settle the power price at the cost of power production from oil. This scenario is however rare and thus it is not investigated further.

As was thoroughly discussed, there are some strong interrelations present among prices of different energy commodities. To avoid the issue of collinearity, it is important to quantify the degree of linear interdependencies among the model input variables. As presented in Table 4.1, there is a strong correlation among prices of gas, coal and oil, presumably due to a significant fuel-switching market mechanism. As the data from year 2016 to 2020 show, the linear dependency of the Czech power price on the price of EUAs was also highly significant, as expected. With regard to large investments of hedge funds into the EU ETS market during the last years, the EUA price became much more correlated with stock market indexes, such as S&P. Last but not least, a substantial correlation was detected also in case of the moving average, due to its strong autocorrelation properties.

5 STATE-OF-THE-ART

There are two main approaches used by professionals for the purposes of estimation of trading signals, i.e., technical analysis, which assumes recurrently appearing trends and patterns over time, and fundamental analysis aspiring to determine intrinsic value of an asset.

Due to its very easy application as well as efficiency, technical analysis has gained importance over time and is now the most equally spread kind of analysis [12]. However, the efficiency of various indicators differs significantly among different types of assets. The effectiveness of Moving Average (MA) based indicators as well as many others is demonstrated for example in [13], [14], [15]. The ability to earn positive returns was also proved in the case of other indicators frequently used, such as Relative Strength Index (RSI) or Stochastic Oscillator (SO) [16]. Furthermore, in some cases RSI, SO as well as parabolic strategies even exceeded the performance of the MA-based indicators [17]. It is important to highlight that the profitability of technical indicators may be affected by volatility, e.g., as demonstrated in [18], some technical trading rules are most profitable during the period with the highest volatility and vice versa. Nevertheless, the use of technical indicators is still not fully standardized, and thus in most cases the expertise of the user is crucial.

Research in the field of the energy industry appears to focus primarily on the analysis of the spot market [19], rather than the forward market, due to its impact on the physical portfolio dispatch and short-term optimisation decisions. Initially, widely used statistical methods such as autoregressive models and Markov models, as well as some artificial intelligence techniques such as support vector machine, random forest and decision trees, were in many cases outperformed by various types of Artificial Neural Networks [20], [21], [22], [23], [24], [25]. However, considering the benefits of specific network structures, the literature is not very united. In the context of spot market forecasting, the outstanding performance of machine learning models, especially deep neural networks, over statistical methods was thoroughly presented in [22], [25], [26]. As discussed in [21], [22] and [24], also GRU, Long Short-Term Memory Neutral Network and some of the hybrid neural networks show promising results in this area of research. On the other hand, according to [23], the best performance was achieved with the convolutional neural network. To summarize, the generalization capability of machine learning techniques provides in many cases an advantage over the conventional statistical methods. However, the network structure must be tailored to the specifics of the task; for example, deep neural networks can provide outstanding performance only in the case of a sufficient number of data samples [22]. In the context of this study, that is, considering the availability of an extensive input dataset and possible autocorrelation dependencies of the time-series, the use of deep neural networks as well as recurrent networks seems reasonable.

Neural networks applications are also very popular in the financial sector as financial services organizations are the second largest sponsors of research in this area [27]. Two main approaches can be taken to improve model accuracy, i.e., improvement of the model structure, and improvement of the input data quality and selection. Even though it seems rather logical that these two approaches have to go hand in hand to obtain reliable results, most of the reported analyses focus on improving the model structure while utilizing only historical samples of the output itself. This imbalance was pointed out and demonstrated, for example, in [28]. Nevertheless, even while using the PCA module, which is a popular feature extraction algorithm, the accuracy of the model was not improved, most probably due to the use of shallow ANNs. At the same time, a convolutional neural network exploiting popular filtering routine used in computer vision showed much worse results compared to other CNN structures, as well as compared to shallow ANN [28]. This analysis demonstrates a strong demand for task-dependent model structures and an adaptive approach to determining input variables.

In general, publications referring to the estimation of trading signals in the financial sector point in a similar direction as the review of articles that focus exclusively on the power spot market. Certain structures of deep feed-forward neural network classifiers [29], convolutional neural networks [28], and recurrent neural networks, including long short-term memory [30], proved to be powerful tools in this field worth further study. When accompanied with an extensive and suitable input data set, these methods are believed to improve the performance of other conventionally used methods.

Unfortunately, analysis of the forward power market, which is the main subject of this study, seems to be rather neglected in the professional literature. Available sources do not sufficiently describe the fundamental pricing and analysis. Instead, many articles follow the risk premium model presented by Fama and French [31], where futures prices are derived as the sum of the expected spot price and risk premium. Unfortunately, a comparison of the power futures market with the power spot market is in certain aspects highly problematic. Contrary to power futures, power spot prices usually show strong autocorrelation dependencies, and as was documented above, their modelling is therefore usually highly efficient.

6 METHODS

6.1 Bias-Variance Trade-Off

Even though the concept of a bias-variance trade-off is routinely familiar, its importance massively increased with the expansion of machine learning techniques, which are in some cases prone to overfitting. This concept is one of the central tenets of the field implying that a model should find a balance between underfitting and overfitting [32].

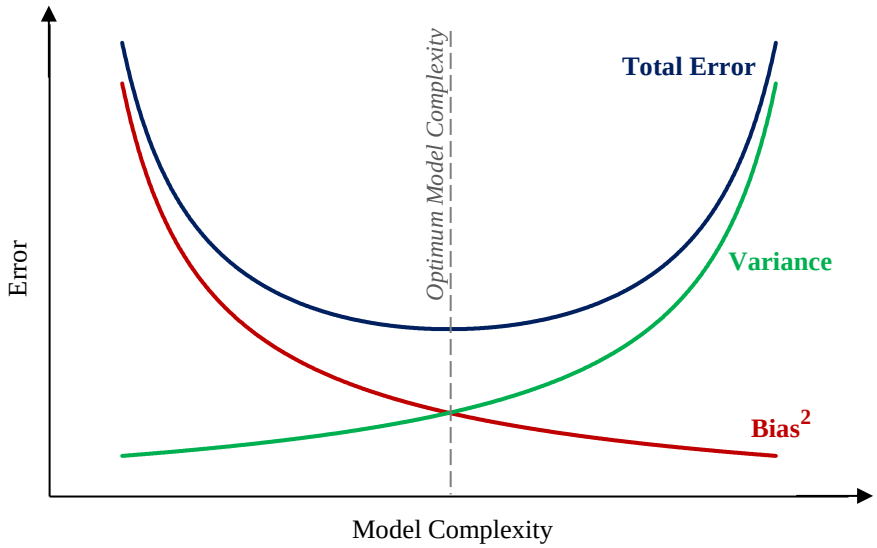

Figure 6.1: Visualization of bias-variance trade-off **[32]**

Let us assume a linear model y with a random noise ε, where $\hat{f}(x)$ is the estimate of the true value $f(x)$.

$$y = f(x) + \varepsilon \tag{6.1.1}$$

$$\varepsilon \sim N(0, \sigma_\varepsilon^2) \tag{6.1.2}$$

Then, the expected mean-squared error of the model is defined as [33]

$$MSE = E\left[\left(y - \hat{f}(x)\right)^2\right] = E\left[\left(f(x) + \varepsilon - \hat{f}(x)\right)^2\right] =$$

$$= E\left[\left(f(x) + \varepsilon - \hat{f}(x) + \bar{f}(x) - \bar{f}(x)\right)^2\right] =$$

$$= E\left[\left(\left(f(x) - \bar{f}(x)\right) - \left(\hat{f}(x) - \bar{f}(x)\right) + \varepsilon\right)^2\right], \tag{6.1.3}$$

where $\bar{f}(x) = E\left[\hat{f}(x)\right]$.

After squaring, the mean-squared error can be further expressed as

$$MSE = E\left[\left(f(x) - \bar{f}(x)\right)^2\right] + E\left[\left(f(x) - \bar{f}(x)\right)\left(\varepsilon - \left(\hat{f}(x) - \bar{f}(x)\right)\right)\right] +$$

$$+E\left[\left(\hat{f}(x) - \bar{f}(x)\right)^2\right] - E\left[\left(\hat{f}(x) - \bar{f}(x)\right)\left(f(x) - \bar{f}(x) + \varepsilon\right)\right]$$

$$+E[\varepsilon^2] + E\left[\varepsilon\left(f(x) - \bar{f}(x) - \left(\hat{f}(x) - \bar{f}(x)\right)\right)\right] \quad (6.1.4)$$

The following identities apply

$$E[\varepsilon] = 0 \quad (6.1.5)$$

$$E[\varepsilon^2] = \sigma_\varepsilon^2 + (E[\varepsilon])^2 = \sigma_\varepsilon^2 \quad (6.1.6)$$

$$\bar{f}(x) = E\left[\hat{f}(x)\right] \quad (6.1.7)$$

$$E\left[\left(\hat{f}(x) - \bar{f}(x)\right)\right] = E\left[\hat{f}(x)\right] - E\left[\bar{f}(x)\right] = 0 \quad (6.1.8)$$

After their substitution into the equation 6.1.4, model error is derived as follows

$$MSE = \left(f(x) - \bar{f}(x)\right)^2 + E\left[\left(\hat{f}(x) - \bar{f}(x)\right)^2\right] + \sigma_\varepsilon^2 =$$

$$= bias^2 + variance + irreducible\ error \quad (6.1.9)$$

As was shown [33], the expected model error consists of three components, i.e., bias, variance, and irreducible error. Bias refers to the delta between the model predictions and the true values. Variance in this context is not a measure of accuracy, but rather a proxy of model complexity, as presented in Figure 6.1. It represents a statistical variance of the predictor over all possible training sets. For example, in the case of overfitting, the models fitted on different training sets significantly differ from each other, i.e., show high variance. Last but not least, irreducible error represents a non-deterministic random noise, which should not be captured by the model. As depicted in Figure 6.1, it is not possible to achieve low bias as well as low variance at the same time. Hence, we attempt to find the sweet spot of the optimum model complexity, where the total error is minimal [32].

One of the essential methods used to optimize the bias-variance trade-off is cross-validation that strives to minimize the test error, and consequently maximize the generalization abilities of the model.

Due to the nature of power prices development, we consider the process a martingale displaying relevant degree of serial correlation, and therefore, in our case it would not be reasonable to split data into training and testing set without taking their sequence into account. Thus, the use of a random split or a k-fold algorithm is not an option. Instead, the dataset was divided chronologically. Due to the definition of target variable, the model parameters were recalculated at the turn of each year, when one fixing period ends, and the respective contract goes into delivery, as indicated in Figure 6.2.

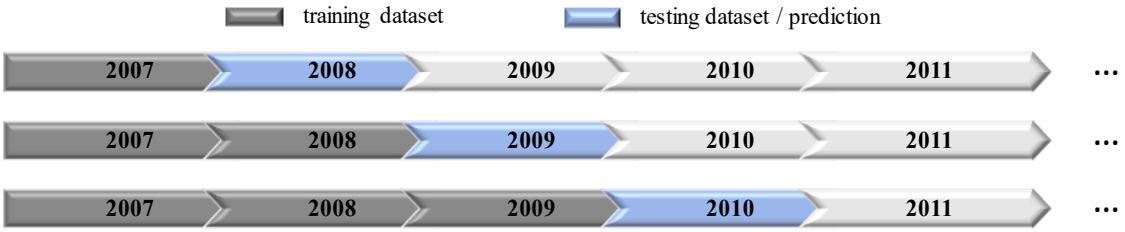

Figure 6.2: Division of training and testing dataset

6.2 Relative Strength Index

This momentum indicator compares the magnitude of recent gains and losses to evaluate overbought or oversold conditions in the market. By its definition, the index lies within 0 and 100, where a value below 30 represents oversold market, and value above 70 indicates overbought situation [34].

$$RSI = 100 - \frac{100}{1+RS} \tag{6.2.1}$$

$$RS = \frac{\sum Up\ changes\ for\ the\ period\ under\ consideration}{\sum |Down\ changes\ for\ the\ period\ under\ consideration|} \tag{6.2.2}$$

RSI is computed over a rolling time period. 14-day time window, which is suggested and widely used in most technical analysis software, was also used for the purposes of this study.

6.3 K-Nearest Neighbor

K-nearest neighbor (KNN) is a nonparametric supervised machine learning method, one of the simplest and easiest algorithms to implement, which memorizes the entire training data, finds a group of k objects that are closest to the test object, and estimates a label based on the predominance of a specific class in this neighborhood. In other words, KNN is a lazy learner, it does not attempt to construct any general internal model, nor is there any explicit training phase of this algorithm. On the other hand, prediction can be computationally very expensive, especially for a large dataset [35] [36].

Despite its simplicity, as was shown by Cover and Hart [37], under certain reasonable assumptions the error of the nearest neighbor rule is capped by twice the Bayes error. Furthermore, the error of the general KNN asymptotically approaches that of the Bayes error; thus, it can be used for its approximation.

Given a training dataset $D = (x, y)$ and test object $z = (x', y')$, the algorithm computes a distance $d(x', x)$ between the test object and every other datapoint.

Then $D_z \subseteq D$ is selected, as a set of k closest training objects to z, and classification is based on the majority class of this selection

$$y' = \underset{v}{\operatorname{argmax}} \sum_{(x_i, y_i) \in D_z} I(v = y_i), \tag{6.3.1}$$

where v is a class label, y_i is the class label for the i^{th} nearest neighbor, and $I(\cdot)$ is an indicator function that returns the value 1 if its argument is true and 0 otherwise [36].

However, *majority vote* approach presented in equation 6.3.1 can be problematic if the nearest neighbor significantly vary in their distance, and the closer ones more reliably indicate the class of the object. In this case an alternative *distance-weighted vote*, which is usually less sensitive to the choice of k, can be used. Weight factor is often defined as a reciprocal of the squared distance $w_i = 1/d(x', x_i)^2$, and consequently, the classification is estimated as

$$y' = \underset{v}{\operatorname{argmax}} \sum_{(x_i, y_i) \in D_z} w_i \times I(v = y_i) \tag{6.3.2}$$

One of the most important aspects affecting the performance of KNN is the choice of hyperparameter k. If k is too small, the model might be sensitive to noise, i.e., there is a danger of over-fitting. On the contrary, large k can lead to oversimplification of the model and its high bias [36].

Finally, the choice of the distance measure is also very important. Even though the Euclidian distance is measure of choice for most applications, it is a well-known fact that it is not a suitable measure for high-dimensional data, furthermore, it is highly scale sensitive. Alternatively, other metrics can be exploited, such as Minkowski distance or cosine distance, which are neither that sensitive to scaling nor to the number of features [38].

$$d_M = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} \tag{6.3.3}$$

$$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \cdot \|\vec{b}\|} \tag{6.3.4}$$

Parameters setting

The algorithm votes are based on the position of 50 nearest neighbors, as this choice of hyperparameter k was experimentally proven to ensure the highest efficiency. The brute-force search algorithm, where all points in each neighborhood are weighted equally, is exploited. Considering the number of input features, the Euclidean distance is exploited as a distance metric.

6.4 Gaussian Naive Bayes

Naive Bayes classifier is a method of supervised learning, which is based on applying Bayes' theorem with the 'naive' assumption of conditional independence between features. Even though it is one of the oldest formal classification algorithms, it has remained one of the most popular until now. This method is known for its exceptional robustness and easy implementation. It does not require high computation resources; no complicated iterative estimation of parameters is needed. Furthermore, the results are easily interpretable. Despite its simplicity and the strong assumption of conditional independence, it is often extremely efficient [35].

Given a class variable y and dependent feature vector $x = (x_1, \dots, x_n)$, Bayes' theorem is defined as [39]

$$P(y|x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n|y)}{P(x_1, \dots, x_n)} \tag{6.4.1}$$

The assumption of conditional independence implies

$$P(x_i|y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i|y) \tag{6.4.2}$$

And therefore, for all i, the relationship in Eq. I can be simplified to

$$P(y|x_1, \dots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, \dots, x_n)} \tag{6.4.3}$$

Since $P(x_1, \dots, x_n)$ is a constant given the input data, we can exploit the following formula for classification purposes. To estimate $P(y)$ and $P(x_i|y)$ we can use Maximum A Posteriori (MAP) method.

$$P(y|x_1, \dots, x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y) \tag{6.4.4}$$

$$\hat{y} = \underset{y}{\operatorname{argmax}}\, P(y)\prod_{i=1}^{n} P(x_i|y) \tag{6.4.5}$$

For the purposes of this study, the likelihood of the features is assumed to be Gaussian

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \tag{6.4.6}$$

, where parameters σ_y and μ_y are estimated using maximum likelihood method.

Parameters setting

Conveniently, Gaussian naive Bayes does not comprise any hyper parameters that require fine-tuning. To promote the calculation stability, a portion of the largest variance of all features was added to the other variance, specifically the factor of 10^{-9}.

6.5 Feed-forward Neural Network

Despite their name, functioning of neural networks cannot be compared in the slightest to the highly complex processes that take place in the human brain, and therefore such exaggerated expectations should be moderated [40]. That being said, ANNs can to a limited extent mimic AI related features such as learning, generalization and abstraction, while achieving good performance in terms of model accuracy, processing speed, fault tolerance, latency, volume and scalability. Following Kolmogorov´s Theorem, a solution to a particular interpretation of Hilbert´s thirteenth problem, the feed-forward neural network containing a single hidden layer with a finite number of nodes can in theory approximate any continuous function [41]. Compared to other conventionally used classification methods such as logistic regression, SVM or decision trees, ANNs offer broader possibilities in terms of non-linear modelling of highly complex systems [42].

Let us assume a feed-forward neural network, where $x = (x_1, \ldots, x_i)$ denotes a high-dimensional input and y a low-dimensional categorical output. Prediction $\hat{y}(x)$ is defined as

$$z_0 = x,\ z_1 = \sigma_1(z_0 W_1 + b_1), \ldots,\ \ z_L = \sigma_L(z_{L-1} W_L + b_L) \tag{6.5.1}$$

$$\hat{y}(x) = softmax(z_L W_{L+1} + b_{L+1}) \tag{6.5.2}$$

where $W_l \in R^{d_l \times d_{l-1}}$ is the weight matrix, $b_l \in R$ is the bias term, d_l is the number of neurons in layer l and σ_l is the activation function [25].

If it is desirable to exploit a multi-class classification task, as in the case of this study, the $softmax$ function is utilized instead of the $sigmoid$, which is on the contrary used only in the case of binary classification. The result of the $softmax$ function is the probability with which the sample is assigned to a class k (see equation 6.5.3). At the same time, this function ensures that all the predicted probabilities sum to one.

$$softmax(a_k) = P(Y = k|X) = \frac{\exp\{a_k\}}{\exp\{a_1\} + \exp\{a_2\} + \cdots + \exp\{a_K\}} \tag{6.5.3}$$

6.5.1 Backpropagation

For the purposes of network training the categorical cross-entropy loss function is being minimized

$$L = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \log y_{nk}, \tag{6.5.4}$$

where t_{nk} is target value and y_{nk} is the predicted probability of the n^{th} observation belonging to the k^{th} category.

To demonstrate the backpropagation algorithm, i.e., short for the backward propagation of errors, which is a standard technique used for training of neural networks, let us assume a two-layer feed-forward neural network as presented in Figure 6.3. X is an input layer represented by an input data matrix of size NxD, Z is a hidden layer consisting of M neurons and Y is an output layer, encompassing K neurons. This structure utilizes weight matrixes $W^{(1)}$ and $W^{(2)}$ of sizes DxM and MxK. Furthermore, a bias term is added at each node in the hidden layer as well as in the output layer, represented by vectors $b^{(1)}$ and $b^{(2)}$. Notice that index (1) is connected to the parameters used between the input and hidden layer, whereas index (2) is utilized for parameters between the hidden and output layer. Let us assume that $tanh$ is used as an activation function in the hidden layer.

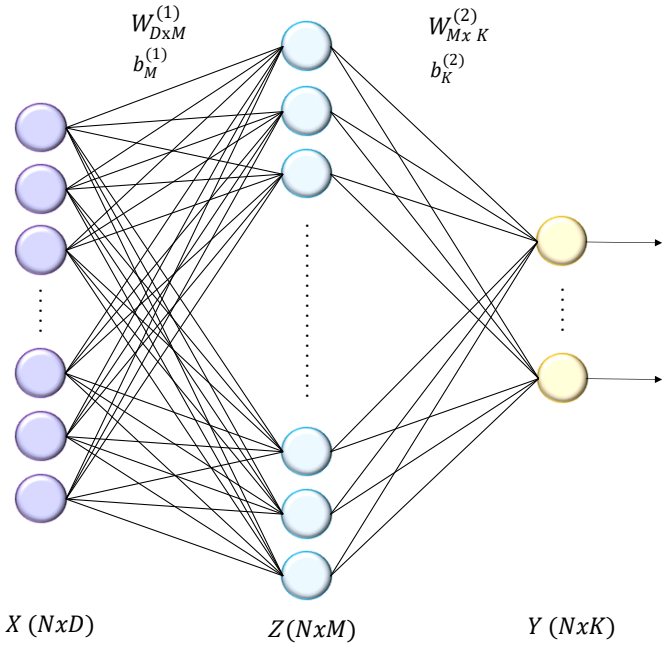

Figure 6.3: Structure of two-layer feed-forward neural network

Utilizing the gradient decent method, the respective derivatives of the loss function will be derived in this chapter [43] [44]. Let us start with recalling the mathematical form of log-likelihood function J, input to the hidden layer α_{nm}, output from the hidden layer z_{nm}, input to the last layer a_{nk}, and model prediction y_{nk}

$$J = -L = \sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \log y_{nk} \tag{6.5.5}$$

$$a_{nm}^{(1)} = W^{(1)}{}_{dm} x_{nd} + b_m^{(1)} \tag{6.5.6}$$

$$z_{nm} = \sigma(\alpha) \tag{6.5.7}$$

$$a_{nk}^{(2)} = W_{:,k}^{(2)} z_n + b_k^{(2)} \tag{6.5.8}$$

$$y_n = softmax(a_n^{(2)}) \tag{6.5.9}$$

Firstly, we will focus on the parameters utilized between the hidden and output layer. Using the chain rule, derivatives of the model parameters can be expressed as

$$\frac{\partial J}{\partial W_{mk}^{(2)}} = \sum_{n=1}^{N}\sum_{k\prime=1}^{K} \frac{\partial J_{nk\prime}}{\partial y_{nk\prime}} \frac{\partial y_{nk\prime}}{\partial a_{nk}} \frac{\partial a_{nk}^{(2)}}{W_{mk}^{(2)}} \tag{6.5.10}$$

$$\frac{\partial J}{\partial b_k^{(2)}} = \sum_{n=1}^{N}\sum_{k\prime=1}^{K} \frac{\partial J_{nk\prime}}{\partial y_{nk\prime}} \frac{\partial y_{nk\prime}}{\partial a_{nk}} \frac{\partial a_{nk}^{(2)}}{\partial b_k^{(2)}} \tag{6.5.11}$$

Derivatives of $\partial J_{nk\prime}$ and ∂a_{nk} are very easy to determine

$$\frac{\partial J_{nk\prime}}{\partial y_{nk\prime}} = \frac{t_{nk\prime}}{y_{nk\prime}} \tag{6.5.12}$$

$$\frac{\partial a_{nk}^{(2)}}{W_{mk}^{(2)}} = z_{nm} \tag{6.5.13}$$

$$\frac{\partial a_{nk}^{(2)}}{\partial b_k^{(2)}} = 1 \tag{6.5.14}$$

However, derivation of $\partial y_{nk\prime}$ is slightly more challenging. In order to efficiently deduct the softmax function, a dummy variable k' was introduced. In case $k' \neq k$, the derivative of $\partial y_{nk\prime}$ is calculated

$$\frac{\partial y_{nk\prime}}{\partial a_{nk}^{(2)}} = (-1) \frac{\exp\left\{a_{nk'}^{(2)}\right\}}{\sum_j \exp\left\{a_{nj}^{(2)}\right\}} \frac{\exp\left\{a_{nk}^{(2)}\right\}}{\sum_j \exp\left\{a_{nj}^{(2)}\right\}} = -\, y_{nk\prime} y_{nk} \tag{6.5.15}$$

From the definition, if $k' = k$ we can write

$$\frac{\partial y_{nk\prime}}{\partial a_{nk}^{(2)}} = \frac{\exp\left\{a_{nk}^{(2)}\right\}}{\sum_j \exp\left\{a_{nj}^{(2)}\right\}} - \frac{\exp\left\{a_{nk}^{(2)}\right\}^2}{\left(\sum_j \exp\left\{a_{nj}^{(2)}\right\}\right)^2} = y_{nk}(1 - y_{nk}) \tag{6.5.16}$$

Both these expressions can be combined using the Kronecker delta function

$$\frac{\partial y_{nk\prime}}{\partial a_{nk}^{(2)}} = y_{nk\prime}(\delta_{kk\prime} - y_{nk}) \tag{6.5.17}$$

where $\delta_{kk'} = 1$ if $k' = k$, and $\delta_{kk'} = 0$ if $k' \neq k$.

Combining the expressions mentioned above, the derivative of model parameters is calculated as

$$\frac{\partial J}{\partial W_{mk}^{(2)}} = \sum_{n=1}^{N} (t_{nk} - y_{nk})\, z_{nm} \tag{6.5.18}$$

$$\frac{\partial J}{\partial b_{k}^{(2)}} = \sum_{n=1}^{N} (t_{nk} - y_{nk}) \tag{6.5.19}$$

In the upcoming part, the focus will be on estimation of the parameters between the input and hidden layer. Using the law of total derivatives, they can be formulated as

$$\frac{\partial J}{\partial W_{dm}^{(1)}} = \sum_{k=1}^{K} \sum_{n=1}^{N} \sum_{k\prime=1}^{K} \frac{\partial J_{nk\prime}}{\partial y_{nk\prime}} \frac{\partial y_{nk\prime}}{\partial a_{nk}^{(2)}} \frac{\partial a_{nk}^{(2)}}{\partial z_{nm}} \frac{\partial z_{nm}}{\partial a_{nm}^{(1)}} \frac{\partial a_{nm}^{(1)}}{\partial W_{dm}} \tag{6.5.20}$$

$$\frac{\partial J}{\partial b_{m}^{(1)}} = \sum_{k=1}^{K} \sum_{n=1}^{N} \sum_{k\prime=1}^{K} \frac{\partial J_{nk\prime}}{\partial y_{nk\prime}} \frac{\partial y_{nk\prime}}{\partial a_{nk}^{(2)}} \frac{\partial a_{nk}^{(2)}}{\partial z_{nm}} \frac{\partial z_{nm}}{\partial a_{nm}^{(1)}} \frac{\partial a_{nm}^{(1)}}{\partial b_{m}^{(1)}} \tag{6.5.21}$$

The remaining derivatives are expressed as follows

$$\frac{\partial a_{nk}^{(2)}}{\partial z_{nm}} = W_{mk}^{(2)} \tag{6.5.22}$$

$$\frac{\partial z_{nm}}{\partial a_{nm}^{(1)}} = 1 - {z_{nm}}^2 \tag{6.5.23}$$

$$\frac{\partial a_{nm}^{(1)}}{\partial W_{dm}} = x_{nd} \tag{6.5.24}$$

Therefore, the respective derivatives of the log-likelihood function are

$$\frac{\partial J}{\partial W_{dm}^{(1)}} = \sum_{k=1}^{K} \sum_{n=1}^{N} (t_{nk} - y_{nk})\, W_{mk}^{(2)} (1 - {z_{nm}}^2)\, x_{nd} \tag{6.5.25}$$

$$\frac{\partial J}{\partial b_m^{(1)}} = \sum_{k=1}^{K}\sum_{n=1}^{N}(t_{nk} - y_{nk})\, W_{mk}^{(2)}(1 - {z_{nm}}^2) \tag{6.5.26}$$

The most important thing which should be observed is that the calculation of derivatives is recursive, and thus, following the same pattern they can also be easily deducted for neural network with more hidden layers, as demonstrated below

$$\nabla_{W^{(l)}} J = z^{(l-1)T}\delta^{(l)} \tag{6.5.27}$$

$$\delta_{nk}^{(L)} = t_{nk} - y_{nk} \tag{6.5.28}$$

$$\delta_{nm^{(l)}}^{(l)} = \sum_{m^{(l+1)}=1}^{M^{(l+1)}} \delta_{nm^{(l+1)}}^{(l+1)} W_{m^{(l)}m^{(l+1)}}^{(l+1)}\, z_{nm^{(l)}}^{(l)\prime} \;, for\ l = 1, \dots, L-1 \tag{6.5.29}$$

The adjustment of model parameters θ during the training phase is proportionate to the estimated error, i.e., the value ∇_θ of the gradient calculated with respect to certain parameter, and to the learning rate η. The adjustment can be expressed as

$$\theta \leftarrow \theta - \eta \nabla_\theta L \tag{6.5.30}$$

Parameters setting

For the purposes of this study, one-, two- and three-layer feed-forward neural network will be examined, i.e., with zero, one and two hidden layers, respectively. It should be noted that the classification with the one-layer neural network is equivalent to a simple logistic regression. The hyperbolic tangent is used as an activation function in the hidden layers. Given the output is categorized into 10 classes, the softmax function is used as an activation function in the output layer. Thus, the topology of the examined networks can be expressed as $(d_1, 10)$, $(d_1, d_2, 10)$ and $(d_1, d_2, d_3, 10)$, where $d_1 = d_2 = d_3 = number\ of\ input\ variables$.

The Adam algorithm, which combines benefits of Momentum (equations 6.5.31 and 6.5.32) as well as Adaptive Learning Rate (equations 6.5.33 and 6.5.34), is used as an optimizer [45].

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\Delta_t \tag{6.5.31}$$

$$\theta_t = \theta_{t-1} + \eta m_t \tag{6.5.32}$$

where m_t is the estimate of momentum, i.e., first moment of gradient Δ, β_1 is a hyper-parameter which takes values from 0 to 1, θ_t is the vector of model parameters and η is the learning rate.

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)\Delta_t^2 \tag{6.5.33}$$

$$\theta_t = \theta_{t-1} - \eta \frac{\Delta_t}{\sqrt{v_t} + \varepsilon}, \tag{6.5.34}$$

where v_t is the estimate of second moment of gradient Δ, β_2 and ε are other hyper-parameters. Bias-corrected first and second moment estimates are thereafter computed as follows

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{6.5.35}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{6.5.36}$$

When combining the equations above, the Adam optimizer can be expressed as

$$\theta_t = \theta_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon} \tag{6.5.37}$$

Learning is processed in maximum of 200 epochs. In case there is no improvement of the testing loss detected in fifty consecutive iterations, the model training is preliminary stopped before the maximum number of epochs is reached. Learning rate is set to 0.001, β_1 and β_2 is set to 0.9 and 0.999 respectively, and epsilon equals 10^{-7}.

Sparse categorical cross entropy, allowing multi-class classification without data transformation to one-hot encoding, is used as the loss function [46].

6.6 Long Short-Term Memory

Recurrent neural network (RNN) differs from the feed-forward structure by the use of a hidden layer with an autoregressive component; let us denote it h_{t-1}. A particular type of RNN called long short-term memory (LSTM) allows a network to learn which of the previous states can be forgotten [47].

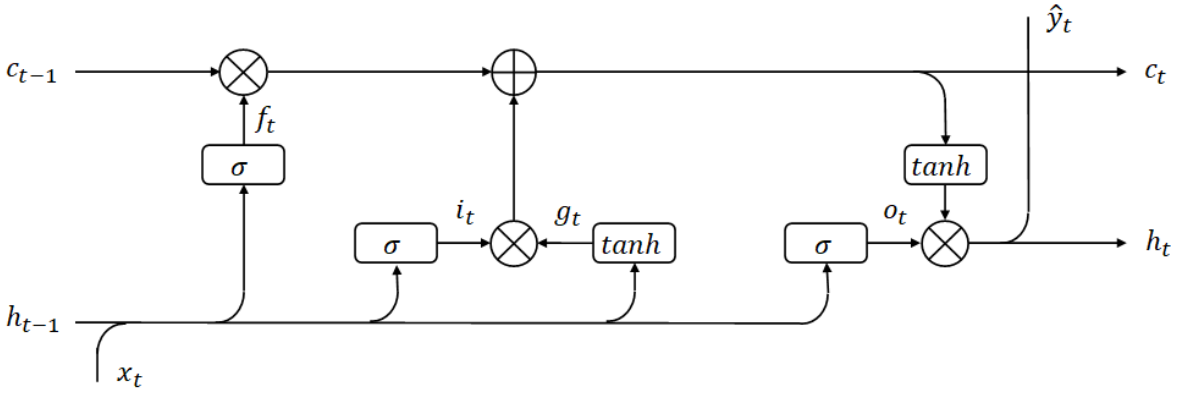

Figure 6.4: Hidden layer of a long short-term memory model

The hidden state is generated by another hidden cell state c_t, which allows the model to remember long-term dependencies. Output is generated as

$$h_t = o_t * \tanh(c_t) \tag{6.6.1}$$

$$c_t = f_t * c_{t-1} + i_t * k_t \tag{6.6.2}$$

$$k_t = \tanh\left(W_c[h_{t-1}, x_t] + b_c\right), \tag{6.6.3}$$

where $*$ denotes the pointwise multiplication, while $f_t * c_{t-1}$ represents the long-range dependence.

State equations can be expressed as

$$\begin{pmatrix} f_t \\ i_t \\ o_t \end{pmatrix} = \sigma(W_c[h_{t-1}, x_t] + b_c), \tag{6.6.4}$$

where f_t, i_t and o_t are input, forget and output states [25].

Parameters setting

For the purposes of this study, a network with one hidden LSTM layer consisting of d neurons is used, where d is a number of input variables. The 14-day time window was exploited to predict the target class. Also in this case, the Adam optimizer with the same set of hyper-parameters is utilized, learning is processed in maximum of 200 epochs, and sparse categorical cross entropy is used as the loss function.

6.7 Support Vector Classifier

Since the 1990´s, when support vector machines (SVMs) were introduced by Vapnik and his colleagues [48] [49] [50], they have gained substantial importance, mainly due to their strong generalization abilities and empirical performance, as well as their advantageous mathematical representations and the possibility of geometrical explanations [51] [52].

As will be demonstrated, SVMs utilize the transformation of the task into a higher-dimensional space where classes are linearly separable. The use of linear classification makes them more robust, easier to train, and less prone to over-fitting. In this way, SVMs often combine the advantages of more complex techniques, while preserving lower computational requirements [51] [52].

6.7.1 Linear SVM (primal problem)

The objective of the task is to maximize the minimum distance between the separating hyperplane and all points, i.e., SVM is considered a 'maximum margin classifier'. In the following paragraph, two different types of margin will be discussed [53].

Let us assume, prediction is correct if

$$y^{(i)}\left(w^T x^{(i)} + b\right) > 0 \tag{6.7.1}$$

the bigger value on the left-hand side of the equation, the higher confidence of prediction. Let us denote $\hat{\gamma}^{(i)}$a functional margin, which quantifies the prediction confidence.

$$\hat{\gamma}^{(i)} = y^{(i)}\left(w^T x^{(i)} + b\right) \tag{6.7.2}$$

Let us also define the functional margin with respect to the training set $D = \{(x_1, y_1), \dots, (x_N, y_N)\}$

$$\hat{\gamma} = \min_{i=1,\dots,N} \hat{\gamma}^{(i)} \tag{6.7.3}$$

The magnitude of the functional margin is dependent on the scale of w and b. Thus, it is also convenient to define the geometric margin $\gamma^{(i)}$, which expresses the actual distance between the line and a data point. The relationship between geometric and functional margins can be expressed as

$$\gamma^{(i)} = \frac{\hat{\gamma}^{(i)}}{\|w\|}) \tag{6.7.4}$$

In a similar way we define a geometric margin with respect to training set $D = \{(x_1, y_1), \dots, (x_N, y_N)\}$

$$\gamma = \min_{i=1,\dots,N} \gamma^{(i)} \tag{6.7.5}$$

Consequently, the objective of the linear SVM classification can be defined as

$$\max_{\gamma,w,b} \gamma \tag{6.7.6}$$

$$s.t.\ y^{(i)}\left(w^T x^{(i)} + b\right) \geq \hat{\gamma}, \forall i = 1, \dots, N \tag{6.7.7}$$

The objective can also be rewritten to more convenient form

$$\min_{w,b} \frac{1}{2}\|w\|^2 \tag{6.7.8}$$

$$s.t.\ y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1, \forall i = 1, \dots, N \tag{6.7.9}$$

To get the 'Soft-Margin SVM' we introduce a slack variable ξ, which enables to meet optimization constrains even if a few points are misclassified, i.e., it acts as a misclassification penalty, and in this way reduces the likelihood of overfitting.

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi^{(i)} \tag{6.7.10}$$

$$s.t. y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1 - \xi^{(i)}, \forall i = 1, \dots, N \tag{6.7.11}$$

$$\xi^{(i)} \geq 0, \forall i = 1, \dots, N \tag{6.7.12}$$

If $\xi^{(i)} = 0$, the point is either on the margin or further away, i.e., is correctly classified. If $0 < \xi^{(i)} < 1$, the point is inside the margin boundaries, but is correctly classified. And for $\xi^{(i)} > 1$, the point is on the wrong side of the decision boundary, i.e., is misclassified [54].

From the above mentioned, it is apparent that loss function for the linear SVM can be defined as a sum of the *large weights penalty* and the *misclassification penalty*. So called Hinge loss function can be also seen as an approximation of the logistic loss function

$$Loss = \frac{1}{2} w^T w + C \sum_{i=1}^{N} \max\left(0, 1 - y^{(i)}\left(w^T x^{(i)} + b\right)\right) \tag{6.7.13}$$

Gradient decent method is used for estimation of model parameters. Note that only samples which violate the margin contribute to the gradient

$$\nabla_w L = w - C \sum_{i:\xi^{(i)}>0} y^{(i)} x^{(i)} \tag{6.7.14}$$

$$\nabla_b L = - C \sum_{i:\xi^{(i)}>0} y^{(i)} \tag{6.7.15}$$

6.7.2 Non-linear SVM (dual problem)

To transform the discussed primal problem into the corresponding dual problem, we start with defining a general form of Lagrangian for the following optimization task [55] [53]

$$\max_x f(x) \tag{6.7.16}$$

$$s.t.\ g_i(x) \leq 0, \forall i = 1, \dots, N \tag{6.7.17}$$

$$h_j(x) = 0, \forall j = 1, \dots, M \tag{6.7.18}$$

$$L(x, \alpha, \lambda) = f(x) + \sum_{i=1}^{N} \alpha_i g_i(x) + \sum_{j=1}^{M} \lambda_j h_j(x) \tag{6.7.19}$$

In this case, the parameters are estimated with the use of Karush-Kuhn-Tucker conditions.

$$\frac{\partial L}{\partial x_d} = 0, \forall d = 1, \dots, D \tag{6.7.20}$$

$$\frac{\partial L}{\partial \lambda_j} = 0, \forall j = 1, \dots, M \tag{6.7.21}$$

$$\alpha_i g_i(x) = 0, \forall i = 1, \dots, N \tag{6.7.22}$$

$$g_i(x) \leq 0, \forall i = 1, \dots, N \tag{6.7.23}$$

$$\alpha_i \geq 0, \forall i = 1, \dots, N \tag{6.7.24}$$

The corresponding Lagrangian form is expressed as

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 + \sum_{i=1}^{N} \alpha_i \left[1 - y^{(i)}\left(w^T x^{(i)} + b\right)\right] =$$
$$= \frac{1}{2}\|w\|^2 + \sum_{i=1}^{N} \alpha_i - \sum_{i=1}^{N} \alpha_i\, y^{(i)} w^T x^{(i)} - \sum_{i=1}^{N} \alpha_i\, y^{(i)} b \tag{6.7.25}$$

From the Karush-Kuhn-Tucker conditions we can derive

$$\nabla_w L(w,b,\alpha) = w - \sum_{i=1}^{N} \alpha_i\, y^{(i)} x^{(i)} = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{N} \alpha_i\, y^{(i)} x^{(i)} \tag{6.7.26}$$

$$\nabla_b L(w,b,\alpha) = \sum_{i=1}^{N} \alpha_i\, y^{(i)} = 0 \tag{6.7.27}$$

After substitution of equation 6.7.26 and 6.7.27 into the Lagrangian, we get the definition of the dual problem. Note that there is only one unknown variable, i.e., alpha.

$$L(w,b,\alpha) = \frac{1}{2} w^T w + \sum_{i=1}^{N} \alpha_i - w^T w = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} w^T w =$$
$$= \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j\, y^{(i)} y^{(j)} x^{(i)T} x^{(j)} \tag{6.7.28}$$

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j\, y^{(i)} y^{(j)} x^{(i)T} x^{(j)} \tag{6.7.29}$$

$$s.t.\, \alpha_i \geq 0, \forall i = 1, \ldots, N \tag{6.7.30}$$

$$\sum_{i=1}^{N} \alpha_i\, y^{(i)} = 0 \tag{6.7.31}$$

Prediction is therefore expressed as

$$w^T x + b = \sum_{i=1}^{N} \alpha_i\, y^{(i)} x^{(i)T} x + b \tag{6.7.32}$$

$$w = \sum_{i\ where\ \alpha_i > 0}^{N} \alpha_i\, y^{(i)} x^{(i)} \tag{6.7.33}$$

$$b = y^{(i)} - w^T x^{(i)} \tag{6.7.34}$$

At this point the only thing which is missing is the integration of a slack variable that allows misclassification of a few data points. The final form of the dual task is then derived as

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi^{(i)} \tag{6.7.35}$$

$$s.t.\, y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1 - \xi^{(i)}, \forall i = 1, \dots, N \tag{6.7.36}$$

$$\xi^{(i)} \geq 0, \forall i = 1, \dots, N \tag{6.7.37}$$

$$L(w, b, \xi, \alpha, \beta) = \frac{1}{2} w^T w + C\sum_{i=1}^{N} \xi^{(i)} + \\ + \sum_{i=1}^{N} \alpha_i \left[1 - \xi^{(i)} - y^{(i)}\left(w^T x^{(i)} + b\right)\right] - \sum_{i=1}^{N} \beta_i \xi^{(i)} \tag{6.7.38}$$

From the Karush-Kuhn-Tucker conditions we can derive

$$\nabla_w L(w, b, \xi, \alpha, \beta) = w - \sum_{i=1}^{N} \alpha_i\, y^{(i)} x^{(i)} = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{N} \alpha_i\, y^{(i)} x^{(i)} \tag{6.7.39}$$

$$\nabla_b L(w, b, \xi, \alpha, \beta) = \sum_{i=1}^{N} \alpha_i\, y^{(i)} = 0 \tag{6.7.40}$$

$$\nabla_\xi L(w, b, \xi, \alpha, \beta) = C - \alpha_i - \beta_i = 0 \quad \Rightarrow \quad \alpha_i + \beta_i = C \tag{6.7.41}$$

$$\alpha_i \left[1 - \xi^{(i)} - y^{(i)}\left(w^T x^{(i)} + b\right)\right] = 0 \tag{6.7.42}$$

$$\beta_i \xi^{(i)} = 0 \tag{6.7.43}$$

After substitution of equations 6.7.39 and 6.7.40 into the Lagrangian we get

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j\, y^{(i)} y^{(j)} x^{(i)T} x^{(j)} \tag{6.7.44}$$

$$s.t. 0 \leq \alpha_i \leq C, \forall i = 1, \dots, N \tag{6.7.45}$$

$$\sum_{i=1}^{N} \alpha_i\, y^{(i)} = 0 \tag{6.7.46}$$

The expressions 6.7.41 – 6.7.43 imply

$$If\ \beta_i > 0, then\ \xi^{(i)} = 0 \tag{6.7.47}$$

$$If\ \beta_i > 0, then\ \alpha_i < C \tag{6.7.48}$$

$$\xi^{(i)} = 0 \quad \Rightarrow \quad \alpha_i \left[1 - y^{(i)}\left(w^T x^{(i)} + b\right)\right] = 0 \tag{6.7.49}$$

Consequently, from equations 6.7.47 – 6.7.49 we can conclude that

$$0 < \alpha_i < C \quad \Rightarrow \quad y^{(i)}\left(w^T x^{(i)} + b\right) = 1 \Rightarrow data\ lies\ on\ the\ margin\ line \tag{6.7.50}$$

$$\alpha_i = 0 \quad \Rightarrow \quad \beta_i = C \ \wedge \ \xi^{(i)} = 0 \ \wedge \ 1 - y^{(i)}\left(w^T x^{(i)} + b\right) < 0$$

$$\Rightarrow data\ lies\ beyond\ the\ margin\ line \tag{6.7.51}$$

As was shown [53], when alpha equals zero, the data lie beyond the margin line. When alpha lies between zero and C, the data are placed directly on the margin line. And in case alpha is equal to C, the data violate the margin.

It is apparent that while the primal problem represents a $minmax$ optimization task, in contrast, the dual problem is expressed as a $maxmin$ optimization

$$primal\ problem \sim \min_{w,b} \max_{\alpha} L(w, b, \alpha) \tag{6.7.52}$$

$$dual\ problem \sim \max_{\alpha} \min_{w,b} L(w, b, \alpha), \tag{6.7.53}$$

where $\min_{w,b} \max_{\alpha} L(w, b, \alpha)$ is always bigger or equal than $\max_{\alpha} \min_{w,b} L(w, b, \alpha)$.

6.7.3 Kernel Methods

As was demonstrated in the previous chapter, the training as well as prediction phases can be expressed only in terms of the inner products of x. This is an important characteristic when we want to transform our linear SVM into a non-linear SVM by applying a nonlinear feature expansion, i.e., the so called Kernel Trick [52] [53].

After replacing the inner products of x with a kernel function K, the training phase can be expressed as

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \, y^{(i)} y^{(j)} K(x^{(i)}, x^{(j)}) \tag{6.7.54}$$

$$s.t. \, \alpha_i \geq 0, \forall i = 1, \ldots, N \tag{6.7.55}$$

$$\sum_{i=1}^{N} \alpha_i \, y^{(i)} = 0 \tag{6.7.56}$$

The prediction phase is derived as

$$\hat{y} = sign\left(\sum_{i=1}^{N} \alpha_i \, y^{(i)} K(x^{(i)}, x) + b \right) \tag{6.7.57}$$

Parameters setting

For the purposes of this study, we will use a Gaussian kernel, also known as Radial Basis Function (RBF) [56].

$$K(x, x') = e^{-\gamma\|x-x'\|^2} \tag{6.7.58}$$

$$\gamma = \frac{1}{No.features \cdot \sigma^2} \tag{6.7.59}$$

where $\|x - x'\|^2$ is the Euclidean distance between two feature vectors, γ can be perceived as a precision (inverse variance). Therefore, higher values of γ represent the skinnier bell curve and vice versa.

The Gaussian kernel corresponds to infinite-dimensional features, i.e., it can be expressed as infinite summation of polynomial kernels. Note that the kernel value only depends on the relative distance between two points, that is, it changes as we move radially outward [56]. Hence, it is often compared to the weighted nearest neighbour model, which was discussed in the previous chapter.

6.8 Ensemble Methods

Ensemble methods, such as random forest, bagging or boosting, are learning algorithms combining multiple weaker learners. Classification is then proceeded as a weighted vote of their predictions. Unique and highly desirable characteristics of ensemble methods is a convergence of its generalization error to a certain limit as the number of base models increases. In other words, with model complexity the train and test error decreases, i.e., its bias as well as variance, contrary to non-ensemble methods, for which train error usually decreases as test error increases [57] [58].

Ensemble methods typically require very little tuning as they are not that sensitive to choices of hyperparameters, compared to other types of machine learning techniques, for instance neural networks. Not only is their performance in many cases exceptional, but, moreover, the algorithm training is fast, does not require a lot of computational resources, and the results are easily interpretable [59].

6.8.1 Bootstrap Estimation & Bagging

Bootstrapping is an important technique used in ensemble modelling. Essentially, it is an input data sampling with replacement, which can under certain conditions significantly reduce variance of the model, as will be demonstrated in this chapter [59].

Given vector $x = (x_1, x_2, \dots, x_n)$, θ_i is defined as a sample with replacement from x of size N, for $i = 1 \dots B$.

As presented below, the expected value of the bootstrapped parameter is equal to the parameter

$$E[\bar{\theta}_B] = E\left[\frac{1}{B}\sum_{b=1}^{B}\hat{\theta}_i\right] = E\left[\frac{1}{B}\left(\hat{\theta}_1 + \cdots + \hat{\theta}_B\right)\right] = \frac{1}{B}BE\left[\hat{\theta}\right] = \theta, \qquad (6.8.1)$$

where $\bar{\theta}_B$ is a sample mean of resampled sample means, and $\hat{\theta}_i$ is a sample mean of bootstrap sample i.

Given a mean μ of $\hat{\theta}$, correlation ρ between bootstrap samples $\hat{\theta}_i$ and $\hat{\theta}_j$, and variance σ^2 of $\hat{\theta}_i$, can be defined as

$$\mu = E\left[\hat{\theta}\right] \qquad (6.8.2)$$

$$\rho = corr\left(\hat{\theta}_i, \hat{\theta}_j\right) = \frac{E\left[\left(\hat{\theta}_i - \mu\right)\left(\hat{\theta}_j - \mu\right)\right]}{\sigma^2} = \frac{E\left[\hat{\theta}_i\hat{\theta}_j\right] - \mu^2}{\sigma^2} \qquad (6.8.3)$$

$$\sigma^2 = var\left(\hat{\theta}_i\right) = E\left[\left(\hat{\theta}_i - \mu\right)^2\right] = E\left[\hat{\theta}_i{}^2\right] - \mu^2 \qquad (6.8.4)$$

The expression of var will be simplifier by using substitution S_B.

$$S_B = \sum_{i=1}^{B}\hat{\theta}_i \qquad (6.8.5)$$

$$\bar{\theta}_B = \frac{1}{B}S_B \qquad (6.8.6)$$

$$\begin{aligned} var(\bar{\theta}_B) = E\left[\left(\frac{1}{B}S_B - \mu\right)^2\right] &= \frac{1}{B^2}E[(S_B - \mu B)^2] \\ &= \frac{1}{B^2}E[S_B^2 - 2\mu B S_B + \mu^2 B^2] = \frac{1}{B^2}E[S_B^2] - \mu^2 \end{aligned} \qquad (6.8.7)$$

$$\begin{aligned} E[S_B^2] = E\left[\left(\hat{\theta}_1 + \cdots + \hat{\theta}_B\right)\left(\hat{\theta}_1 + \cdots + \hat{\theta}_B\right)\right] = \\ = BE\left[\hat{\theta}_i^2\right] + B(B-1)E_{i\neq j}\left[\hat{\theta}_i\hat{\theta}_j\right] \end{aligned} \qquad (6.8.8)$$

After substitution of ρ and σ^2 from the equations 6.8.3 and 6.8.4, we get

$$E[S_B^2] = B(\sigma^2 + \mu^2) + B(B-1)(\rho\sigma^2 + \mu^2) = B\sigma^2 + B(B-1)\rho\sigma^2 + \mu^2B^2 \qquad (6.8.9)$$

$$var(\bar{\theta}_B) = \frac{1}{B^2}(B\sigma^2 + B(B-1)\rho\sigma^2 + \mu^2B^2) - \mu^2 = \frac{1-\rho}{B}\sigma^2 + \rho\sigma^2 \qquad (6.8.10)$$

It is important to emphasize that for $\rho = 1$, $var(\bar{\theta}_B)$ is equal to the original variance. However, if there is no correlation between bootstrap samples, that is $\rho = 0$, the variance decreases by factor $1/B$.

The most significant advantage of bootstrapping appears when highly non-linear models are used, such as decision trees, which produce very irregular decision boundaries. For comparison, it can be shown that in the case of a linear model the correlation is defined as $\rho = N/(2N-1)$, which can be approximated by 0.5 [59].

The bagging algorithm, also known as *bootstrap aggregating*, utilizes the bootstrap distribution to generate different base learners, which are then aggregated. In the case of classification, the *voting* strategy is exploited for the aggregation, i.e., the original dataset X is fed into the set of base models created, and the final result is the label predicted most frequently [60].

As can be shown, the probability that the i^{th} training sample is selected can be approximated by Poisson distribution with $\lambda = 1$, therefore, the probability that the i^{th} sample is chosen at least once is $1 - 1/e \approx 0.632$. This means that in bagging, every base classifier omits approximately one third of original data samples during its training [60].

6.8.2 Stacking [59]

Stacking is a procedure that assumes different influences of each base model, and thus, they are combined by weighting their output, so that better learners have higher weights and vice versa.

$$f(x) = \sum_{m=1}^{M} w_m f_m(x) \tag{6.8.11}$$

Unfortunately, it is not possible to solve the optimization task by minimizing the mean square error, because the probability distribution is not known, and therefore, we cannot estimate its expected value.

$$\begin{aligned} \widehat{w} = \underset{w}{\operatorname{argmin}}\, E_{POP}\left[\left(Y - f(X)\right)^2\right] = \underset{w}{\operatorname{argmin}}\, E_{POP}\left[\left(Y - \sum_{m=1}^{M} w_m f_m(X)\right)^2\right] = \\ = E_{POP}\left[\left(F(X)^T F(X)\right)^{-1} F(X)^T Y\right] \end{aligned} \tag{6.8.12}$$

As an alternative, an error over each i^{th} data pair is calculated as follows

$$\widehat{w}_{stack} = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{N}\left(y_i - \sum_{m=1}^{M} w_m {f_m}^{-i}(x_i)\right)^2 \tag{6.8.13}$$

where ${f_m}^{-i}(x_i)$ is the m^{th} model, which is trained on all input data except (x_i, y_i).

Similarly to SVM, this problem leads to a quadratic programming task

$$\min \sum_{i=1}^{N}\left(y_i - \sum_{m=1}^{M} w_m {f_m}^{-i}(x_i)\right)^2 \tag{6.8.14}$$

$$s.t. w_m \geq 0, \forall m = 1, \dots, M \tag{6.8.15}$$

$$\sum_{m=1}^{M} w_m = 1 \tag{6.8.16}$$

6.8.3 Random Forest

Random forest is an extension of bagging procedure and is usually referred to as one of the state-of-the-art ensemble methods.

It also aims to reduce the correlation between base models, i.e., create a set of decorrelated trees and, in this way, reduce model variance. For this purpose, not only samples, but also features, are randomly chosen for training. The randomized feature selection is one of the main differences which distinguishes random forest from bagging and usually helps it achieve better performance during the training stage as well as lower test error [60].

From a structural perspective, decision tree is a set of nested if-statements of arbitrary depth splitting space orthogonally to axes of the coordinate system. At each node, a sample with replacement θ_b is first chosen from the input data, and then d features are randomly selected. For the purposes of classification, it is recommended that the number of chosen features is equal to floor of $\sqrt{D}$. Based on a preferred criterion, the best split is determined, for example, utilizing the *maximum information gain* objective. The process is repeated until a terminal node or specified maximum depth is reached [61].

6.8.4 Information Entropy & Information Gain [60]

At each non-leaf tree node, the *information gain* criterion is employed to select a split, which maximizes reduction of model uncertainty. Therefore, we define entropy, a measure of how much information we get from finding the value of the random variable.

Given a training set X, the entropy is expressed as

$$H(X) = -\sum_{y \in Y} P(y|X) \log_b P(y|X) \tag{6.8.17}$$

where logarithm base b is usually set to 2.

Let the training set X be divided into subsets $X_1, \dots, X_k$, then the information gain of X is defined as a reduction of information entropy

$$IG(X; X_1, \dots, X_k) = H(X) - \sum_{i=1}^{k} \frac{|X_k|}{|X|} H(X_k) \tag{6.8.18}$$

Hence, the feature-value pair with the largest information gain is selected for a split. If the information gain is equal to zero, it means that there is no gain from splitting the node, and consequently, the node should be made a leaf.

As the definition suggests, features which acquire a lot of different values are favoured, disregarding their factual influence on the classification. To battle this issue, some of the algorithms use a *gain ratio* instead of the *information gain criterion*, which normalizes the number of feature values, and in this way prioritize among features with information gains that are better than average.

$$GR(X; X_1, \dots, X_k) = IG(X; X_1, \dots, X_k) \cdot \left(-\sum_{i=1}^{k} \frac{|X_k|}{|X|} \log \frac{|X_k|}{|X|} \right)^{-1} \quad (6.8.19)$$

One of the most popular criterions used for split selection is maximization of *Gini index*.

$$G(X; X_1, \dots, X_k) = I(X) - \sum_{i=1}^{k} \frac{|X_k|}{|X|} I(X_k) \quad (6.8.20)$$

$$I(X) = 1 - \sum_{y \in Y} P(y|X)^2 \quad (6.8.21)$$

Parameters Setting

The number of trees in the random forest was set by our empirical analysis at 100. When building the trees, a bootstrap sampling is applied. The maximum depth of the tree was not limited, i.e., the nodes were expanded until all leaves were pure, or until the minimum of two samples was reached. The Gini impurity is used as a criterion for split selection, and the number of features considered for the best split was set to the square root of the number of features.

6.8.5 AdaBoost

AdaBoost, short for adaptive boosting, is one of the most powerful ensemble methods in existence. The main objective of ensemble methods usually is to create a low bias and high variance base models; on the contrary, AdaBoost aims to create high bias base learners with accuracy around 50% to 60%. The premise is that by combining many relatively weak and inaccurate learners, a model with high prediction power can be obtained. A decision stump, which divides space into two parts, or logistic regression are examples of the most used weak learners [58].

Formally, the model is defined as [59]

$$F_M(x) = sign\left(\sum_{m=1}^{M} \alpha_m f_m(x) \right), \quad (6.8.22)$$

where F_M is the ensemble model with M base learners and f_m is the m^{th} base learner, which is weighted by α_m.

Contrary to random forest, the AdaBoost model is trained on all data samples without utilizing any bootstrapping technique. Instead, a weight $w_i, \forall i = 1, \dots, N$ representing significance is assigned to every sample. If the y_i is incorrectly classified based on x_i during training, the w_i is increased and vice versa. After training, the base model error weighted by w_i is estimated, and then the weight α_m is derived as a function of the error [58].

$$\varepsilon_m = \frac{\sum_{i=1}^{N} w_i I\left(y_i \neq f_m(x_i)\right)}{\sum_{i=1}^{N} w_i} \tag{6.8.23}$$

$$\alpha_m = \frac{1}{2}\ln\left(\frac{1-\varepsilon_m}{\varepsilon_m}\right) \tag{6.8.24}$$

If the prediction is correct, i.e., $y_i = f_m(x_i)$, w_i is decreased and vice versa.

$$w_i = w_i \exp\left(-\alpha_m y_i f_m(x_i)\right), i = 1, \dots, N \tag{6.8.25}$$

Then w_i is normalized.

$$w_i = \frac{w_i}{\sum_{j=1}^{N} w_j} \tag{6.8.26}$$

The additive model is fitted with the use of *forward stagewise additive modelling* algorithm. At each stage, a new base model is added without modifying the existing base learners [59].

$$(\alpha'_m, \theta'_m) = \underset{\alpha_m, \theta_m}{\operatorname{argmin}} \sum_{i=1}^{N} L\left(y_i, F_{m-1}(x_i) + \alpha_m f_m(x_i, \theta_m)\right) \tag{6.8.27}$$

$$F_m(x) = F_{m-1}(x) + \alpha'_m f_m(x, \theta_m) \tag{6.8.28}$$

where L is the loss function, f_m is the m^{th} base model, and $F(x)$ is the full model.

For the model parameters estimation, AdaBoost usually greedily minimizes the *exponential loss* function

$$L\left(y, f(x)\right) = exp\left(-yf(x)\right) \tag{6.8.29}$$

If the prediction is correct, i.e., the sign of $f(x)$ is the same as the correct label y, the loss function is very small, if not, the loss function acquires a great value [60].

After substitution of loss function into the equation, the following expression is obtained

$$\begin{aligned}(\alpha'_m, f'_m) &= \underset{\alpha_m, f_m}{\operatorname{argmin}} \sum_{i=1}^{N} \exp\left\{-y_i\left(F_{m-1}(x_i) + \alpha_m f_m(x_i)\right)\right\} = \\ &= \underset{\alpha_m, f_m}{\operatorname{argmin}} \sum_{i=1}^{N} \exp\{-y_i F_{m-1}(x_i)\} + \exp\{-y_i \alpha_m f_m(x_i)\} = \\ &= \underset{\alpha_m, f_m}{\operatorname{argmin}} \sum_{i=1}^{N} w_i^{(m)} \exp\{-y_i \alpha_m f_m(x_i)\}\end{aligned} \tag{6.8.30}$$

$$J = \sum_{i=1}^{N} \exp\{-y_i(F_{m-1}(x_i) + \alpha_m f_m(x_i))\} =$$

$$= \sum_{i=1}^{N} \exp\{-y_i F_{m-1}(x_i)\} + \exp\{-y_i \alpha_m f_m(x_i)\} =$$

$$= e^{-\alpha_m} \sum_{y_i = f_m(x_i)} w_i^{(m)} + e^{\alpha_m} \sum_{y_i \neq f_m(x_i)} w_i^{(m)} \quad (6.8.31)$$

After substituting the summations and setting the derivative of J with respect to α to zero, we get the expression for α [59].

$$\frac{\partial J}{\partial \alpha} = e^{-\alpha_m} A + e^{\alpha_m} B = 0 \quad (6.8.32)$$

where A is the weighted number of correct predictions, and B is the weighted number of incorrect predictions.

After solving the equation, the formula 6.8.24 was proven

$$\alpha_m = \frac{1}{2} \ln\left(\frac{A}{B}\right) = \frac{1}{2} \ln\left(\frac{1 - \varepsilon_m}{\varepsilon_m}\right) \quad (6.8.33)$$

Parameters Setting

Two hundred base learners, i.e., decision trees with maximum depth of 2, are utilized in the case of AdaBoost classification. Specifically, the SAMME.R real boosting algorithm, which was designed particularly for the purposes of multi-class classification problem with K classes is exploited as follows [62]

1. *Initialization of the observation weights* $w_i = {}^{1}/_{n}, i = 1,2, \dots, n$
2. *From m=1 to M:*

   (a) A classifier $f_m(x)$ *is fitted to the training data by adjustment of weights* w_i

   (b) The weighted class probability estimates are obtained as follows

   $$p_{m,k}(x) = P_w(c = k|x), k = 1, \dots, K \quad (6.8.34)$$

   (c) Set

   $$h_{m,k}(x) \leftarrow (K-1)\left(log\, p_{m,k}(x) - \frac{1}{K}\sum_{k'} log\, p_{m,k'}(x)\right), k = 1, \dots, K \quad (6.8.35)$$

   where $h_{m,k}(x)$ *is the solution of Lagrange for minimizing the exponential loss of* K*-class classification problem*

   $$\min_{h(x)} E\left[exp\left(-\frac{1}{K} y^T (f_{m-1}(x) + h(x))\right)|x\right] \quad (6.8.36)$$

   $$subject\ to\ h_1(x) + \cdots + h_K(x) = 0 \quad (6.8.37)$$

(d) Set

$$w_i \leftarrow w_i \cdot exp\left\{-\frac{K-1}{K} y_i^T \, log\, p^{(m)}(x_i)\right\}, i = 1, \dots, n \quad (6.8.38)$$

(e) Renormalization of w_i

3. *Calculation of output*

$$C(x) = \underset{k}{argmax} \sum_{m=1}^{M} h_{m,k}(x) \quad (6.8.39)$$

6.9 Programming Environment

Data classification was proceeded in a Python programming environment. In case of k-nearest neighbor, Gaussian naive Bayes, random forest and AdaBoost, calculations were processed with the use of scikit-learn software machine learning library. Feed forward as well as recurrent neural networks were computed utilizing the Keras software that covers implementation of frequently used neural-network building blocks, such as layers, objectives, activation functions and optimizers. Keras acts as an interface for the TensorFlow library. Subsequent data analysis was processed in the Microsoft Office Suite.

6.10 Simulation of Price Fixing

Prices which were estimated to belong to the 1st and 2nd categories, representing a very strong buy signal, will be utilized with highest priority for the purposes of the simulated price fixing procedure. However, in case the examined model does not distinguish any prices as the strong buying opportunity in the first two quarters of the respective year, the third class is considered for the price fixing, and in case none of those is distinguished during the first three quarters, the fourth category is taken into account. If the procedure fails and no buying signal is recognized, price fixing is automatically proceeded 15 days before the end of contract expiry, as it is a common practice.

7 RESULTS

Based on empirical research conducted, the most relevant variables were estimated to be the absolute value of Czech power price, a year-to-date minimum and maximum price of Czech power, the clean spark spread and the clean dark spread, i.e., the vector of input variables unites fundamental as well as technical indicators, and can be defined as $x = (x_{CZ\_power}, x_{\min\ \_YTD}, x_{\max\ \_YTD}, x_{CSS}, x_{CDS})$. Due to the nature of the task, the model parameters were recalculated at the turn of each year, when one fixing period ends, and the respective contract goes into delivery, as specified earlier in Chapter 6.1, Figure 6.2.

7.1 Evaluation Metrics

While solving a classification task, the commonly used evaluation metric is a percentage of correctly classified data, i.e., accuracy. However, in case of a multinominal classification problem, which can be perceived as a generalization of logistic regression, it is highly beneficial to calculate other types of evaluation metrics, which are commonly used in the context of estimating continuous output, such as mean absolute error (MAE), mean square error (MSE) or root mean square error (RMSE). Furthermore, confusion matrices will be presented, allowing a deeper understanding of the model generalization abilities.

For the purposes of this study, model estimates need to be evaluated not only from a quantitative but also from qualitative perspective. Even when low prediction accuracy is achieved, the model could still offer a significant improvement in the context of price fixing. However, in this case, the RMSE should be lower than 4, considering the number of output classes. Thus, in the following chapter accuracy as well as RMSE are scrutinized. To fully demonstrate the generalization abilities of the models, accuracy with the error tolerance of one class is also examined.

7.2 Prediction Performance

To ensure a sufficient number of data samples for model training, the prediction performance is evaluated mainly with the emphasis on later fixing periods, specifically, aggregated statistics for years 2016 till 2020 were calculated. As indicated in Chapter 4, year 2021 was also avoided for the evaluation purposes because of the ongoing unprecedented changes in causalities of the pricing mechanisms, which took place especially in the second half of that year. It corresponds to the results obtained, showing that none of the investigated methods was able to provide a satisfactory prediction performance in that year.

Even though the testing accuracies might seem relatively low at first sight, considering the number of classes the predictions are far away from a random selection, as will be thoroughly discussed in this chapter. Table 7.2, which shows the validation accuracy of models with error tolerance of one class, provides further evidence of the solid prediction

performance of the methods. Additionally, it is important to emphasize that due to the nature of defined task, results of simulation of power purchase are primarily sensitive to correct estimation of strong buy trading signals; however, if strong sell signal is falsely detected, it is not reflected in the results of simulation.

Upon examination of Tables 7.1, 7.2 and 7.3, it is apparent that the model accuracy and the RMSE do not necessarily improve with increasing complexity of the model structure. In terms of the highest accuracy, naive Bayes, 1-, 2- and 3-layer feed forward neural network, AdaBoost and long short-term memory show the best prediction performance, i.e., its average between years 2016 and 2020 varies from 23.7% to 32.9%. Their qualities are further emphasized by the results of validation accuracy with an error tolerance of one class, which spans from 58% to 68%. As expected, RMSE of these methods also acquire low values between 1.66 and 2.13. Although the k-nearest neighbor, support vector and random forest classifier show rather below the average accuracy compared to the other techniques, looking at the RMSE, they provide good results of 1.81, 2.25 and 2.18 respectively, and thus require our further attention.

Especially the long short-term memory shows exceptional performance compared to any other method, which is manifested in excellent, and more importantly much more consistent, prediction accuracy as well as low RMSE throughout the whole dataset.

As can be seen in Table 7.1, the random forest is heavily overfitted during the training phase. Assuming a convergence of the generalization error of the ensemble methods to a certain limit as the number of base models increases, the complexity of the structure of the ensemble models was intentionally boosted, and thus low training error was expected. Nevertheless, contrary to our premise, the testing error was not considerably improved, neither in the case of random forest, nor AdaBoost. Significant difference between training and testing error can also be observed when k-nearest neighbor and support vector classifier algorithms were utilized, implying a non-negligible degree of overfitting, and, therefore, lower generalization capabilities. Training and testing accuracies for different types of neural networks and naive Bayes seem to be in proportion with our expectations.

Table 7.1: Comparison of train and validation accuracy of models

Train and Validation Accuracy [%]

Year	KNN		Naive Bayes		SVC		Random Forest		AdaBoost		1-layer NN		2-layer NN		3-layer NN		LSTM	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
2008	65%	5%	97%	0%	91%	4%	100%	4%	82%	4%	21%	10%	50%	11%	48%	13%	46%	14%
2009	58%	16%	37%	27%	76%	13%	100%	15%	100%	8%	16%	22%	26%	21%	42%	21%	37%	24%
2010	56%	20%	45%	7%	76%	13%	100%	8%	87%	14%	24%	29%	30%	25%	22%	22%	41%	28%
2011	52%	20%	36%	16%	68%	24%	100%	56%	62%	31%	31%	20%	29%	29%	38%	38%	47%	26%
2012	54%	9%	36%	8%	68%	4%	100%	32%	61%	34%	36%	27%	24%	26%	54%	25%	38%	32%
2013	56%	7%	35%	7%	63%	39%	100%	14%	51%	33%	20%	18%	24%	18%	40%	21%	65%	30%
2014	59%	10%	31%	7%	62%	18%	100%	6%	41%	23%	32%	24%	38%	25%	33%	25%	37%	28%
2015	57%	10%	28%	15%	58%	10%	100%	8%	49%	9%	30%	32%	27%	20%	21%	27%	36%	38%
2016	57%	13%	28%	24%	55%	20%	100%	20%	46%	28%	31%	26%	37%	29%	26%	19%	36%	26%
2017	58%	32%	27%	52%	58%	28%	100%	25%	53%	37%	28%	20%	28%	27%	35%	26%	54%	54%
2018	59%	5%	28%	19%	61%	25%	100%	28%	37%	29%	30%	18%	24%	25%	43%	43%	45%	37%
2019	59%	37%	28%	8%	58%	6%	100%	2%	43%	15%	24%	19%	25%	24%	47%	21%	33%	21%
2020	59%	17%	27%	35%	58%	12%	100%	20%	41%	26%	32%	35%	41%	32%	31%	41%	33%	25%
2021	59%	2%	28%	2%	59%	15%	100%	1%	30%	14%	11%	2%	29%	2%	34%	2%	19%	8%
Avg. (2016-2020)	**58%**	**21%**	**28%**	**28%**	**58%**	**18%**	**100%**	**19%**	**44%**	**27%**	**29%**	**24%**	**31%**	**27%**	**36%**	**30%**	**40%**	**33%**

Table 7.2: Comparison of validation accuracy of models with error tolerance of one class

Validation Accuracy with Error Tolerance of One Class [%]									
Year	**KNN**	**Naive Bayes**	**SVC**	**Random Forest**	**AdaBoost**	**1-layer NN**	**2-layer NN**	**3-layer NN**	**LSTM**
2008	13%	0%	9%	9%	9%	49%	25%	43%	33%
2009	36%	69%	36%	53%	22%	66%	60%	53%	60%
2010	54%	31%	54%	32%	52%	70%	64%	49%	66%
2011	42%	43%	67%	83%	71%	51%	68%	74%	61%
2012	40%	40%	34%	57%	82%	50%	63%	57%	62%
2013	26%	26%	60%	54%	63%	59%	59%	38%	70%
2014	34%	27%	37%	25%	70%	53%	48%	68%	71%
2015	29%	33%	28%	24%	30%	52%	42%	68%	80%
2016	50%	57%	55%	39%	58%	43%	45%	65%	65%
2017	87%	95%	82%	76%	77%	69%	70%	68%	95%
2018	41%	59%	50%	48%	59%	58%	74%	61%	64%
2019	74%	17%	22%	14%	40%	45%	65%	62%	57%
2020	50%	72%	67%	52%	63%	76%	82%	83%	55%
2021	3%	2%	26%	3%	18%	5%	3%	3%	31%
Avg. (2016-2020)	**60%**	**60%**	**55%**	**46%**	**59%**	**58%**	**67%**	**68%**	**67%**

Table 7.3: Comparison of root mean square error of models

Root Mean Square Error									
Year	**KNN**	**Naive Bayes**	**SVC**	**Random Forest**	**AdaBoost**	**1-layer NN**	**2-layer NN**	**3-layer NN**	**LSTM**
2008	4.62	4.88	5.84	3.97	5.39	2.43	4.30	2.28	3.03
2009	2.71	2.11	3.08	2.31	2.74	1.72	1.81	2.25	2.04
2010	2.78	2.85	1.87	2.27	2.34	1.65	2.32	2.02	2.13
2011	3.16	2.03	1.62	1.45	2.02	1.89	1.47	1.29	1.91
2012	2.88	2.99	3.03	3.50	1.85	2.53	1.72	2.80	2.00
2013	2.43	2.47	1.65	3.28	2.83	2.26	2.06	2.30	1.62
2014	2.41	4.14	2.53	4.52	1.51	2.04	2.23	1.71	1.66
2015	2.72	3.86	3.08	3.28	4.21	2.52	2.26	1.60	1.16
2016	1.81	2.23	1.94	2.06	1.92	2.30	2.24	1.79	1.78
2017	1.13	0.80	1.40	1.43	1.72	1.46	1.56	1.65	1.01
2018	2.81	3.01	2.87	2.75	2.47	2.47	1.44	2.34	2.16
2019	1.66	2.65	3.04	2.66	2.52	2.52	1.77	2.04	2.16
2020	1.65	1.95	1.99	2.03	2.01	1.72	1.26	1.24	1.92
2021	6.48	6.61	5.62	7.08	4.08	4.32	5.88	5.76	3.25
Avg. (2016-2020)	**1.81**	**2.13**	**2.25**	**2.18**	**2.13**	**2.09**	**1.66**	**1.81**	**1.81**

Even though some of the methods might be favoured based on the discussed accuracy measures, their generalization ability still must be thoroughly examined. Hence, confusion matrixes of the classification results in the period 2016-2020 are presented in Figures 7.1 – 7.10.

As can be observed, even though the relative strength index maps the distribution of classes decently, it does not capture sufficiently the extremes, which is a crucial feature in the case of this specific task. This is reflected during the simulation of price fixing, when relative strength index offers on average the worst savings compared to the usual fixing procedure, both in relative (-8.6%) and absolute terms (-3.48 EUR/MWh).

K-nearest neighbor as well as naive Bayes show reasonable accuracy and generalization abilities (see Figures 7.2 and 7.3), despite the exceptional simplicity of the algorithms. The average cost reduction lies in the range of 10%-11%, which implies average savings of around -4 EUR/MWh in the period 2016-2020.

Despite the fact that the support vector classifier displays slightly worse RMSE and accuracy among the other presented methods, its generalization abilities, especially for the marginal classes, seem to be one of the best (see Figure 7.4). The average savings calculated between years 2016 and 2020 in this case achieve a value of -11.2%, which corresponds to -4.20 EUR/MWh.

The ensemble methods, namely random forest and AdaBoost, represents another group of algorithms examined. As in the case of the support vector classifier, these also show slightly worse results in terms of accuracy and RMSE. However, random forest does a great job in capturing the extremes of the distribution of classes, as is documented in Figure 7.5, resulting in average savings of -11.4%, i.e., -4.26 EUR/MWh. Even though AdaBoost is often thought of as one of the best out-of-box classifiers, its prediction performance is one of the worst from all the methods analysed. As presented in Figure 7.6, it maps the marginal classes very poorly, which is reflected in lower average savings of -9.9%, i.e., -3.81 EUR/MWh.

The last and largest group of algorithms examined are neural networks. Figures 7.7 – 7.10 show a superior prediction performance of 1-, 2-, and 3-layer feed forward neural network, and long short-term memory, compared to the other models. Between years 2016 and 2020, the average savings for the 1-, 2- and 3- layer neural network would reach -11.6% (-4.42 EUR/MWh), -11.4% (-4.23 EUR/MWh) and -11.4% (-4.32 EUR/MWh), respectively. Although the benefits connected to price fixing are comparable for these methods, the more complex structures seem to offer better generalization abilities, as demonstrated by one of the highest classification accuracies and the lowest RMSE achieved. Furthermore, there are other considerable differences which should be taken into account. First, contrary to the 1-layer neural network, 2- and 3- layer neural networks were able to provide more consistent results throughout the whole dataset. That being said, the 3-layer feed forward neural network unfortunately failed to provide a strong buy trading signal in four consecutive years (2010-2013), which is more than any other model.

This insufficiency represents a high risk for purposes of the subsequent price fixing procedure, and thus, should be adequately penalized.

The efficiency of the optimization process can also be assessed based on the development of the loss function and the accuracy during the training process of neural networks. It is apparent that the 1-layer neural network converges to an optimum much smoother than the 2- and 3-layer neural network (Figures 7.11 and 7.12), and without any signs of overfitting. The more complex structures seem to converge to an optimum much faster, and approximately after the 30th iteration the models start to get overfitted, as presented by the increase in validation loss (see Figures 7.13 and 7.17). However, as was proved by additional experiments, in our case this issue can be easily eliminated by decreasing the learning rate value (see Figures 7.15 and 7.19), i.e., for the 2-layer neural network learning rate equal to 0.0005 and for 3-layer neural network learning rate of 0.0001 seems to be the most favourable.

Long short-term memory achieves the highest accuracy and one of the lowest root mean-square-error among all the examined methods. Average savings of -10.8% (-4.09 EUR/MWh) match the exceptional generalization abilities presented in Figure 7.10. From the development of loss function during training it can be deduced that the long short-term memory is considerably less prone to overfitting than the 2- and 3-layer neural network (Figure 7.21).

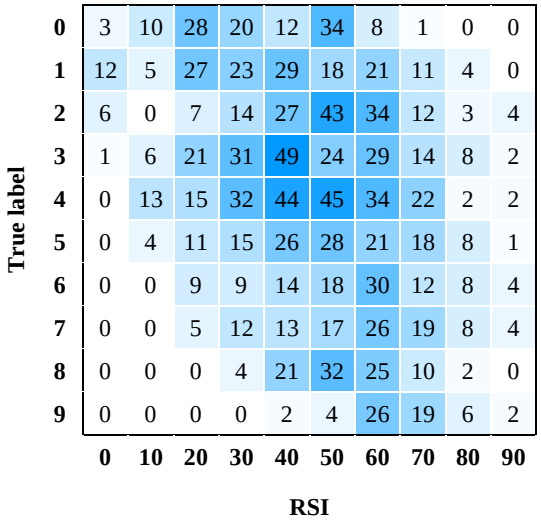

Figure 7.1: Classification with relative strength index

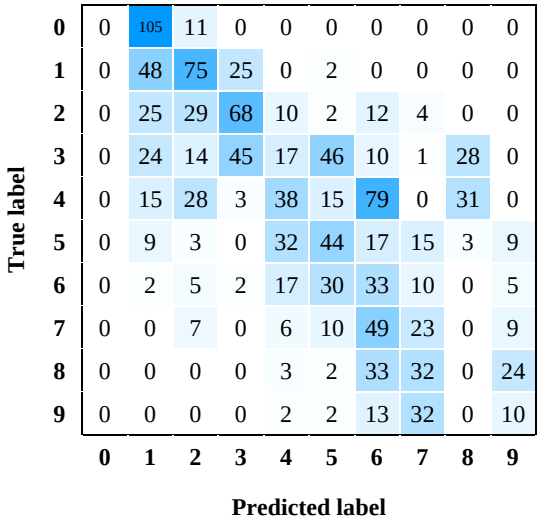

Figure 7.2: Classification with k-nearest neighbor

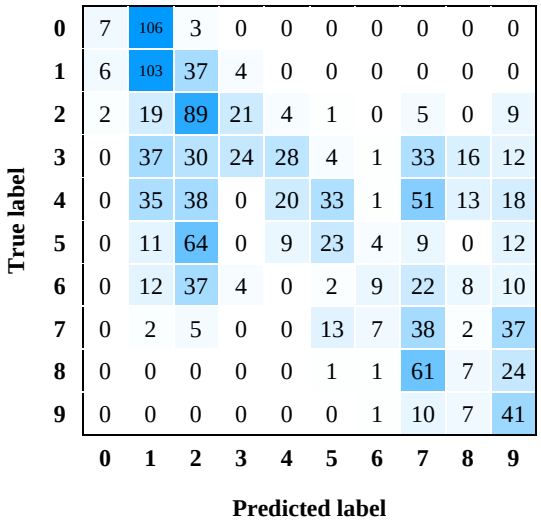

Figure 7.3: Classification with naive Bayes

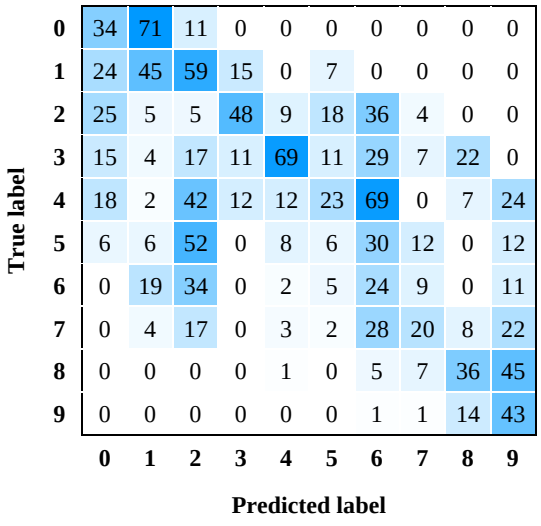

Figure 7.4: Classification with support vector classifier

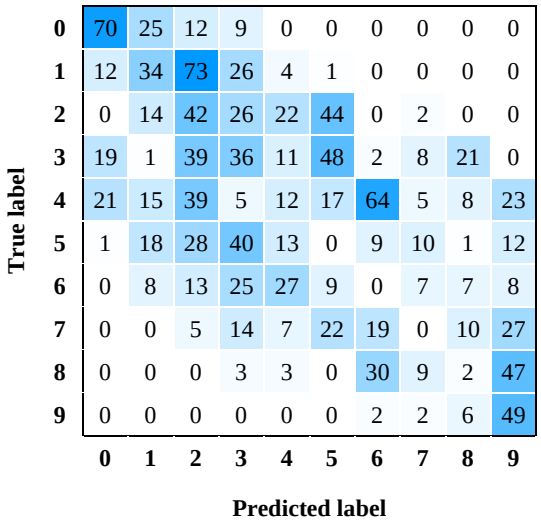

Figure 7.5: Classification with random forest

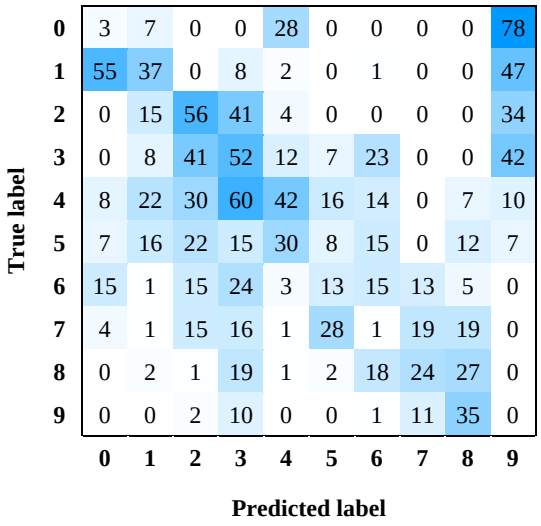

Figure 7.6: Classification with AdaBoost

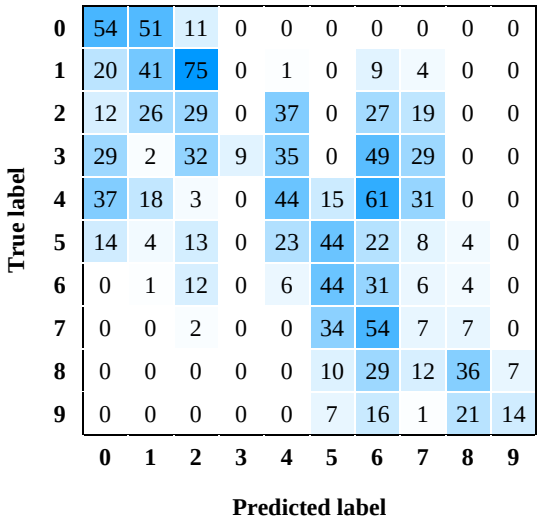

Figure 7.7: Classification with one-layer feed forward neural network

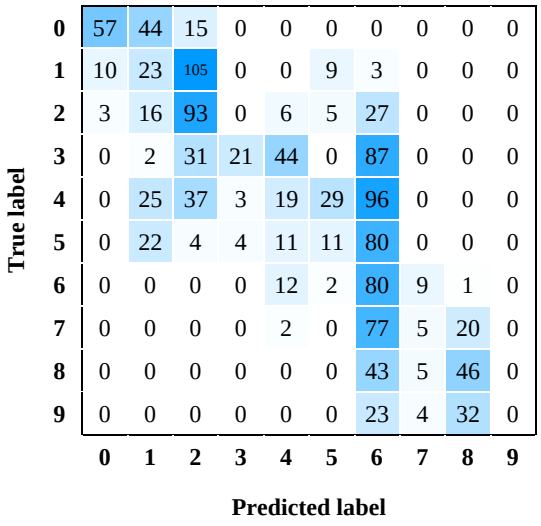

Figure 7.8: Classification with two-layer feed forward neural network

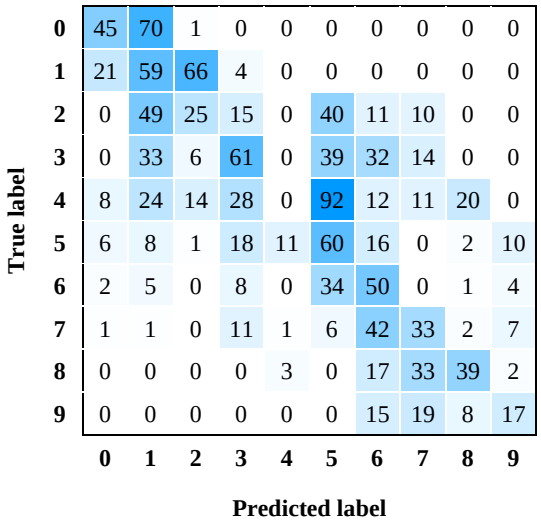

Figure 7.9: Classification with three-layer feed forward neural network

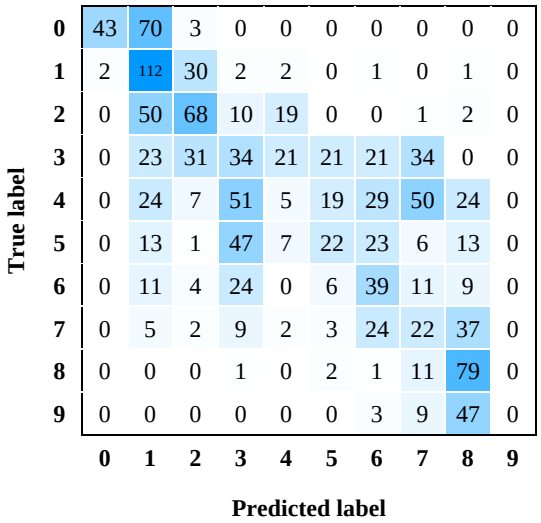

Figure 7.10: Classification with long short-term memory

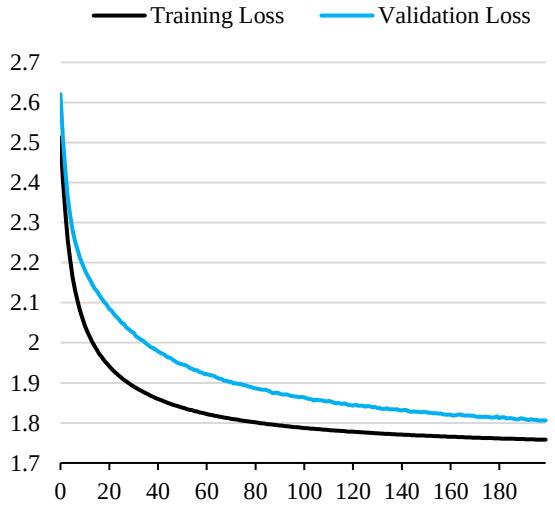

Figure 7.11: Development of loss function during the training and validation phase of 1-layer neural network (lr=0.001)

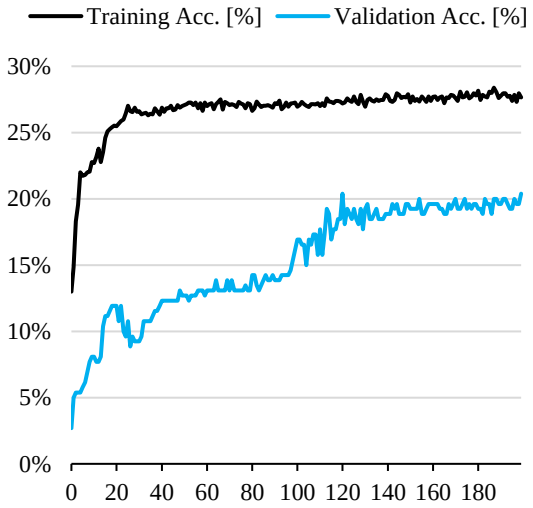

Figure 7.12: Development of accuracy during the training and validation phase of 1-layer neural network (lr=0.001)

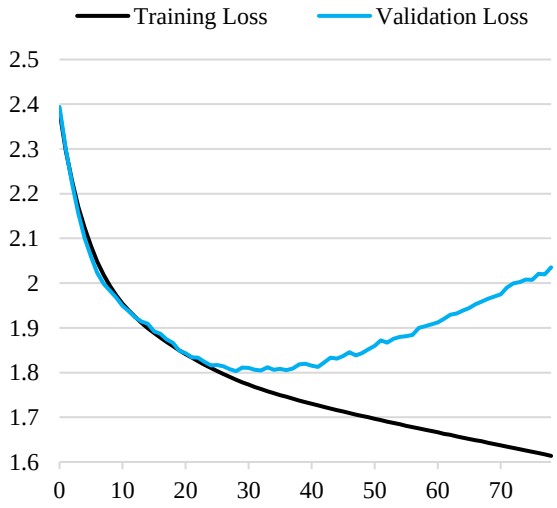

Figure 7.13: Development of loss function during the training and validation phase of 2-layer neural network (lr=0.001)

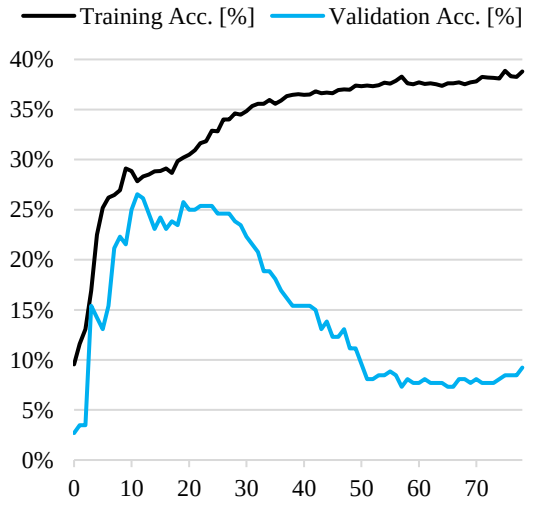

Figure 7.14: Development of accuracy during the training and validation phase of 2-layer neural network (lr=0.001)

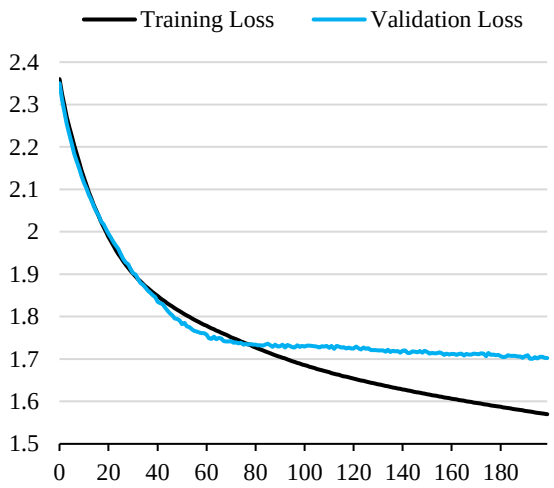

Figure 7.15: Development of loss function during the training and validation phase of 2-layer neural network (lr=0.0005)

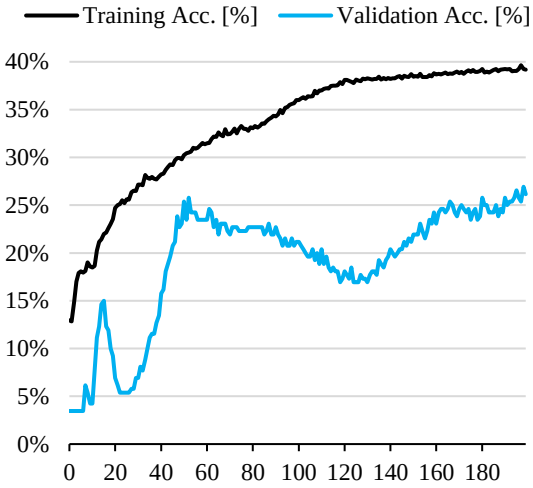

Figure 7.16: Development of accuracy during the training and validation phase of 2-layer neural network (lr=0.0005)

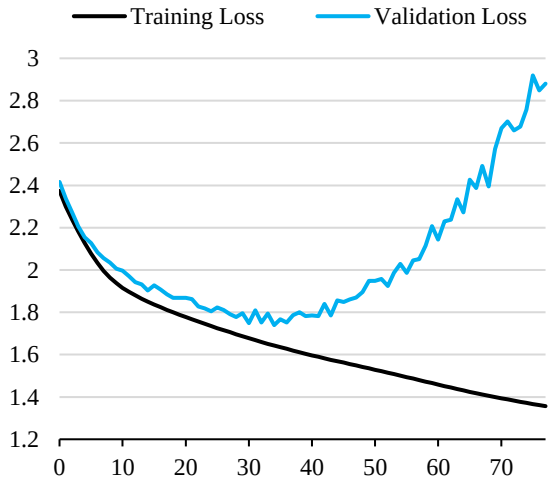

Figure 7.17: Development of loss function during the training and validation phase of 3-layer neural network (lr=0.001)

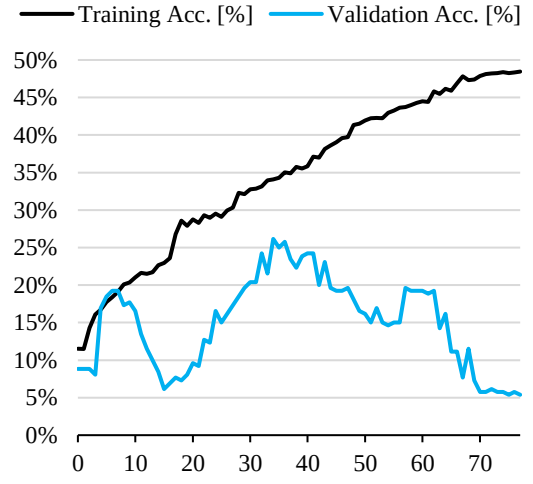

Figure 7.18: Development of accuracy during the training and validation phase of 3-layer neural network (lr=0.001)

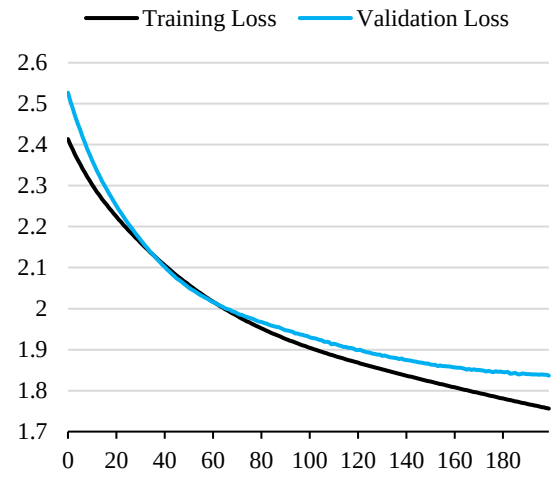

Figure 7.19: Development of loss function during the training and validation phase of 3-layer neural network (lr=0.0001)

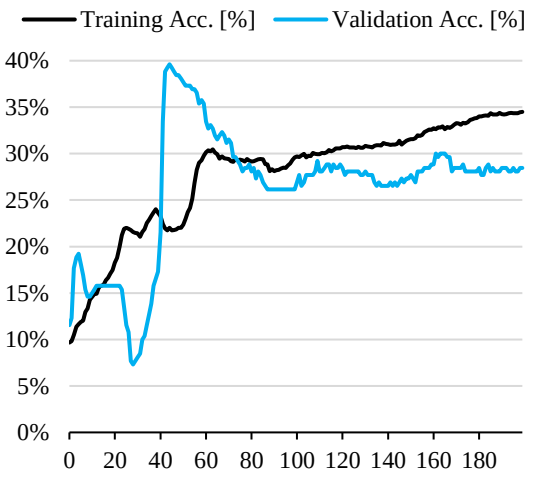

Figure 7.20: Development of accuracy during the training and validation phase of 3-layer neural network (lr=0.0001)

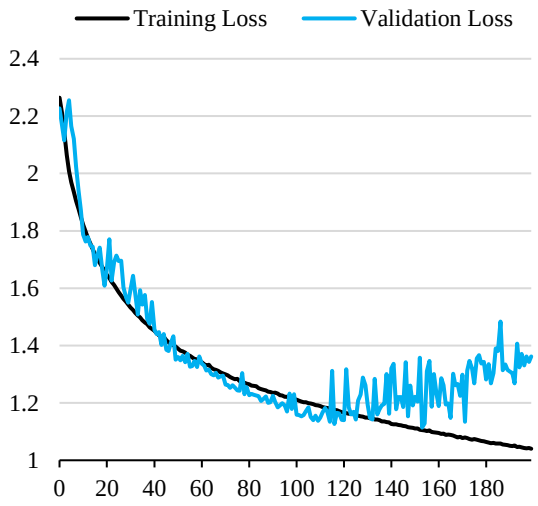

Figure 7.21: Development of loss function during the training and validation phase of long short-term memory network (lr=0.001)

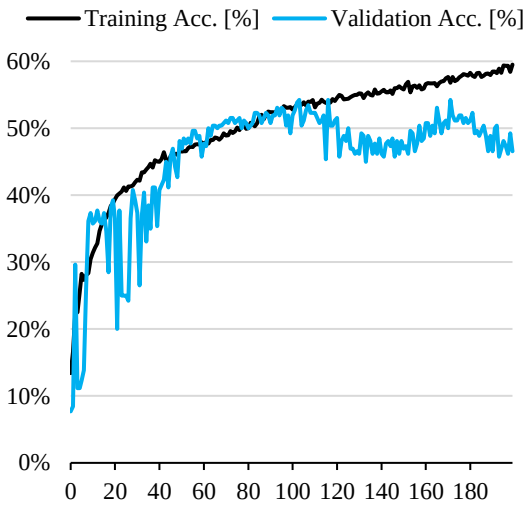

Figure 7.22: Development of accuracy during the training and validation phase of long short-term memory network (lr=0.001)

7.3 Simulation of Price Fixing

The simulation of price fixing was conducted according to the procedure described in detail in Chapter 6.10. Unfortunately, in some cases models failed to provide a buy signal throughout the whole fixing period, which results in fixing at last instance, i.e., 15 business days before the contract expiry, regardless of the actual price. These instances are displayed in red. However, this problem appears mainly during the first year, most likely due to the lack of training samples (see Table 7.4).

As can be observed in Table 7.4, all of the investigated methods exceeded the defined benchmark, i.e., resulted in substantially lower cost compared to the usual price fixing procedure. Nevertheless, long short-term memory seems to be superior among all the analysed techniques. It provided exceptional results in terms of most of the criteria examined. It not only excelled in accuracy and RMSE statistics, but most importantly it offered high prediction performance with the greatest consistency. Significant generalization capabilities were also presented in the confusion matrix in Figure 7.10, which emphasized the low error of the predictions. Furthermore, long short-term memory managed to estimate strong buy signal in most of the years, and in this way eliminated risks of price fixing in the last instance. On the other hand, the disadvantages of this method cannot be neglected. The most prominent weaknesses are high requirements for programming capacity, long training time, sensitivity to initialization of parameters as well as limited possibility of results interpretation. Thus, utilization and maintenance of long short-term memory on an everyday basis might be challenging. To make the procedure more practical and accessible, another solution combining two methods is proposed and discussed in detail in the following chapter.

Table 7.4: Results of simulation of progressive power purchase

Simulation of Progressive Power Purchase											
Year	**Usual Fixing Procedure**	**RSI**		**KNN**		**Naive Bayes**		**SVC**		**Random Forest**	
		Abs.	**Δ %**	**Abs.**	**Δ %**	**Abs.**	**Δ %**	**Abs.**	**Δ %**	**Abs.**	**Δ %**
2008	69.46	66.26	-4.6%	57.50	-17.2%	57.50	-17.2%	57.50	-17.2%	57.50	-17.2%
2009	47.51	45.69	-3.8%	46.36	-2.4%	53.35	12.3%	46.08	-3.0%	45.78	-3.6%
2010	47.70	45.81	-4.0%	48.66	2.0%	45.30	-5.0%	45.65	-4.3%	45.32	-5.0%
2011	53.98	52.74	-2.3%	50.73	-6.0%	49.69	-7.9%	50.48	-6.5%	50.39	-6.7%
2012	47.94	46.73	-2.5%	47.37	-1.2%	48.91	2.0%	46.83	-2.3%	49.00	2.2%
2013	38.31	38.36	0.1%	36.60	-4.5%	36.66	-4.3%	36.90	-3.7%	38.29	-0.1%
2014	34.35	34.13	-0.6%	34.30	-0.1%	34.35	0.0%	34.30	-0.1%	34.40	0.1%
2015	30.81	30.71	-0.3%	30.48	-1.1%	30.80	0.0%	30.63	-0.6%	29.88	-3.0%
2016	27.06	26.19	-3.2%	24.35	-10.0%	23.04	-14.9%	22.76	-15.9%	22.74	-16.0%
2017	32.87	28.96	-11.9%	29.09	-11.5%	29.61	-9.9%	28.71	-12.7%	28.58	-13.0%
2018	45.29	36.55	-19.3%	35.47	-21.7%	35.77	-21.0%	35.77	-21.0%	35.44	-21.7%
2019	50.46	48.68	-3.5%	50.28	-0.4%	47.91	-5.1%	48.48	-3.9%	49.05	-2.8%
2020	43.93	41.81	-4.8%	39.20	-10.8%	43.89	-0.1%	42.90	-2.4%	42.52	-3.2%
2021	90.47	125.73	39.0%	208.09	130.0%	208.09	130.0%	105.91	17.1%	208.09	130.0%
Avg. (2016-2020)	**39.92**	**36.44**	**-8.6%**	**35.68**	**-10.9%**	**36.04**	**-10.2%**	**35.72**	**-11.2%**	**35.67**	**-11.4%**

Table 7.4: Results of simulation of progressive power purchase

Simulation of Progressive Power Purchase

Year	AdaBoost		1-layer NN		2-layer NN		3-layer NN		LSTM	
	Abs.	**Δ %**	**Abs.**	**Δ %**	**Abs.**	**Δ %**	**Abs.**	**Δ %**	**Abs.**	**Δ %**
2008	55.72	-19.8%	56.89	-18.1%	62.09	-10.6%	64.63	-7.0%	57.50	-17.2%
2009	45.98	-3.2%	46.92	-1.2%	44.69	-5.9%	46.73	-1.6%	45.61	-4.0%
2010	45.35	-4.9%	46.30	-2.9%	46.62	-2.3%	47.45	-0.5%	46.32	-2.9%
2011	50.03	-7.3%	50.95	-5.6%	50.95	-5.6%	50.95	-5.6%	52.09	-3.5%
2012	45.07	-6.0%	45.15	-5.8%	47.45	-1.0%	45.15	-5.8%	45.15	-5.8%
2013	38.01	-0.8%	37.53	-2.1%	36.68	-4.3%	36.60	-4.5%	36.97	-3.5%
2014	34.05	-0.9%	33.61	-2.1%	33.41	-2.7%	33.78	-1.7%	35.21	2.5%
2015	30.66	-0.5%	30.24	-1.9%	29.61	-3.9%	30.15	-2.2%	29.71	-3.6%
2016	23.57	-12.9%	22.59	-16.5%	22.39	-17.3%	22.66	-16.3%	22.60	-16.5%
2017	29.64	-9.8%	29.22	-11.1%	29.02	-11.7%	29.02	-11.7%	29.88	-9.1%
2018	35.57	-21.5%	35.13	-22.4%	34.93	-22.9%	35.89	-20.8%	35.55	-21.5%
2019	48.27	-4.4%	48.53	-3.8%	50.30	-0.3%	48.59	-3.7%	48.63	-3.6%
2020	43.52	-0.9%	42.04	-4.3%	41.81	-4.8%	41.85	-4.7%	42.48	-3.3%
2021	52.89	-41.5%	208.09	130.0%	208.09	130.0%	208.09	130.0%	101.55	12.2%
Avg. (2016-2020)	**36.11**	**-9.9%**	**35.50**	**-11.6%**	**35.69**	**-11.4%**	**35.60**	**-11.4%**	**35.83**	**-10.8%**

7.4 Combination of Methods

From a practical point of view, it seems highly convenient to combine a simple model with a more complex technique. The simple model in this case serves for an elementary detection of oversold market conditions on a day-to-day basis, considering its lower accuracy and larger variance compared to other methods. If oversold conditions are recognized, the trading signal will be confirmed or disproved by a complementary model with greater prediction performance, but larger processing requirements.

Understandably, the relative strength index, k-nearest neighbor and naive Bayes were examined as potential candidates for the simple method. The greatest benefits are achieved when the RSI is used as the base model. Combination of RSI with other methods not only saves computational resources, but also results in a further small decrease in cost during the price fixing procedure. Table 7.5 depicts the results of price fixing using the combined approach as well as the percentage decrease of costs compared to classification with a single method.

The benefits of the proposed combined approach during the years 2008 to 2020 are clearly depicted in Figure 7.23. At the first sight it is apparent that the solution offers significantly better results compared to the usual fixing procedure, and in most cases successfully detects the oversold market conditions. However, as was outlined, in the year 2021 all of the examined methods failed to provide sufficient classification accuracy. This year is not displayed to ensure better readability of the Figure 7.23.

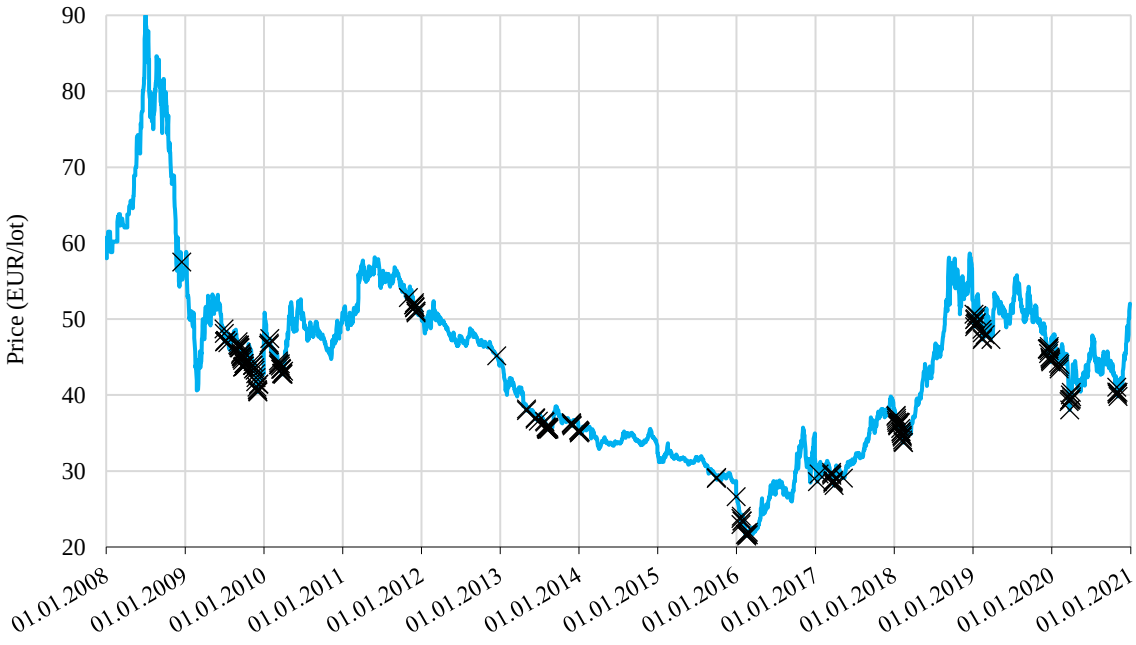

Figure 7.23: Points of the estimated price fixing (combining RSI with LSTM)

Table 7.5: Results of simulation of progressive power purchase utilizing combination of methods

Combination of RSI with Other Methods								
Year	**KNN**		**Naive Bayes**		**SVC**		**Random Forest**	
	Abs.	**Δ %**	**Abs.**	**Δ %**	**Abs.**	**Δ %**	**Abs.**	**Δ %**
2008	57.50	0.00%	57.50	0.00%	57.50	0.00%	57.50	0.00%
2009	45.69	-1.43%	50.89	-4.62%	45.24	-1.82%	44.89	-1.94%
2010	49.03	0.75%	44.52	-1.71%	45.01	-1.41%	44.81	-1.13%
2011	50.95	0.44%	49.35	-0.69%	50.95	0.92%	49.35	-2.06%
2012	47.40	0.06%	48.53	-0.78%	45.55	-2.74%	47.89	-2.26%
2013	36.60	0.00%	36.35	-0.84%	36.67	-0.64%	38.36	0.19%
2014	34.30	0.00%	34.13	-0.62%	34.30	0.00%	34.24	-0.44%
2015	30.43	-0.17%	30.71	-0.28%	30.43	-0.67%	30.13	0.86%
2016	24.59	0.98%	24.06	4.43%	23.19	1.87%	22.73	-0.03%
2017	28.79	-1.02%	28.96	-2.21%	28.51	-0.69%	28.58	0.02%
2018	35.73	0.74%	36.08	0.87%	36.08	0.87%	35.66	0.63%
2019	49.95	-0.66%	47.59	-0.65%	47.63	-1.74%	48.16	-1.81%
2020	39.20	0.00%	41.67	-5.07%	41.45	-3.37%	40.86	-3.90%
2021	208.09	0.00%	208.09	0.00%	208.09	96.47%	208.09	0.00%
Avg. (2016-2020)	**35.65**	**0.01%**	**35.67**	**-0.53%**	**35.37**	**-0.61%**	**35.20**	**-1.02%**

Table 7.5: Results of simulation of progressive power purchase utilizing combination of methods

Combination of RSI with Other Methods										
Year	**AdaBoost**		**1-layer NN**		**2-layer NN**		**3-layer NN**		**LSTM**	
	Abs.	**Δ %**	**Abs.**	**Δ %**	**Abs.**	**Δ %**	**Abs.**	**Δ %**	**Abs.**	**Δ %**
2008	55.03	-1.25%	57.43	0.96%	63.70	2.60%	64.02	-0.95%	57.50	0.00%
2009	45.61	-0.80%	47.00	0.17%	44.46	-0.51%	45.80	-2.00%	44.55	-2.33%
2010	44.89	-1.03%	45.81	-1.05%	43.89	-5.85%	47.45	0.00%	44.19	-4.59%
2011	50.95	1.84%	50.95	0.00%	50.95	0.00%	50.95	0.00%	51.63	-0.89%
2012	45.01	-0.14%	45.15	0.00%	45.15	-4.85%	45.15	0.00%	45.15	0.00%
2013	38.20	0.48%	37.92	1.04%	35.75	-2.54%	36.60	0.00%	36.21	-2.06%
2014	33.93	-0.34%	33.58	-0.10%	33.37	-0.12%	33.78	-0.02%	35.21	0.00%
2015	30.55	-0.36%	30.40	0.53%	29.95	1.15%	30.43	0.93%	29.09	-2.07%
2016	24.07	2.10%	23.14	2.42%	22.66	1.19%	22.46	-0.90%	22.77	0.75%
2017	28.96	-2.30%	28.79	-1.46%	28.69	-1.13%	28.69	-1.11%	28.96	-3.07%
2018	35.66	0.26%	35.33	0.56%	34.63	-0.87%	36.08	0.53%	35.73	0.49%
2019	47.56	-1.47%	48.18	-0.72%	50.51	0.42%	47.70	-1.84%	47.36	-2.61%
2020	41.37	-4.96%	41.55	-1.17%	41.49	-0.78%	41.55	-0.72%	40.64	-4.34%
2021	208.09	293.47%	208.09	0.00%	208.09	0.00%	208.09	0.00%	208.09	104.90%
Avg. (2016-2020)	**35.52**	**-1.27%**	**35.40**	**-0.07%**	**35.59**	**-0.23%**	**35.29**	**-0.81%**	**35.09**	**-1.75%**

8 DISCUSSION

Trading signals were estimated with the use of ten well-established methods, varying from simple technical analysis to more sophisticated Bayesian techniques, and finally to highly complex machine learning algorithms, such as ensemble methods or neural networks.

Even though the list of methods might seem highly extensive, each of them provide significantly different advantages and disadvantages which might be relevant in the context of the analysed task. Due to the lack of published results connected to the progressive power purchase, it seemed necessary to approach the task in a complex way.

Before the results presented above are thoroughly discussed and compared, we would like to dedicate a few chapters to a discussion of potential insufficiencies of the input data, as well as a comparison of models' structure and loss functions, which can explain certain similarities and differences in performance of the methods utilized.

8.1 Input Data

This subchapter is dedicated to a discussion of the potential insufficiencies of the input data. One of the most prominent criticisms might be the difference in the timing of the daily settlements among different commodities and/or exchanges. Due to the fact that the examined settlements of prices of power are determined approximately one hour earlier compared to the other commodities, there might be a small distortion present among the relationship of these variables. Another bias might be connected to the behaviour of market participants, who have the ability to push the market in certain direction or hold prices in a certain range during the settlement period, due to their specific trading positions or business commitments. However, these deviations are perceived small enough to be considered negligible in the context of the defined task.

Another alternative would be to utilize intraday data and track all changes of orders. This solution would offer more samples for model calibration and allow price fixing during the day.

All of these deficiencies might be prevented by creating a separate database with snapshots of prices with certain time stamps, but financial as well as personal expenses connected to such a solution are extensive. Therefore, the presented solution based on daily settlement prices is perceived as a happy medium, offering a more accessible solution for a wider community of experts while ensuring a considerable degree of relevance.

8.2 Comparison of Models' Structure

Despite deep learning and AdaBoost being considered highly distinct techniques, there are some strong similarities that might be observed. Let us compare the base structural unit of both classifiers below [36]

$$z = \tanh\,(W^T x) \tag{8.2.1}$$

$$y = \text{sign}(\alpha^T z) \quad \text{or} \quad \tanh\,(\alpha^T z) \tag{8.2.2}$$

As can be observed, neural networks are essentially networks of linear classifiers, i.e., each of the hidden units play a role of a logistic regressor.

When using a linear classifier as the base learner, the AdaBoost output is defined as

$$\hat{y} = sign\left(\sum_{m=1}^{M} \alpha_m sign(w_m^T x)\right) \tag{8.2.3}$$

Notice how similar the structure of AdaBoost output is to the output of neural network.

$$\hat{y} = sign\left(\sum_{m=1}^{M} \alpha_m tanh(w_m^T x)\right) \tag{8.2.4}$$

The main difference appears to be in the use of a 'hard sign' function in the case of AdaBoost, which returns values -1 or +1, compared to the neural network that utilizes a 'soft sign' function, i.e., a hyperbolic tangent, returning values from the <-1, 1> interval.

The output of the AdaBoost algorithm has a similar structure as the output of a neural network with one hidden layer. However, AdaBoost training is greedy, i.e., model parameters are set based on the values of the previous parameters only. In contrast, the aspiration of the neural network training process is to find a global optimum, thus, all parameters are adjusted concurrently [36].

Surprisingly, similarities with deep learning can also be defined in the case of support vector classifier, which is often considered superior to perceptron because of its ability not only to classify data accurately, but also to separate the classes with *suitable* linear functions. On the contrary, the perceptron is navigated during training only by the measure of accuracy, and therefore, in many cases, the training is often terminated before the appropriate separating function is found [36].

In case the support vector classifier utilizes the sigmoid kernel (equation 8.2.5), which implies the $tanh$ function, its structure seems to be very similar to deep learning (equation 8.2.6).

$$\hat{y} = sign\left(\sum_i \alpha_i y^{(i)} \underbrace{\tanh\left(\gamma {x^{(i)}}^T x + r\right)}_{input} + b\right) \tag{8.2.5}$$

$$\hat{y} = \sigma\left(\sum_i \alpha_i \underbrace{\tanh(w_i{}^T x + r_i)}_{input} + b\right) \tag{8.2.6}$$

Let us first discuss the differences between the two expressions observed in their input. While γ and r are hyperparameters chosen by the user that are static during the entire training process, parameters w_i and r_i are being derived by the gradient decent procedure, during which their values are dynamically changing. Next, we compare the $sign$ function with the $tanh$ activation function. Besides some minor differences, the hyperbolic tangent can be in this context interpreted as an approximation of the $sign$ function [36].

8.3 Comparison of Loss Functions

The loss functions are another important factor having a great effect on the algorithm efficiency during its training, and thus, will be a subject of more detailed analysis in this chapter. Apparently, no training in a conventional sense takes place when using RSI and the k-nearest neighbor, and thus no loss function is considered in these cases.

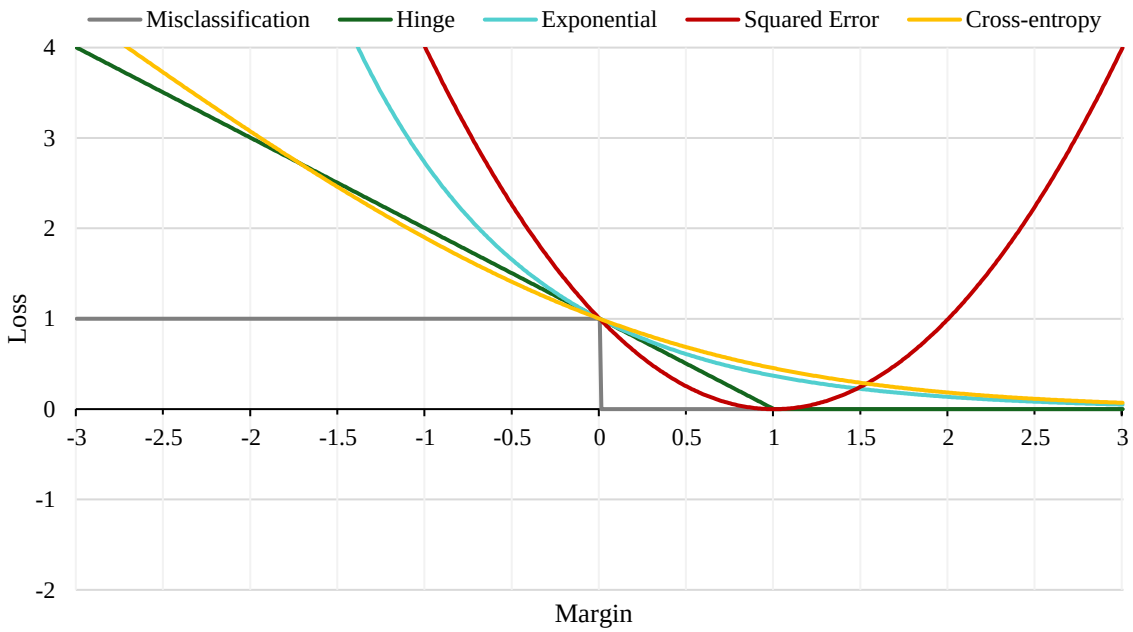

Figure 8.1: Comparison of different types of loss functions

As was verified by Domingos and Pazzani [63], contrary to the squared error loss, when the zero-one loss function is exploited, the naive Bayesian classifier performs quite well even if the independence assumption is violated by a wide margin. Consequently, in this case it has much broader applicability than might be expected. As it was also documented by the results of our study, the Gaussian Bayes classifier showed strong prediction performance, and strong competitiveness against much more complex methods.

For the purposes of neural networks training, the sparse categorical entropy was utilized as the loss function. Due to shape of the activation function, i.e., hyperbolic tangent, the derivation of loss function acquires much higher values around zero, around which the

weights are initialized. As a consequence, the adjustment of the parameters at the beginning of training is large and decreases directly with the decrease in the magnitude of the prediction error, giving this loss a competitive advantage. On the contrary, the derivation of the mean square error loss is very small around zero, which causes inefficiencies and poor model performance during the first stages of training [64].

The support vector classifier utilizes the hinge loss, which penalizes predictions not only when they are false, but also when they are correct but not confident. As mentioned in Chapter 6.7, in practice it means that the loss function equals zero only when the sign of prediction and target match, and the score is bigger or equal to one. Hence, hinge loss strives to classify each point with the emphasis not only on the correctness of classification, but also on its confidence. On the contrary, the cross-entropy loss is derived from a maximum likelihood estimate of the model parameters. For that reason, cross entropy in many cases evokes a larger loss than hinge loss and might result in a less robust prediction efficiency [64].

In the context of AdaBoost algorithm, an exponential loss function was introduced, which converges to zero when prediction and target have the same sign, and converges to infinity in case the sign is opposite. Thus, it has an asymptotic effect very similar to the cross-entropy loss function, as shown in Figure 8.1 [59].

8.4 Comparison of Results

One of the most prominent observations is that the prediction efficiency did not necessarily improve with the increasing complexity of the technique utilized. All of the analysed methods exceeded the defined benchmark and achieved steadily better results through the vast majority of the examined years compared to the usual fixing procedure. It underlines the extraordinary consistency of prediction performance of all the investigated methods in the context of the defined task.

Although the indication of the oversold and overbought market provided by the relative strength index showed the highest level of variance, the exceptional simplicity of its calculation and implementation certainly justifies its importance. From 2016 to 2020, average savings against the defined benchmark were -8.6%. This method is perceived to be a very useful tool for everyday usage, which offers an initial information on the market conditions that might serve as input for further careful analysis.

The k-nearest neighbor represents the second simplest approach analysed in this thesis. Although this algorithm is very simplistic, it provided sufficient results in terms of model accuracy. Between the years 2016 and 2020 examined, this method achieved average savings of -10.9% compared to the benchmark. Nevertheless, as anticipated the classification was slightly more time consuming, as the complexity of brute-force computation of distances between all pairs of data samples approach scale of $O[DN^2]$. Despite its large requirements on memory, this method proved to be satisfactory within

the scope of our task, but with increasing number of samples the brute-force approach might quickly become unfeasible.

The naive Bayes classifier is our only representative of a generative model, which excelled by its simplicity, high training speed, and yet delivered strong prediction performance and average savings of -10.2%. Although the prediction by naive Bayes was less accurate compared to neural networks, its robustness and possibility of reasonable model interpretation grounded in the utilization of probabilistic approach make it a very powerful technique in the context of the defined task.

The support vector classifier shows very similar generalization abilities to the naive Bayes, as well as benefits related to price fixing, which in this case counted for -11.2%. Contrary to the k-nearest neighbor, the processing of this algorithm was very fast and not as memory consuming.

Ensemble methods, including random forest and AdaBoost, are often thought of as the best out-of-box classifiers, mainly because of the proved convergence of its generalization error to a specific limit. Therefore, the relatively weak prediction accuracy of those methods, especially in the random forest, was quite surprising. Even though the results of price fixing simulation show average savings of -11.4% and -9.9% respectively, the generalization abilities of the models are one of the worst among all the methods investigated. Furthermore, these algorithms are highly demanding on the time as well as processing costs, and results are less interpretable.

The last group of algorithms, i.e., neural networks, achieved the best performance among all the methods examined in terms of prediction accuracy as well as savings against the defined benchmark varying from -4.09 to -4.42 EUR/MWh. However, the exceptional generalization capabilities are redeemed by significant processing disadvantages, such as high memory requirements, low training speed and sensitivity to random initiation of parameters. Furthermore, the interpretability of the results is highly limited. Even though in our case the often-inflected threat of overfitting could have been significantly mitigated by an adjustment of learning rate, robustness of the optimization process is generally lower compared to the other algorithms. Long short-term memory was determined as the most appropriate technique for the defined task, combining larger prediction robustness that is manifested in higher consistency of the above-normal results of price fixing simulation. The main edge that long short-term memory has compared to the other techniques is the presence of a feedback loop, i.e., it is able to process sequences of data instead of single points, recognize autocorrelation dependencies, and in this way partially capture time-dependent features of the process.

Due to the significant amount of resources required by LSTM, it seems not convenient to use this algorithm on its own on an everyday basis. Therefore, another approach was proposed that combines the simplicity and low maintenance requirements of the relative strength index and exceptional accuracy of the long short-term memory. The combined approach not only saves valuable computational resources, but also proved to slightly increase the expected value of savings during the price fixing procedure. The average

savings against the defined benchmark for years 2016 to 2020 in this case count for -12.10%.

Despite various techniques that were analysed to find the most beneficial solution for power price fixing, one important approach was left out, i.e., the progressive purchase managed by an expert. Although we did not have the possibility to arrange a simulation of such a kind, there is only a little doubt that an expert judgement would not exceed all the methods studied. Especially in extreme situations where causalities between variables can change rapidly, an expert usually offers an outstanding level of adaptability compared to an artificial system. However, the price of expert is also several times higher than costs connected to the management of an automated system. Considering the typical group of potential customers targeted by this study, i.e., municipalities, factories, hospitals etc., which generally demand a small to medium-size volume of power, the cost of expert seems to be excessive compared to the size of a contract. Therefore, a middle-ground solution offering a substantial value of savings compared to the defined benchmark without an excessive maintenance requirement is preferred.

Considering an average auctioned volume in the order of tens of thousands of MWhs, the potential average savings while utilizing the proposed solution reach a value in the order of tens to hundreds of thousands of EUR per one auction in comparison to the benchmark.

8.5 Reflection on Future Work

Due to the exceptional advantages of long short-term memory with reference to the defined task, it is proposed to focus in further detail on methods that are capable of mining dynamical features of time-series. Therefore, for future research purposes, it is highly recommended to examine, for example, k-nearest neighbor with dynamic time warping, interval-based time-series prediction, time series forest or convolutional neural networks [65].

Furthermore, year 2021 fully revealed weaknesses of the analysed methods. Unfortunately, none of them succeeded to provide sufficient prediction accuracy, and consequently, reliable results in terms of progressive power purchase in that year. Therefore, it seems highly convenient to investigate tools which would help to detect dynamic changes in causalities of the system, such as the change point detection method.

9 CONCLUSION

The main goal of this dissertation thesis was to estimate oversold and overbought market conditions with the use of various classification techniques in the context of the highly challenging task of hedging of the power price by retail customers. The Czech power baseload yearly futures are used as a reference contract for this purpose. Continuous price fixing, which is a very popular and commonly used method for ensuring average profit-loss result, was used as a benchmark to evaluate the benefits of the exploited methods.

To increase model robustness, the price of the reference contract was discretized for each fixing period into ten categories, which represented various market conditions, i.e., scale from strongly oversold to strongly overbought territory. The input dataset consisted of carefully selected variables, which combined the fundamental and technical approach, and were tested not to contain any significant collinearities. Ten well-established techniques were thereafter exploited for data classification, i.e., estimation of trading signals, namely relative strength index, k-nearest neighbor, naive Bayes, support vector classifier, random forest, AdaBoost, 1-, 2- and 3-layer feed forward neural network, and long short-term memory.

Although all of the models examined exceeded the defined benchmark, long short-term memory proved its exceptional qualities among the other methods in terms of consistent prediction performance and generalization abilities. Furthermore, compared to other structures of neural networks, it was proved to be less prone to overfitting. Nevertheless, its weaknesses, such as high requirements for programming capacity, long training time, sensitivity to initialization of parameters as well as limited possibility of results interpretation, should be taken into account. As a result, a solution combining low maintenance and simplicity of relative strength index and high accuracy of long short-term memory was proposed to make the price fixing procedure more practical and efficient. Considering an average auctioned volume in the order of tens of thousands of MWhs, the potential average savings when employing the proposed solution are estimated to reach value in order of tens to hundreds of thousands of EUR per one auction in comparison to the defined benchmark.

In this last paragraph, I would like to briefly look back and recall the very first sentence of this thesis, which was written a few years ago. The text emphasized the importance of market liberalization, which substantially contributes to the efficiency of pricing mechanisms as well as technical progress within the field, without which this thesis would never have been created. Due to recent extreme political tensions, radically amplified by the war in Ukraine, we are now witnesses of an entirely unprecedented situation, which exposed dreadful weaknesses of the European energy system and which can partially or fully compromise liberal principles of the market. Even though no one can tell with certainty what the future arrangement will look like, we are inevitably starting to write a brand-new chapter of the European energy sector.

9.1 Contributions of the Dissertation Thesis

This thesis unfolds a highly challenging task of progressive power purchase by retail customers, i.e., a risk mitigating tool deriving the cost based on several price fixing steps. Due to the lack of publications focusing on this problematic and its increasing importance, especially among small to medium-sized consumers, this thesis successfully contributed to the following areas:

- Variables relevant in the process of estimating overbought/oversold conditions of the Czech power derivatives market were successfully established.
- Causalities and relationships among these variables were examined.
- Ten different classification methods, ranging from a simplistic technical analysis to highly complex machine learning techniques, were analysed in the context of the defined task, and most importantly **their performance was comprehensively compared and evaluated.**
- Taking into account the specific properties of the utilized methods as well as the practicalities of the price fixing procedure, an approach combining relative strength index with the long short-term memory was proposed.
- With regard to the conclusions of our research, a course of further research was suggested.

BIBLIOGRAPHY

[1] ASOCIACE ENERGETICKÝCH MANAŽERŮ. *Úvod do liberalizované energetiky*. Prague: Asociace energetických manažerů, 2016.

[2] POWER EXCHANGE CENTRAL EUROPE, A.S. *Trh s elektrickou energií a zemním plynem pro koncové odběratele - kurzovní lístek*. Available: https://pxe.cz/cs/komoditni-trh/parc/kurzovni-listek

[3] ČESKOMORAVSKÁ KOMODITNÍ BURZA KLADNO. *Energetická burza*. Available: https://www.cmkbk.cz/sekce/energeticka-burza/

[4] S&P GLOBAL PLATTS, A DIVISION OF S&P GLOBAL INC. *Specification Guide: European Electricity*. 2021.

[5] UNITED NATIONS ECONOMICS COMMISSION FOR EUROPE. *Life Cycle Assessment of Electricity Generation Options*. Geneva, 2021.

[6] SCIKIT-LEARN. *Sklearn.preprocessing.StandardScaler*. 2022. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[7] SCIKIT-LEARN. *Sklearn.preprocessing.RobustScaler*. 2022. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html

[8] EUROPEAN COMMISSION. *EU Emissions Trading System*. 2021. Available: https://ec.europa.eu/clima/policies/ets_en

[9] OTE, A.S. *Národní energetický mix*. 2020. Available: https://www.ote-cr.cz/cs/statistika/narodni-energeticky-mix

[10] CHLUMSKÝ, Martin. *Srovnání nákladů jaderných a uhelných elektráren*. In: . Praha: České vysoké učení technické v Praze, 2014. Available: http://www.fel.cvut.cz/education/prace/00024.pdf

[11] EUROPEAN COMMISSION. An official website of the European Union. In: *European Green Deal: Commission proposes transformation of EU economy and society to meet climate ambitions*. 2021. Available: https://ec.europa.eu/commission/presscorner/detail/en/IP_21_3541

[12] GEHRIG, Thomas a Lukas MENKHOFF. *Extended evidence on the use of technical analysis in foreign exchange*. 2006, s. 327-338. Available: doi:10.1002/ijfe.301

[13] BESSEMBINDER, Hendrik a Kalok CHAN. *The profitability of technical trading rules in the Asian stock markets*. 1995, s. 257-284. Available: doi:10.1016/0927-538X(95)00002-3

[14] BROCK, William A., Josef LAKONISHOK a Blake D. LEBARON. *Simple Technical Trading Rules and the Stochastic Properties of Stock Returns*. 1992, s. 1731-1764. Available: doi:10.1111/j.1540-6261.1992.tb04681.x

[15] SHAN, Wang, Jiang ZHI-QIANG, Li SAI-PING a Zhou WEI-XING. *Testing the performance of technical trading rules in the Chinese markets based on superior predictive test*. 2015, s. 114-123. Available: doi:10.1016/j.physa.2015.07.029

[16] TERENCE TAI-LEUNG, Chong a Ng WING-KAM. *Technical analysis and the London stock exchange: testing the MACD and RSI rules using the FT30*. 2008, s. 1111-1114. Available: doi:10.1080/13504850600993598

[17] YI-CHEIN, Chiang, Ke MEI-CHU, Liao TUNG LIANG a Wang CIN-DIAN. *Are technical trading strategies still profitable? Evidence from the Taiwan Stock Index Futures Market*. 2012, s. 955-965. Available: doi:10.1080/09603107.2011.631893

[18] KOZYRA, James a Camillo LENTO. *Using VIX data to enhance technical trading signals*. 2011, s. 1367-1370. Available: doi:10.1080/13504851.2010.537623

[19] VIJAYALAKSHMI, S. a G.P. GIRISH. Artificial Neural Networks for Spot Electricity Price Forecasting: A Review. *International Journal of Energy Economics and Policy*. 2015, s. 1092–1097. Available: http://www.econjournals.com/index.php/ijeep/article/view/1446

[20] CATALãO, J.P.S., S.J.P.S. MARIANO, V.M.F. MENDES a L.A.F.M. FERREIRA. Short-term electricity prices forecasting in a competitive market: A neural network approach. *Electric Power Systems Research* . 2007, s. 1297-1304. Available: doi:10.1016/j.epsr.2006.09.022

[21] UGURLU, Umut, Iikay OKSUZ a Oktay TAS. Electricity Price Forecasting Using Recurrent Neural Networks. *Energies*. 2018. Available: doi:10.3390/en11051255

[22] LAGO, Jesus, Fjo DE RIDDER a Bart DE SCHUTTER. *Forcasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms*. 2018, s. 386-405. Available: doi:10.1016/j.apenergy.2018.02.069

[23] PING-HUAN KUO, Chiou-Jye. *A High Precision Artificial Neural Networks Model for Short-Term Energy Load Forecasting*. 2018. Available: doi:10.3390/en11010213

[24] CHANG, Zihan, Yang ZHANG a Wenbo CHEN. *Electricity price prediction based on hybrid model of adam optimizedLSTM neural network and wavelet transform*. 2019. Available: doi:10.1016/j.energy.2019.07.134

[25] POLSON, Michael a Vadim SOKOLOV. *Deep learning for energy markets*. 2020, s. 195-209. Available: doi:10.1002/asmb.2518

[26] SINGHAL, Deepak a K. S. SWARUP. *Electricity price forecasting using artificial neural networks*. 2011, s. 550-555. Available: doi:10.1016/j.ijepes.2010.12.009

[27] KAASTRA, Iebeling a Milton BOYD. *Designing a neural network for forecasting financial and economic time series*. 1996, s. 215-236. Available: doi:10.1016/0925-2312(95)00039-9

[28] HOSEINZADE, Ehsan a Saman HARATIZADEH. *CNNpred: CNN-based stock market prediction using a diverse set of variables*. 2019, s. 273-285. Available: doi:10.1016/j.eswa.2019.03.029

[29] DIXON, Matthew, Diego KLABJAN a Jin Hoon BANG. *Classification-based Financial Markets Prediction using Deep Neural Networks*. 2016. Available: doi:10.2139/ssrn.2756331

[30] SANG, Chenjie a Massimo DI PIERRO. *Improving trading technical analysis with TensorFlow Long Short-Term Memory (LSTM) Neural Network*. 2019, s. 1-11. Available: doi:10.1016/j.jfds.2018.10.003

[31] FAMA, Eugene F. a Kenneth R. FRENCH. *Commodity Futures Prices: Some Evidence on Forecast Power, Premiums, and the Theory of Storage*. 1987, Available: https://www.jstor.org/stable/2352947.

[32] FORTMANN-ROE, Scott. *Understanding the Bias-Variance Tradeoff*. In: . 2012. Available: http://scott.fortmann-roe.com/docs/BiasVariance.html

[33] CORNELL BOWERS CIS. *Bias-Variance Tradeoff*. 2018. Available: https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote12.html

[34] INSTITUTE, CFA. *CFA Program Curriculum Level 1, Volume 1*. Hoboken, New Jersey : John Wiley & Sons, Inc., 2018.

[35] WU, Xindong, Vipin KUMAR, J. Ross QUINLAN, Joydeep GHOSH a ET AL. *Top 10 algorithms in data mining*. 2007. Available: doi:10.1007/s10115-007-0114-2

[36] LAZY PROGRAMMER TEAM, LAZY PROGRAMMER INC. *Data Science: Supervised Machine Learning in Python*. 2022. Available: https://www.udemy.com/course/data-science-supervised-machine-learning-in-python/

[37] COVER, T. a P. HART. *Nearest neighbor pattern classification*. 1967, s. 21-27. Available: doi:10.1109/TIT.1967.1053964

[38] SCIKIT-LEARN. Distance Metric. In: *Scikit-learn*. 2022. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.DistanceMetric.html#sklearn.metrics.DistanceMetric

[39] ZHANG, Harry. *The Optimality of Naive Bayes*. 2004.

[40] WASSERMAN, Philip. *Neural Computing: Theory and Practice*. New York: Van Nostrand Reinhold, 1989.

[41] NOVÁK, Mirko. *Umělé neuronové sítě: teorie a aplikace*. C.H. Beck, 1998.

[42] ABIODUN, Oludare, Jantan AMAN, Esther Omolara ABIODUN, Victoria Dada KEMI, AbdElatif Mohamed NACHAAT a Arshad HUMAIRA. *State-of-the-art in artificial neural network applications: A survey*. 2018. Available: doi:10.1016/j.heliyon.2018.e00938

[43] KARPATHY, Andrej, Fei-Fei LI a Justin JOHNSON. *Backpropagation, Neural Networks 1*. 2016. Available: https://www.youtube.com/watch?v=i94OvYb6noo

[44] LAZY PROGRAMMER TEAM, LAZY PROGRAMMER INC. *Data Science: Deep Learning and Neural Networks in Python*. Udemy, 2022. Available: https://www.udemy.com/course/data-science-deep-learning-in-python/

[45] P. KINGMA, Diederik a Jimmy LEI BA. Adam: A Method for Stochastic Optimization. In: *ICLR 2015*. San Diego, CA, USA, 2015.

[46] HO, Yaoshianh a Samuel WOOKEY. *The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling*. 2020. Available: doi:10.1109/ACCESS.2019.2962617

[47] HOCHREITER, Sepp a Jürgen SCHMIDHUBER. *LSTM can solve hard long time lag problems*. 1996, Available: https://www.researchgate.net/publication/221620298_LSTM_can_solve_hard_long_time_lag_problems.

[48] CORTES, Corinna a Vladimir VAPNIK. *Support-Vector Networks*. Kluwer Academic Publishers, Boston, 1995, s. 273-297.

[49] VAPNIK, Vladimir. *The Nature of Statistical Learning Theory*. New York: Springer, 1996.

[50] VAPNIK, Vladimir. *Statistical Learning Theory*. New York: John Wiley & Sons Inc, 1998.

[51] CRISTIANINI, Nello a John SHAWE-TAYLOR. *An Introduction to Support Vector Machines and other Kernel-Based Learning Methods*. Cambridge University Press, 2000.

[52] SCHLKOPF, Bernhard a Alexander J. SMOLA. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, 2002.

[53] LAZY PROGRAMMER TEAM, LAZY PROGRAMMER INC. *Machine Learning and AI: Support Vector Machines in Python*. Udemy, 2022. Available: https://www.udemy.com/course/support-vector-machines-in-python/

[54] DENG, Naiyang, Yingjie TIAN a Chunhua ZHANG. *Support Vector Machines: Optimization Based Theory, Algorithms, and Extensions*. CRC Press, 2013.

[55] BOYD, Stephen a Lieven VANDENBERGHE. *Convex Optimization*. New York: Cambridge University Press, 2004.

[56] RASMUSSEN, Carl Edward a Christopher K. I. WILLIAMS. *Gaussian Processes for Machine Learning*. London: MIT Press, 2006.

[57] BREIMAN, Leo. *Random Forests*. 2001, s. 123-145. Available: doi:10.1023/A:1010950718922

[58] SCHAPIRE, Robert E. Explaining AdaBoost. *Empirical Inference*. Springer Verlag GmbH, 2014.

[59] LAZY PROGRAMMER TEAM, LAZY PROGRAMMER INC. *Ensemble Machine Learning in Python: Random Forest, AdaBoost*. Udemy, 2022. Available: https://www.udemy.com/course/machine-learning-in-python-random-forest-adaboost/

[60] ZHOU, Zhi-Hua. *Ensemble Methods: Foundations and Algorithms*. CRC Press, 2012.

[61] HASTIE, Trevor, Robert TIBSHIRANI a Jerome FRIEDMAN. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer, 2009.

[62] ZHU, Ji, Saharon ROSSET, Hui ZOU a Trevor HASTIE. *Multi-class AdaBoost*. 2009, s. 349-360. Available: doi:10.4310/SII.2009.v2.n3.a8

[63] DOMINGOS, Pedro a Michael PAZZANI. *On the Optimality of the Simple Bayesian Classifier under Zero-One Loss*. 1997, s. 103-130. Available: doi:10.1023/A:1007413511361

[64] VARMA, Rohan. *Picking Loss Functions - A comparison between MSE, Cross Entropy, and Hinge Loss*. 2018. Available: https://rohanvarma.me/Loss-Functions/

[65] BAGNALL, Anthony, Aaron BOSTROM a Jason LINES. *The Great Time Series Classification Bake Off: An Experimental Evaluation of Recently Proposed Algorithms*. 2016.

PUBLICATIONS

JONÁKOVÁ, Lenka a Ivan NAGY. Power purchase strategy of retail customers utilizing advanced classification methods. *Neural Network World*. 2021, 31(2), 89-107. ISSN 23364335. Available: doi:10.14311/NNW.2021.31.005

LIST OF FIGURES

LIST OF TABLES