Electricity Price Prediction

Asal Asgari
UiT the Arctic University of Norway
Narvik, Norway
aas047@uit.no

***Abstract*—Time series data analysis is complex and challenging for energy price forecasting due to a number of factors that can impact energy prices (such as inflation, seasonality, societal forces, and economic policies). In this study we compare two methods based on advanced machine learning and deep learning to enhance the performance of a prediction system. First, we applied extreme gradient boosting as a feature-selection technique to extract important features in addition to studying correlations between high-dimensional time-series data and removing redundant features. Then, we fed selected features into a deep long-short-term memory (LSTM) network and an XGboost model to forecast prices and compare the performance of each method. Experimental results based on the Nordpool dataset covering 2017-2021 showed that XGboost had a better performance result compared to LSTM.**

***Index Terms*—Energy-price forcasting; XGBoost; LSTM; Deep learning; Machine learning**

I. Introduction

As renewable energy sources become more prevalent in today's power systems, electricity generation becomes more volatile, and resulting electricity prices are harder to predict. In recent years, energy prices have soared throughout Europe, making the idea of forecasting an attractive and challenging research area. Various factors affect these prices, such as inflation, seasonality, economic policy and economic shocks. Consequently, any forecasting system can lose accuracy as a result of such factors. An accurate forecast, however, is beneficial to companies and investors; it can also serve as a key measure for assessing economic trends. Using historical data from the NordPool energy exchange we will try to to develop a model that would predict these prices in the prices for price zone NO1(Oslo) and NO2(Kristiansand)with two different methods, and we will evaluate which models are working best on the prediction. The two methods to be compared are XGBoost and LSTM. Comparison will be presented in the form of graphs, lists, and/or tables.
We seek to answer the question: Under the same circumstances, which method is more effective? How accurate is each method when it comes to peaks and outliers?

A. XGboost

Gradient boosting is a supervised learning algorithm that combines estimates from several simpler, weaker models in an attempt to predict a target variable accurately and minimize the error by working on the residuals. XGBoost [1] [2] Short for extreme gradient boosting is a robust machine learning algorithm and an extension to gradient-boosted decision trees, designed to improve speed and performance. XGBoost is widely used for feature selection because of its high scalability, parallelization, efficiency, and speed. Like in Gradient boosting, we fit our trees to the residuals of the previous trees' predictions. XGBoost builds these trees by calculating similarity scores between the observations that end up in a leave node. Using XGBoost, we can regularise our trees, which reduces the possibility of overfitting in the ensemble model [3].

B. LSTM

LSTM (Long Short-Term Memory) [4] is an advanced Recurrent Neural Network (RNN) based architecture that can process entire long-term sequences of data and is widely used in natural language processing and time series forecasting. Compared to typical recurrent neural networks, LSTM is capable of handling the vanishing gradient problem which causes a loss of information. In addition to contextual information within a sequence or series, the model can also convey information of a sequence output based on past and future contexts. LSTM learns the long dependencies of the inputs, captures significant features from the inputs, and preserves the information over a long period.

Fig. 1. illustrates the structure of a basic LSTM unit for calculating cells. An LSTM unit [5] generally consists of a memory cell, an input gate, an output gate, and a forget gate. Past information stored in the memory cell is just as significant as future information. The input and output gates enable the cell to store and retrieve information over an extended period of time. The input gate decides whether to add new information to the memory; the output gate decides what part of the LSTM unit memory contributes to the output. The forget gate is used to clear the memory in the cell. This gate captures the long-term dependency in time series since it decides which information will be discarded from memory [6].

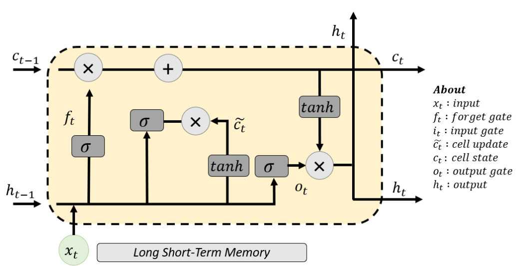

Fig. 1. Understanding the basics of LSTM-units [5]

II. Related Work

Over the last decades, there has been a lot of research directed at understanding and predicting the further. These research gives us many forecasting methods, most of them are relying on statistical models. Although these traditional statistical time series methods perform well on some forecasting problems, they have some inherent limitations due to the assumption of the statistical models. Some researchers tried to employ machine learning and deep learning techniques to forecast the time series. While prior studies primarily focus on spot [11] [12] or day-ahead [13] electricity price forecasting, little research addresses the long-term forecasting horizon [14] or even moving between short-term and long-term horizons [15].

In one study [8], a genetic programming system was presented that incorporated semantic awareness into the search process, as well as a local search optimizer to speed up the learning curve. Meanwhile, some other studies [9] [10] combined XGBoost and LSTM models to deal with time series data. They found that the resulting improvement is much higher than that of the original XGBoost single model, which maximizes the advantages of the two prediction models. Other studies [16] [17] that applied LSTM networks to time series prediction has found that LSTM models outperformed classification methods such as random forest, logistic regression, multiple kernel learning, and support vector machines.

Our study compares the performance of LSTM and XGBoost for time series forecasting on the same dataset. Similar to [9] we use XGBoost to obtain important features in training as well as studying the correlation between these features and use them in both methods to see which one gives a better result.

III. Method

A. tools

For this project, Python 3.7 was selected as the programming language. A number of software libraries were used in developing the Price prediction program, including Pandas and NumPy for handling data, Matplotlib and Seaborn for rendering graphs and plots, Xgboost and TensorFlow through Keras to develop a Timeseries prediction model, and Sklearn for calculating training metrics.

B. Dataset

We evaluated our proposed method using a dataset collected from the Nordpool energy exchange. The dataset contains information covering 01/01/2017 to 06/08/2021 and has 40,059 total observations. We randomly split the data set into two groups, approximately 70% for training and 30% for testing.

C. Parameter Analysis

A preprocessing of the data is done to determine if there is a possible seasonality in the time frame that can be used for training. This is based on months, hours of the day, days of the week, and months of the year. A series of box plots were created using the Seaborn library. These plots showed the average and mean values for all data from 2017 to 2021 in those featured timeframes.

Upon viewing the plot, I realized there is in fact a pattern in how prices change. For example, we can see in Fig. 2. prices tend to be higher in January and then moderately decrease afterward. There were more days with higher prices in July than the month before. In addition, according to Fig. 3. prices begin to peak at 8 am and then peak again at 6 pm during the day. Considering this information, I have added columns for these time splits to my data.

In addition, I examined the correlations between prices across the cities. This is to see if these price are in any means related and if they have an impact on one another. Figure 6 shows a heatmap of the correlations between different cities and we can see from this figure that the prices in Oslo and kristiansand are highly correlated with eachother and as well Bergen and SYS are correlated with krisitansand.

A benefit of using gradient boosting is that after the boosted trees are constructed, it is relatively straightforward to retrieve importance scores for each attribute. We used XGBoost to select feature importance based on the F-score value. Fig. 4 and Fig. 5 presents some important features selected based on XGBoost. We realized the important features or region in the case of our dataset in predicting the price for No1 and No2.

Generally, importance provides a score that indicates how useful or valuable each feature was in the construction of the boosted decision trees within the model. The more an attribute is used to make key decisions with decision trees, the higher its relative importance. This importance is calculated explicitly for each attribute in the dataset, allowing attributes to be ranked and compared to each other.

Importance is calculated for a single decision tree by the amount that each attribute split point improves the performance measure, weighted by the number of observations the node is responsible for. The performance measure may be the purity (Gini index) used to select the split points or another more specific error function. The feature importances are then averaged across all of the the decision trees within the model. [7]

I used feature Importance to evalute which cities from the correlation map and the data split data to use for the the training. From Figure 4 and 5 I found out that Oslo, Bergen, Tromsø, and DK1 prices would play a crucial role in predicting prices in Kristiansand, while Kristiansand, Bergen, SE3, SE1, SYS, and DK2 prices could be vital in predicting prices in Oslo. Also I noticed that the month has the highest influence on predicting prices. However, other time variables are so small that they can be ignored.

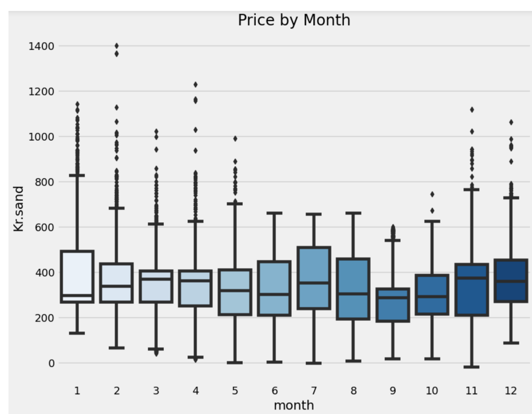

Fig. 2. Price per month(2017-2021)

To evaluate the ability of the program to train price prediction models, a series of models were trained and evaluated. The accuracy, validation accuracy, loss, and validation loss for each epoch of each model, as well as the time expenditure, were tracked.

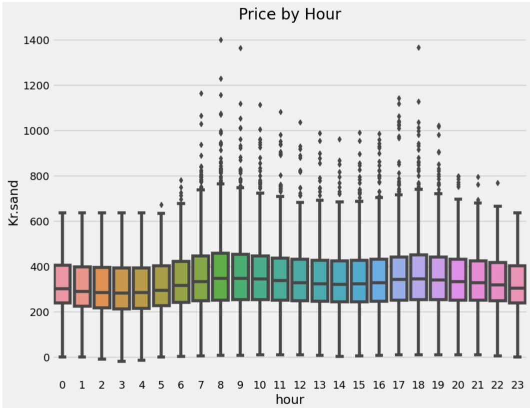

Fig. 3. Price per Hours of day(2017-2021)

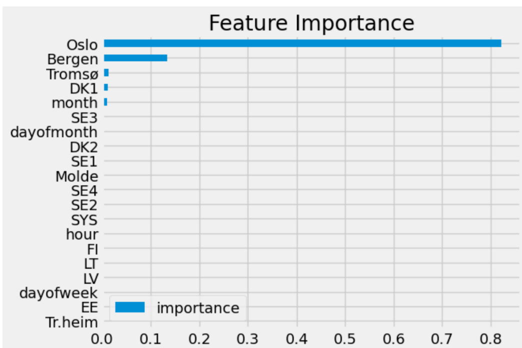

Fig. 4. Feature importance in kristiansand

D. XGBoost[1]

The XGboost [20]model offers three types of boosters: gbtree, gblinear, and dart; gbtree and dart use tree-based models, whereas gblinear uses linear functions. Nevertheless, decision trees are much

[1]The code for this part was inspired by Medallion Data Science channel on Youtube

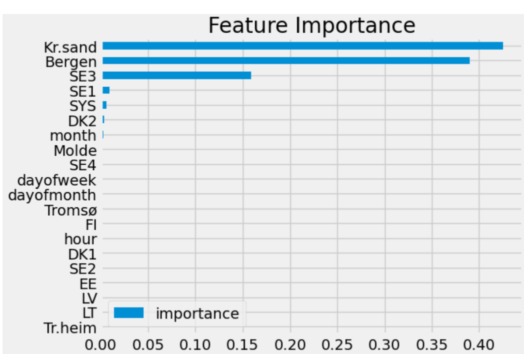

Fig. 5. Feature importance in Oslo

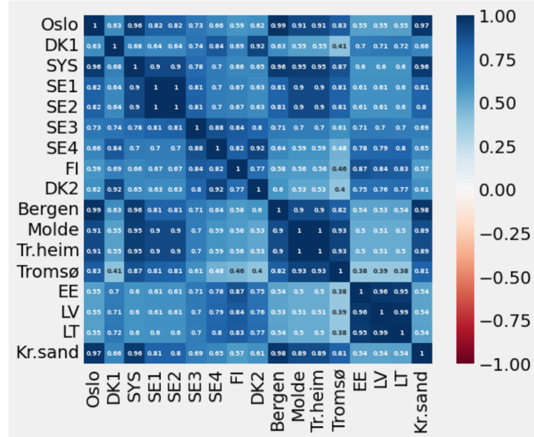

Fig. 6. Price per month(2017-2021)

better at detecting nonlinear relationships between predictors and outcomes. Dart Drop trees in order to solve the over-fitting and Training can be slower than gbtree because the random dropout prevents usage of the prediction buffer. However in this problem "gbtree" has been chosen as the booster type with 1000 estimators and a depth of 20 as Maximum depth of a tree, Increasing this value will make the model more complex and more likely to overfit. as well as "auto" as the tree method which uses heuristic to choose the fastest method, and squarederror as the training objective.

The objective determines the learning task, thus the type of the target variable. The available options include regression, logistic regression, binary, and multi-classification or rank. The objective is independent of the booster. As well as performing classification tasks, decision trees can also predict continuous variables with a certain degree of granularity for the data input range used in training.

in addition, When five epochs passed without any decrease in validation loss, the training halted early to save time and avoid overfitting. Oslo, Bergen, Tromsø, and DK1 and month has been chosen as features for predicting prices in Kristiansand, while Kristiansand, Bergen, SE3, SE1, SYS, and DK2 and months have been chosen for predicting prices in Oslo.

E. LSTM[2]

A sequential model with 4 layers, input containing 64 units and a dropout rate of 0.2, two hidden layers with the same size of units and a dropout rate of 0.2, and an output layer.

Same as the approach in XGBoost, When five epochs pass without any decrease in validation loss, the training halts early to save time and avoid overfitting. Training takes place only on essential features and time split for both targeted zones meaning Oslo, Bergen, Tromsø, and DK1 and month has been chosen as features for predicting prices in Kristiansand. In contrast, Kristiansand, Bergen, SE3, SE1, SYS, and DK2 and months have been chosen for predicting prices in Oslo.

With 100 epochs and a batch size of 15. In addition to the previous method, a window size of 72 hours has been chosen for training with LSTM. Therefore, we consider price changes over the past 72 hours when predicting each day. Fig. 7. shows a summary of the LSTM model.

```
Model: "sequential_13"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_45 (LSTM)              (None, 5, 64)             16896

 dropout_45 (Dropout)        (None, 5, 64)             0

 lstm_46 (LSTM)              (None, 5, 64)             33024

 dropout_46 (Dropout)        (None, 5, 64)             0

 lstm_47 (LSTM)              (None, 5, 64)             33024

 dropout_47 (Dropout)        (None, 5, 64)             0

 lstm_48 (LSTM)              (None, 64)                33024

 dropout_48 (Dropout)        (None, 64)                0

 dense_13 (Dense)            (None, 1)                 65

=================================================================
Total params: 116,033
Trainable params: 116,033
Non-trainable params: 0
_________________________________________________________________
```

Fig. 7. LSTM model Summary

[2]The code for this part was inspired by the code provided by Bernt A.Bremdal in Exercise 7 course DTE-3606-1 Artificial Intelligence and Intelligent Agents-project

IV. Result

The metrics defined to evaluate the results are the Root Mean Square Error (RMSE) and the Mean Square Error (MSE). Table I. Shows data comparing two models. Overall, XGBoost performed much better in prediction than LSTM. Additionally, we compared our proposed method using zones with a high correlation with our target for prediction and emphasizing the month feature with predicting without any additional information based only on the target zone and the date (marked as * in the table). Consequently, our proposed method of incorporating key features in training produced better results with both LSTM and XGBoost. In addition Table II. shows that the training and testing accuracy were really high for XGBoost in both zones.

Method	Zone	MSE	RMSE
XGBoost	Kristiansand	54.784413	7.40
XGBoost*	Kristiansand	558.533866	23.633321
XGBoost	Oslo	131.825531	11.48
XGBoost*	Oslo	1244.043454	35.271000
LSTM	Kristiansand	576.145	24.003
LSTM*	Kristiansand	1506.9924	38.82
LSTM	Oslo	2368.850	48.671
LSTM*	Oslo	3321.2169	57.63

TABLE I
RMSE AND MSE SCORE FOR EACH METHOD(*TRAINING WITHOUT USING FEATURES)

Method	Zone	test-set accuracy	train-set accuracy
XGBoost	Kristiansand	0.9979391	0.9999946
XGBoost	Oslo	0.99538234	0.99999865

TABLE II
ACCURACY OF TRAINING AND TESTING ON XGBOOST

Table III and IV indicates the 10 predictions with the highest error rate for each method in each zone and their date.

V. Discussion

The purpose of this research was to develop and evaluate two ML methods for forecasting Energy Prices using data from 2017 to 2021, dividing them into training and testing sets. As depicted in Figs 8-11, the XGBoost technique is a little more robust to outlier data than LSTM, where the first recognized

XGBoost		LSTM	
Date	MSE	Date	MSE
2021-04-06	71.173193	2021-02-11	233.760457
2021-06-14	51.259312	2021-01-08	183.871190
2018-03-28	49.958292	2019-01-24	133.895147
2021-06-26	42.158044	2021-01-14	125.654565
2021-06-15	38.322486	2018-11-26	109.964041
2021-06-20	36.727661	2021-02-10	100.573517
2018-03-16	33.232669	2018-12-14	96.285124
2021-06-28	30.615146	2021-06-20	95.908450
2021-06-16	28.037740	2021-03-08	91.911764
2021-02-06	27.855091	2021-06-19	90.272919

TABLE III
10 DATE WHERE THE MODEL MADE THE MOST ERRORS IN KRISTIANSAND

XGBoost		LSTM	
Date	MSE	Date	MSE
2018-03-02	108.466845	2021-02-11	549.842058
2018-03-01	79.362670	2021-02-01	380.811659
2021-02-12	75.858233	2021-02-05	333.166062
2017-01-05	64.132749	2021-02-12	307.405128
2021-02-11	63.420723	2018-03-01	231.306767
2021-04-06	57.300537	2021-01-08	149.499241
2017-08-31	41.892275	2021-03-08	135.329201
2018-08-30	35.951526	2019-01-24	119.753210
2021-04-16	31.430914	2018-11-26	116.316348
2021-04-15	29.102265	2019-01-31	94.321127

TABLE IV
10 DATE WHERE THE MODEL MADE THE MOST ERRORS IN OSLO

the peak occurrences better. Considering that these occurrences are highly likely to happen in the future, real systems must be prepared to face the input data with uncommon values. The XGBoost model presented better metrics values and accuracy for almost all of the years. This means that under normal conditions and even with larger error values during peak occurrences, its metrics are superior to those of LSTM. However, the use of deeper layers and more time steps could improve results in

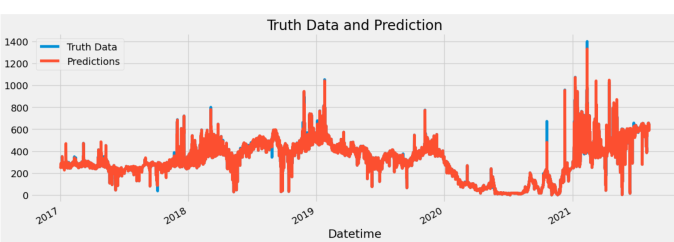

Fig. 8. kristiansand Truth data VS Prediction - XGBoost

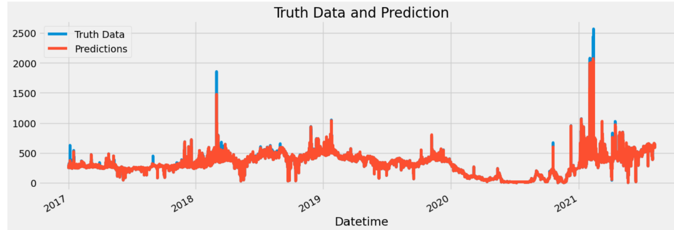

Fig. 9. Oslo Truth data VS Prediction - XGBoost

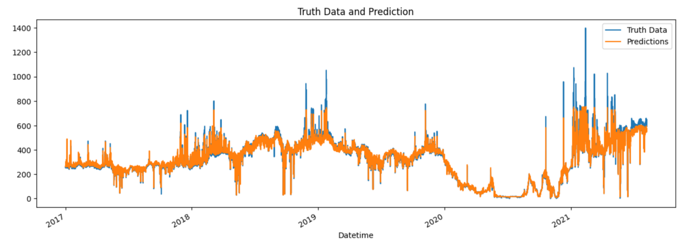

Fig. 10. Kristiansand Truth data VS Prediction - LSTM

LSTM by more effectively generalizing the data. In addition XGBoost was faster in training compared to LSTM.

As we saw in the results section, incorporating features into our training process produced better results, and I believe for future work adding even more features, such as inflation rates, economic shocks, weather, and temperature, electricity consumption, the efficiency of powerhouses, and fuel price could improve our result and increase its accuracy even further. In addition we can test if integrating prices only from 24 hours before our prediction target makes an improvement to the results as well.

VI. Conclusion

The development of intelligent systems that can predict future energy prices could aid the world

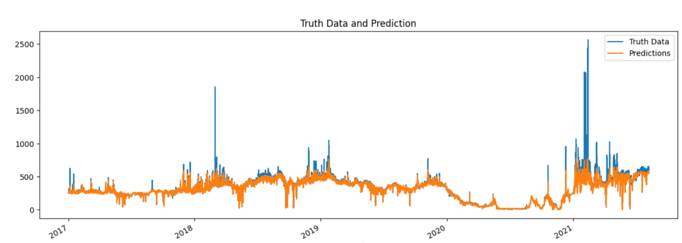

Fig. 11. Oslo Truth data VS Prediction - LSTM

in making better preparations for the future. In this paper we examined two well-known machine learning methods, the LSTM and the XGBoost, for forecasting the electricity price in Kristiansand and Oslo. To validate the performance of both methods, a data set containing energy prices in Europe (2017-2021) was provided by the Nordpool exchange. The indicated features were electricity prices in neighbouring cities and the month of the year. The input data is divided into train and test data. According to the train data, the XGBoost, and LSTM models have been tuned for all their parameters and the most optimal coefficients are identified. For determining the rate of error related to each of the XGBoost, and LSTM models estimations and selecting the most preferred model(s), the rate of forecasted output of each model is compared with its actual data. XGBoost has higher precision than the other approach, making it a recommended method for electricity price estimation in the targeted zone.

REFERENCES

[1] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. of the 22nd ACM SIGKDD Int.

[2] Y. Wang and X. S. Ni, "A XGBoost risk model via feature selection and Bayesian hyper-parameter optimization," International Journal of Database Management Systems (IJDMS), vol. 11, no. 1, pp. 1–10, 2019.

[3] Bernt A. Bremdal, "Boosting – getting the hyper parameters right," UiT campus Narvik 2022.
Conf. on Knowledge Discovery and Data Mining, pp. 785–794, 2016.

[4] The Value of LSTM in Time Series Forecasting, https://www.predicthq.com/events/lstm-time-series-forecasting

[5] Gers Felix,A.,Jurgen, S.Cummins,F.(2000), 'Learning to forget: Continual prediction with lstm', Neural computation 12(10), 2451-2471

[6] Hochreiter, S. Schmidhuber, J. (1997), 'Long short-term memory', Neural computation 9(8), 1735-1780.

[7] "Relative Importance of Predictor Variables" The Elements of Statistical Learning: Data Mining, Inference, and Prediction, page 367.

[8] Mauro Castelli, Aleš Groznik and Aleš Popovič , "Forecasting Electricity Prices: a Machine Learning Approach",8 May 2020

[9] He Wei, QingTao Zeng, "Research on sales Forecast based on XGBoost-LSTM algorithm Model"

[10] Pham Hoang Vuong, Trinh Tan Dat, Tieu Khoi Mai, Pham Hoang Uyen and Pham The Bao , "Stock-Price Forecasting Based on XGBoost and LSTM", 4 May 2021

[11] Crespo Cuaresma, J.; Hlouskova, J.; Kossmeier, S.; Obersteiner, M. "Forecasting electricity spot-prices using linear univariate time-series models", Appl. Energy 2004, 77, 87–106.

[12] Nowotarski, J.; Raviv, E.; Trück, S.; Weron, R. An empirical comparison of alternative schemes for combining electricity spot price forecasts. Energy Econ. 2014, 46, 395–412.

[13] Conejo, A.J.; Plazas, M. A.; Espinola, R.; Molina, A. B. Day-ahead electricity price forecasting using the wavelet transform and ARIMA models. IEEE Trans. Power Syst. 2005, 20, 1035–1042.

[14] Azadeh, A.; Moghaddam, M.; Mahdi, M.; Seyedmahmoudi, S. H. Optimum Long-Term Electricity Price Forecasting in Noisy and Complex Environments. Energy Sources Part B Econ. Plan. Policy 2013, 8, 235–244.

[15] Pao, H.-T. Forecasting electricity market pricing using artificial neural networks. Energy Convers. Manag. 2007, 48, 907–912.

[16] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," European Journal of Operational Research, vol. 270, no. 2, pp. 654–669, 2018.

[17] Z. Jin, Y. Yang and Y. Liu, "Stock closing price prediction based on sentiment analysis and LSTM," Neural Computing and Applications, vol. 32, pp. 9713–9729, 2020.

[18] "XGBoost Parameters" , https://xgboost.readthedocs.io /en/stable/parameter.html

[19] "Medallion Data Science", https://www.youtube.com/ channel/UCxladMszXan-jfgzyeIMyvw

[20] Bernt A.Bremdal,"LSTM TrafoExercise.py"