Short-Term Electricity Price Forecasting Using ARIMA and Transformer Model

Project Report

Junpeng Chen 03723009
Han Liang 03737290
Yicong Li 03736435
Jiaxin Yang 03746531
Runyao Yu 03737108
Xuyang Zhong 03747664
Yuqicheng Zhu 03741887

September 3, 2021

Motivation

In the background of the global marketization of electricity, price forecasting is critical for all market participants. The motivation for forecasting electricity prices has political and economic reasons. A power market company can adjust its bidding strategy and production or consumption schedule to reduce the risk or maximize the profits in day-ahead trading [5]. If grid operators know the price of electricity in advance, they can plan and avoid building unnecessary power plants and transmission infrastructure, which will contribute to environmental protection.

However, electricity price forecasting is a complex task, even if in the short term. The power market can become highly unpredictable due to many unforeseen issues. Many factors can affect electricity prices, such as weather (sunshine, temperature, wind speed, precipitation, etc.) and the intensity of business and daily activities (peak vs. off-peak, weekdays vs. weekends, holidays, etc.) [4]. These specific characteristics lead to electricity prices exhibiting seasonality at daily, weekly, and annual levels, as well as sudden price spikes. There are no clear patterns to describe the trend of electricity prices. However, it has inspired many researchers to invest more efforts in this field. Based on this, we explore how to predict electricity prices in this project, and the predicted city is set in Leipzig.

---

1 Project Description

In the field of machine learning, the performance of two parts which are represented by traditional machine learning and deep learning, are often compared.

Traditional machine learning model has more straightforward and more interpretable structures. When the scale of training data is relatively small, its generalization performance may be better than that of the deep learning model. However, the researcher needs to pay more effort to make manually feature engineering. Contrarily, the deep learning model consists of many layers of the neural network, such as CNN, LSTM, and Transformer. However, it needs a larger scale of data to train and has a high risk of overfitting. Thanks to the strong capability of representation learning, hand-engineering features are not required anymore, but it also results in the curse of interpretability.

In the project, our group wants to predict Leipzig's electricity price in the short-term period (i.e., 1 hour ahead, 1 day ahead, and 1 week ahead). We initially tended to observe the performance by utilizing simple models such as Ridge Regression and Random Forest. However, their performance was barely satisfied as expected. Therefore, we decided to use ARIMA and Transformer to represent traditional machine learning and deep learning, respectively. The forecast models, developed through historical data, are evaluated and verified by the symmetric mean absolute percentage error in the actual case application, i.e., the electricity prices in Leipzig. The prediction using these two models will be shown on the web interface.

Research Question

What are the advantages and disadvantages of traditional machine learning models (represented by the ARIMA model) and deep learning models (represented by the Transformer model) in forecasting short-term electricity prices?

Goals

The symmetric mean absolute percentage error (1 week ahead) for both models should be lower than 10. Besides, we also want to reduce the model complexity and training time as much as possible. Moreover, we hope the accuracy of cross-validation and test set can be as high as possible, indicating better fitting and generalization performance.

2 Data Basis

Data is the most fundamental ingredient of our analysis. The quality of data will directly affect the performance of the final model. It is necessary to list all potentially relevant impact factors before starting the data analysis.

To find impactful features, we create a factor table to summarize possible relevant features, merge these data collected from different sources, explore and analyze the correlation between various features, and finally preprocess the data before training it.

Sources and Data Collection

Through literature review and discussion in the group, we summarized possible features and corresponding sources preliminarily. The historical spot price in Leipzig is the most correlated feature to predict the future spot price, and the historical spot price can obtain from Montel API through corresponding token and correct indexing. Besides, electricity generation and consumption significantly impact electricity, so we collect the data of total power and residual power consumption for electricity consumption, electricity power generated by wind onshore, fossil brown, hard coal, hydro pumped storage, and photovoltaics, from the website smard.de. We also observe that photovoltaics account for a large proportion of total generation. Weather, especially radiation, influences the generation of photovoltaics. Hence, introducing weather-related features, such as air temperature at 200cm high and 20cm high, the temperature at 5cm and 20cm away from the ground, humidity, wind speed, and rainfall, into our research is necessary. It should be noted that the weather features are collected from Leipzig. The above features are collected from 01.01.2015 to current, but they are always updated in real-time. To keep the timeliness of data, we need to download the latest data regularly.

From the statistics aspect, the correlation between the spot price and different features was analyzed. Spearman coefficient [7] is utilized to quantify the correlation in this project. The coefficient ranges between -1 and 1. 0 denotes there is no correlation between the two variables. From that, we can exclude features uncorrelated to the label to ease the model training.

Preprocessing

Data Merge

Since the features are collected from different databases, for model training, it is necessary to merge them into one file. Considering the different sampling rates of features, we need to downsample or upsample them in hours.

Missing Value Processing

In addition, we also need to deal with missing values in the spot price dataset. We observe that there exist missing values in the period of 2:00-3:00 at the end of March every year. For these single missing values, we fill them with the next observation. Besides, for the 24 continuous missing points of the entire March 28th in 2021, we use the simple autoregression model to generate them.

Feature Engineering

To improve the predictive accuracy of the ARIMA model, we need to do feature engineering, which includes feature extraction, feature selection and interpretation. For feature extraction, primitively, we applied PCA on original features collected from different sources and compared with the feature extracted from spot price. We decided to use a scrolling window with a size of 24 to firstly divide the spot price into many short sequences. Then we used the Python package "tsfresh" to extract 786 features in each window. After that, we selected the features in two stages, respectively. In the first stage, we used Kendall rank correlation coefficients with a significance level of 0.05 to filter out uncorrelated variables. After this, we continued to use Lasso regression, keeping only the features with non-zero coefficients. At last, we merely used the top 17 features. To prove that the new features extracted from the spot price still have practical significance, we applied the correlation matrix (Figure.1) to observe the relationship between them and the exogenous variables. In the correlation matrix, we found that the features extracted by the Fourier transformation and wavelet transformation and other basic characteristics of the time series such as the mean value, the second derivative mean value, and RMS of the time series are positively correlated with the exogenous features which indicates the seasonality of the spot price plays a vital role.

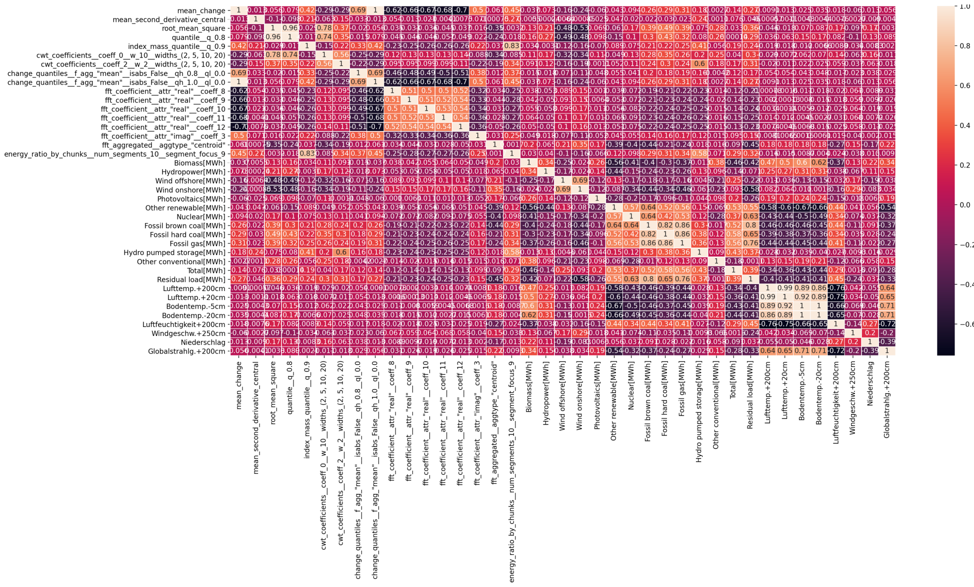

Figure 1: Correlation between features extracted from spot price and external features like weather condition etc.

3 Data Model

In terms of the traditional machine learning method, the well-known ARIMA, short for 'Auto Regressive Integrated Moving Average', with previous values of the time series and exogenous variables, was used for our forecasting. For comparison, the Transformer model representing deep learning likewise is implemented in our project.

Approach

Many research papers for time series forecasting problems used ARIMA as a basic model. For example, the autoregressive integrated moving average (ARIMA) models [1], [6] have been tested in the Spanish and the Norwegian markets for short-term forecasting. ARIMA is a class of models that 'explains' a given time series based on its past values, that is, its lags and the lagged forecast errors, so that equation can be used to forecast future values. The main advantage of ARIMA forecasting is that it requires data on the time series in question only.

Transformer [9] is a deep neural network model based on a self-attention mechanism. It has also been proved as a state-of-the-art model in many fields, such as NLP [8], CV [2] and Time Series Forecasting [10]. Compared to the traditional machine learning methods, Transformer does not need too much preprocessing to input data. Thus, it can be seen as an end-to-end model, which facilitates deployment. Besides, thanks to its enormous parameters, Transformer has a much stronger learning capability than traditional machine learning models. For the sake of time series forecasting, LSTM [3] is also a representative model, which is a kind of recurrent neural network. However, RNN often suffers from long-term dependency problems. Nonetheless, Transformer computes the attention of each timestep to the rest timesteps, which tackles this problem.

Training

We attempt to find the maximum effectiveness of the ARIMA model. There are two training processes involved in the ARIMA model. In order to select a suitable number of features, we need to find the best hyperparameters in the Lasso regression. During the training process, we gradually increased the value of alpha and found that the performance of our model is almost the same until the alpha equals 2, and then the performance will decrease. In order to avoid overfitting and shorten the prediction time, we set the value of alpha equals two at the end. The length of the training set can also significantly influence the performance of the ARIMA model. We use sMAPE (symmetric mean absolute percentage error) as a performance metric and found that the performance of the ARIMA model with the training length of 2 months is the best.

Our Transformer is implemented by PyTorch, MSE and Adam are selected as the loss function and optimizer, respectively. The input is a historical price series, like $[x_1, x_2, \ldots, x_N]$, and we require to predict a k long series after the Nth time step, namely $[x_{N+1}, x_{N+2}, \ldots, x_{N+k}]$. However, the model only predicts a series one time step ahead at one time, namely $[x_2, x_3, \ldots, x_{N+1}]$. Then we use $[x_2, x_3, \ldots, x_{N+1}]$ to predict $[x_3, x_4, \ldots, x_{N+2}]$. This process repeats k times until obtaining a complete prediction series.

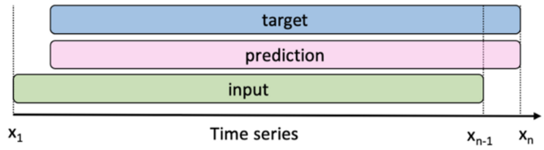

Figure 2: Training of Transformer

While training, we can always feed the model true values at each step. This method is called teacher forcing. By teacher forcing, the model only has to learn how to predict one timestep in advance. However, during inference, the model now must predict longer sequences. To tackle this problem, we stochastically replace the true value with the predicted one as the input of the model at each new prediction step, and the probability increases as the epoch grow. The detailed hyperparameters are summarized in Table 1.

Table 1: Hyperparameters of Transformer

Transformer	
Input dimension	7
Number of layeres	3
Number of head	7
Dropout rate	0.1
Training	
Number of epochs	500
Batch size	30
Input length	72
Forecast window	24
Loss: MSE	
L2 weight decay	1e-5
Optimizer: ADAM	
Learning rate	1e-3
Beta1	0.9
Beta2	0.999

Evaluation

In order to provide a statistically robust evaluation of the ARIMA model, we used time-based cross-validation, which forms a type of “sliding window” validation strategy (Figure.3). This method slides a window over the training samples while using several future samples as a test set. To obtain a meaningful performance estimation of the future spot price in September 2021, the folds of cross-validation are uniformly distributed in the period 2021.

Due to the limitation of computer resources and time, we cannot do cross-validation for Transformer, so we just simply split the dataset into a training set and test set. We randomly select several

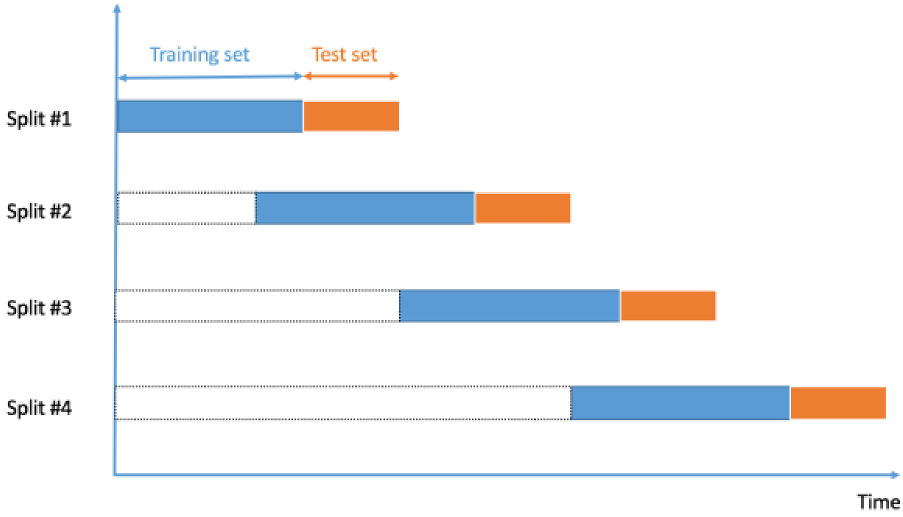

Figure 3: "Sliding Window" cross-validation

sequences in the test set to evaluate the performance of the model. MSE and sMAPE are used as the metrics for evaluation. To predict the electricity price series with different lengths, we have tried different corresponding input lengths. The following gives the best performance. To predict 1 hour, 1 day, and 1 week, the input length is 6h, 18h, and 168h, respectively.

4 Results

The forecast results for Leipzig's short-term electricity prices using the two models are displayed on the web interface. As shown in the graph, the ARIMA model performs pretty well on 1 hour ahead and 1 day ahead forecasts, but not on long-term forecasts. Transformer also exists this problem. In addition, both two models can't provide meaningful forecasts in special cases, such as when the spot price is negative.

Observations

After collecting nearly 40 features, we tried to discover the correlation between each feature. In the correlation matrix, we find that there are roughly five features that have a significant positive correlation with historical electricity prices, such as fossil gas, residual load, fossil hard coal, etc. It was quite surprising for us to find that there is almost no correlation between weather and electricity prices. Since there is no automatic update of the feature available in this project, we try to extract the feature from the historical electricity price. After feature engineering, we finally found the most relevant 17 features which can improve the ARIMA model prediction accuracy.

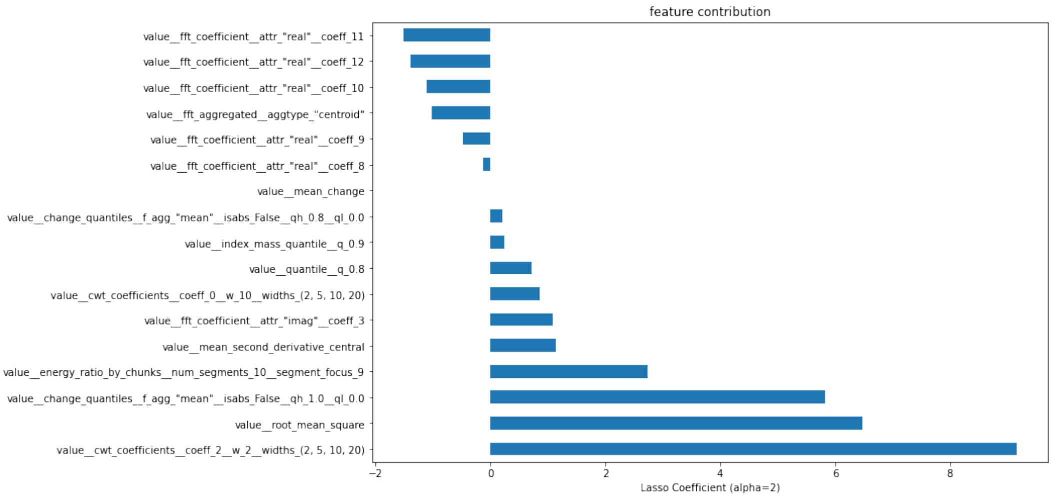

Figure 4: Remaining features in ARIMA model

Another surprising finding is that a longer training set does not provide better model performance in our case. (Figure.5) The reason could be that we did not consider enough long-term features in our models. The highest performance of the ARIMA model was achieved when the length of the training set was two months of historical electricity prices. The concrete prediction results of ARIMA are shown in Figure.6 and Figure.7.

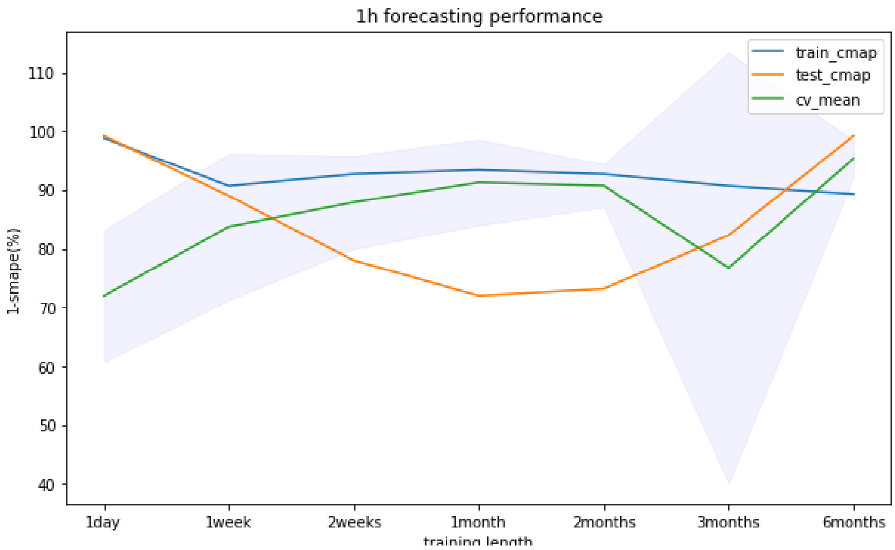

Figure 5: Training length experiments for ARIMA (cross-validation results for 1 hour prediction)

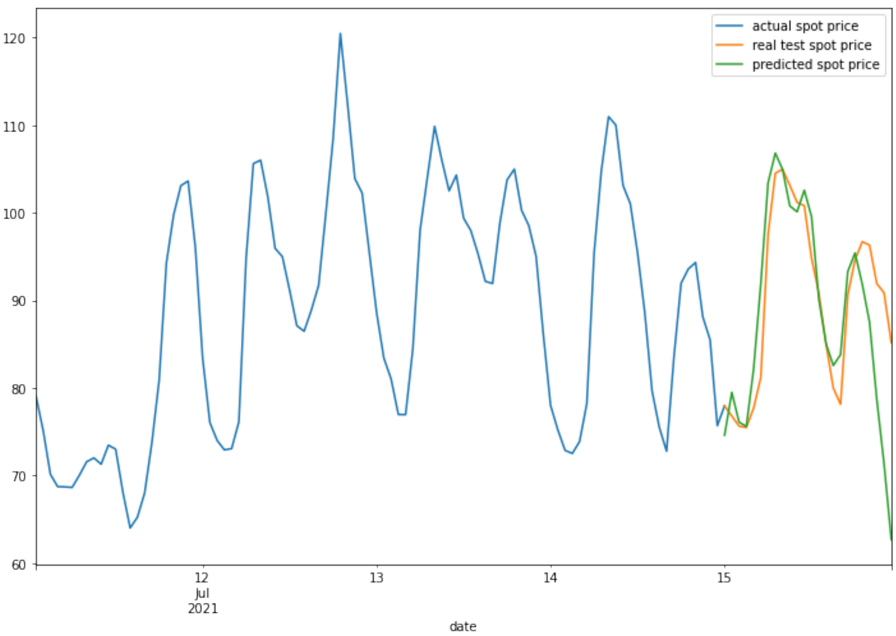

Figure 6: 1-day prediction of ARIMA

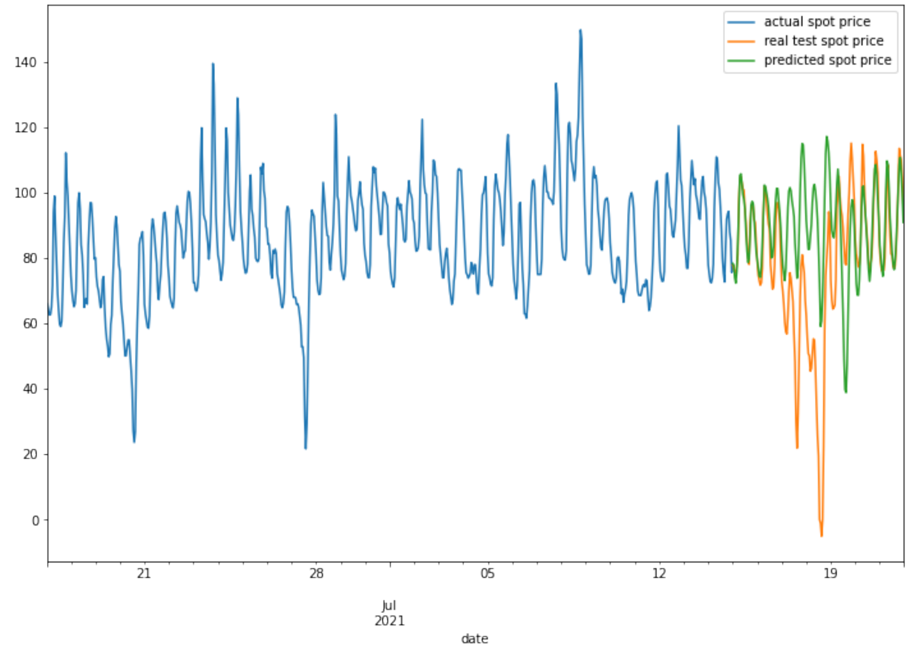

Figure 7: 1-week prediction of ARIMA

As for Transformer, so as to compare the model with single feature and multi-features, we use the data from the first 3 months of 2015 to train the model with 50 epochs. We find that the MSE of the spot-price-only model, i.e., 0.041 (computed from normalized value), is lower than that of the multi-feature model, i.e., 0.056.

Thus, we choose a single feature model, and we use spot price from 2015 to 2020 as the training set and the data of 2021 as the test set. However, the MSE is around 0.3 and sMAPE is around 50, which are both too high, compared with the result (MSE is about 0.2) of the model trained with the data of 2015 only.

The only difference between these two models is that for the model trained with 6-year-data, we also consider "year" as a positional embedding. Unlike month, day and hour, the year is not periodic, so we just scale the value of year to a value between 0 and 1 as the embedding.

However, when there are spot prices of neighboring years in the input sequence, or we want to use the prices at the end of a year to predict the prices at the beginning of the next year, the result would be unacceptable, we call this "cross-year problem". So as to avoid this problem, we try to use data consisting of at least 2 years as the training set, which may contain "cross-year" information. We use 2019 and 2020 as the training set, without "year-coding", we observe that its performance is better than both the 6-year model and the model trained with the data of 2015 only. For 1-hour, 1-day, and 1-week predictions, the average sMAPE is 6.73, 12.36, and 35.75. Its standard deviation is 4.03, 6.24, and 17.67, respectively. Due to the training set doesn't contain the spot price of 2021, which exists negative prices, so the model doesn't perform as well as we expect. The prediction results are shown in Figure 8 and Figure 9.

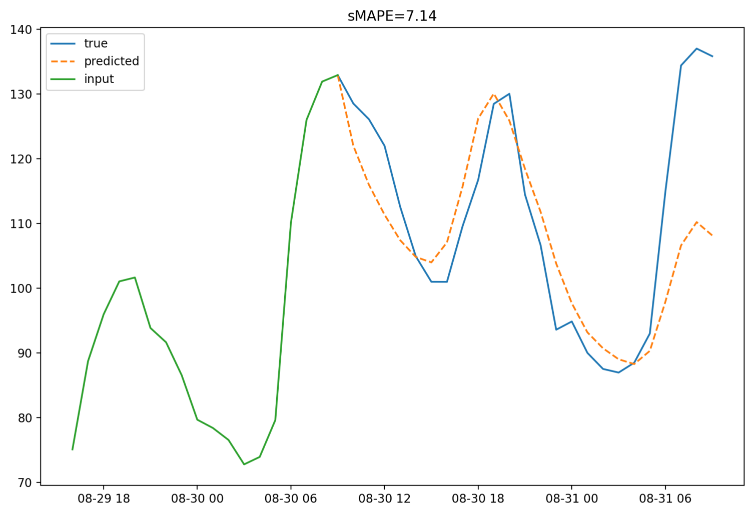

Figure 8: 1-day prediction of Transformer

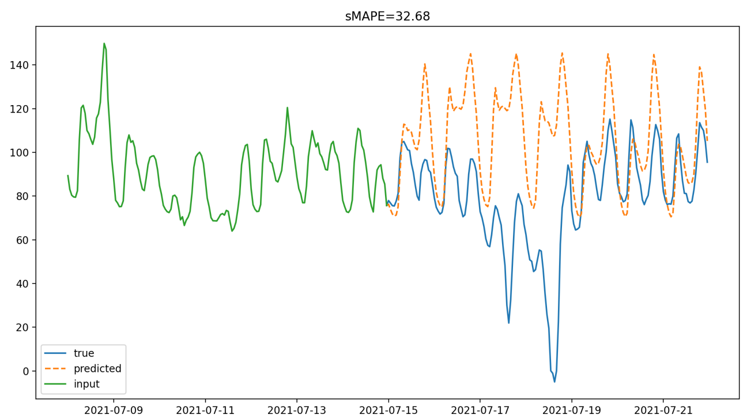

Figure 9: 1-week prediction of Transformer

Trends

ARIMA and Transformer models were used to forecast the electricity prices in Leipzig, and sMAPE was used as evaluation indicators. The result of the Transformer is not as good as we expected, and it is very time-consuming during the model training step. It is very inefficient to optimize the model iteratively and systematically. Because we are not able to improve the performance of Transformer significantly by directly adjusting the parameters, in this case, if we extract features in an intelligent way, ARIMA might have better performance in this specific forecast task. In terms of more efficient optimization iteration, interpretability, and robustness. The comparison between ARIMA and Transformer in terms of sMAPE is shown in Table 2.

Table 2: sMAPE comparison between Transformer and ARIMA

Prediction length	1 hour	1 day	1 week
ARIMA	7.7	8.7	12.4
Transformer	6.73	12.36	35.75

5 Discussion

In this section, we focus on discussing the possible reasons for the test results obtained from ARIMA and Transformer. Then, we assess both models from different perspectives. Lastly, we review the research questions proposed at the beginning.

Interpretation of Results

For ARIMA, we used the historical electricity prices and the extracted features from that as features. The accuracy of the prediction for the short-term as well as we expected. The conclusion that training time longer than 2 months does not improve the model performance also shows that a longer training set does not necessarily provide better model performance. Besides overfitting, another possible reason could be the complex underlying data mechanism of electricity prices [1]. Unexpected major events like the Coronavirus could change the underlying mechanism massively. The related policies for electricity prices could also be changed due to pandemics to keep the electricity price stable. Therefore, the historical prices long ago might not be useful. As we can see, the model performance of the ARIMA model can be slightly improved by adjusting the Lasso parameter (Alpha), which indicates the level of penalty. The larger alpha, the smaller the number of the remaining features. Because we extracted many features at first, which including some noise that might overfit the data, therefore reduction of the number of input features can not only avoid overfitting of the model but also reduce the prediction time of the model. Moreover, since all features used in the ARIMA model need to be computed in real-time, a large amount of data takes a lot of time. According to our results in Figure.9, a larger training set makes the model performance worse, and the model does not work for long-term prediction. Both facts indicate that we did not consider enough long-term features, which are also challenging to find out. A possible way to consider long-term factors is that we can extract more features with larger window sizes. For example, we can set the window size for 1 week or 1 month, even for 1 year, if we have more time in the future work.

For Transformer, the spot-price-only model performs better than the multi-feature model. One of the reasons may be that the multi-feature model not only predicts the spot price of the next time step but also needs to predict the other features. Multi-feature prediction is a much harder task than single feature prediction. Moreover, we use MSE as our loss function such that the error of spot price contributes the same as that of other features. The errors of other features can largely affect the optimization direction of parameters. The model performance is thereby decreased.

If we use the data from 2015 to 2020 to train the model, we will observe severe overfitting. One of the reasons might be that no regularization was utilized while training, which affects the generalization. Besides, as mentioned in the Result part, the model trained with the data of only 2015 is even better than that trained with 6-year data. Except for the scale of training data, the only difference is that we also consider "year" as a positional embedding for the model trained with 6-year data. Our initial intention is to introduce year embedding to solve the cross-year problem, but the year embedding is not a periodic coding like month and day, which may result in bad model performance. The reason is implied from the observation, and the deeper reason remains to be explored.

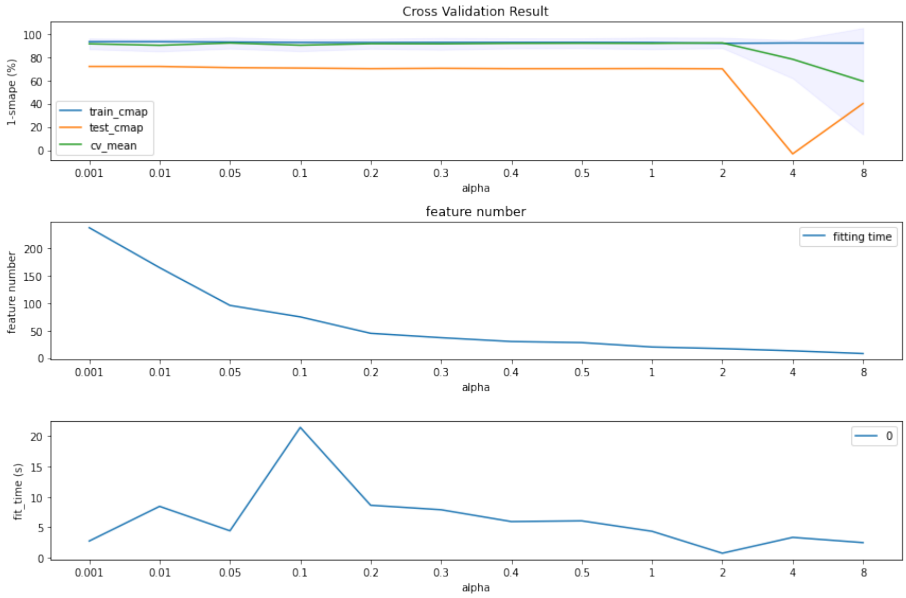

Figure 10: feature reduction experiment for ARIMA

Critical Assessment of Results and Assumptions

As the results shown, for ARIMA, its accuracy match our assumptions. It performs well in short-term forecasting but does not work as well in long-term forecasting. The results of long-term forecasting using the ARIMA method are only trend-based and do not predict the peaks well. There are two reasons for this result. First, we don't have enough long-term features. Second, we only have the actual exogenous features calculated from spot price in the cross-validation progress. However, we do not have exogenous features to forecast the spot price in the future. Our strategy is to predict the exogenous features using their historical values. Therefore, long-term forecasting requires accurate feature prediction, which is also very challenging. In addition, the ARIMA model does not provide good prediction results when there are anomalies in the electricity prices.

As for Transformer, its performance does not meet our expectations. Although 1-hour and 1-day predictions gave an acceptable result, its long-term prediction performs not well. When electricity price fluctuates sharply or there exists negative prices, the model cannot give a satisfying result, either. Nevertheless, since the training of Transformer has been completed offline, the waiting time for prediction results is shorter than ARIMA.

Proposed Answer to Research Questions

There are superior and inferior models, and the ARIMA advantage model is very simple. First, the ARIMA model has better interpretability. We can get information from the results of the model, such as which features have more effect on the spot price. Second, the ARIMA model has better generalization performance which can be seen from cross-validation results. In addition, the ARIMA model requires less computational cost. Therefore, it is more efficient to optimize iterations compared to Transformers. The advantages of the Transformer complement the ARIMA disadvantages, which need a lot of work to deal with the features. The features need to be calculated from the spot price in real-time, which is time-consuming during prediction. Besides that, the feature needs to be predicted if we do not have a spot price. Accurate prediction is needed to reproduce the performance of cross-validation, which is challenging. Conversely, the disadvantage of Transformer is that the parameter setting is complex.

The Transformer is a pure end-to-end model, which does not need manual feature engineering. The preprocessing is easy and can be integrated into the pipeline. Compared to ARIMA, Transformer was trained offline, the trained parameters are saved as a file, and the inference is relatively fast. However, Transformer is easy to be overfitted, so fine hyperparameter tuning is needed. The Transformer also has much higher computation complexity than traditional machine learning models, so, the training is time-consuming. Besides, due to its enormous parameters, the scale of the training dataset should be larger to get an acceptable generalization, which makes training even longer.

Conclusion

In this work, we presented ARIMA and Transformer to forecast Leipzig's electricity price.

For the ARIMA model prediction results of cross-validation, 1-hour, 1-day, and 1-week average sMAPEs are 7.7, 8.7, and 12.4, respectively. For the Transformer prediction results, 1-hour, 1-day, and 1-week average sMAPEs are 6.73, 12.36, and 35.75, respectively. The sMAPE values of ARIMA are smaller than Transformer, which proves that in this case, the ARIMA model is more suitable for electricity price prediction than Transformer.

Summary of Results

For ARIMA, the results showed that the multi-features model performs better than the single-feature model. Seasonality was not considered in the ARIMA model but the exogenous features. (e.g., Fourier transformation extract the seasonality features). 17 features are enough to capture the pattern of spot price in ARIMA. In terms of forecasting accuracy, it performs well in the short term but not so well on long-term forecasts. Then it cannot make accurate forecasts for special cases, such as the occurrence of negative electricity prices.

For Transformer, the results indicated that the spot-price-only model outperforms the multi-feature model; severe overfitting occurs when we used the data from 2015 to 2020 with year coding, the model trained with the data of only 2015 without year coding is even better, improper year coding may cause the problem; if we only use the model trained with 2015 data, "cross-year problem" will not be predicted in some specific cases; the model trained with 2019 and 2020 data solves "cross-year problem", its performance is also better than the model trained with 2015 data.

Overall, we found that although deep learning methods have strong representation ability, they require much more computational resources and experience on parameter tuning. Besides, it turns out it is easy to cause overfitting in our case. On the other hand, if we add more non-linearity into simple traditional models like ARIMA by appropriate feature engineering, they could even provide more stable results. Despite the huge trend of applying deep learning, in this project we realized that we should not completely abandon the traditional simple algorithms neither.

Future Work

For the design of the website, the user experience still needs to be improved. For example, when the user clicks on the 'predict' button, it takes a long time for the back-end data to be transmitted to the front-end. Besides, users do not know how long they have to wait. So in the future, we can design a more user-friendly interactive interface. What's more, due to our limited resources and time in this project, we are unable to download features from publicly available datasets that can be updated in real-time except for electricity prices. So it is important to have access to important features for future work.

In the feature extraction process, we use a scrolling window with a size of 24 to divide the spot price into many short sequences firstly. However, from the autoregression model of spot price, we can find that spot price variation also has a weekly and monthly pattern. In the future, we will consider increasing the size of the window to consider longer-term behaviors.

Due to limited time and computation resources, we could not explore the possibility of a Transformer enough. In the future research, we will try to optimize the model structure and training method further to get better performance.

6 Comments to Group Work Experience

This project lasted around three months. During this time, the members of our group work together for a common goal. In this project, we not only gained experience in feature collection, model training and website design but also learned how to improve the efficiency of teamwork.

To improve the prediction accuracy of the model, our team spent a lot of time collecting nearly 40 features that potentially affect electricity prices. However, we finally had to give up using these features because they were almost not updated in real-time. We tried to extract the features from historical spot electricity prices. The complexity of the training data will also make the model more difficult to interpret, which can be important when justifying real-world decision-making as a result of model outputs.

Apart from that, due to the complexity of the model itself, Transformer needs a large scale of the dataset as the training set, each training epoch would spend a lot of time, so we need to watch out for bug extremely. Meanwhile, due to a lack of computation resources, the experiments we could do are limited, which leads to slow optimization iteration.

To improve the efficiency of our group, the group leader assigned specific tasks to each member in different periods. However, in practice, we found that the workload of each member was hard to distribute fairly. We also found that systematic thinking is very important when dealing with problems. For example, in feature engineering, systematic planning can save a lot of time. And in the case of turning hyperparameters, pre-analyzing the impact of hyperparameters on the model saves time compared to randomly turning these parameters.

References

[1] Menzie David Chinn, Michael LeBlanc, and Olivier Coibion. The predictive characteristics of energy futures: Recent evidence for crude oil, natural gas, gasoline and heating oil. *Natural Gas, Gasoline and Heating Oil (October 2001). UCSC Economics Working Paper*, (490), 2001.

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[4] Debra Javeline. The most important topic political scientists are not studying: adapting to climate change. *Perspectives on Politics*, 12(2):420–434, 2014.

[5] Katarzyna Maciejowska and Rafał Weron. Forecasting of daily electricity prices with factor models: utilizing intra-day and inter-zone relationships. *Computational Statistics*, 30(3):805–819, 2015.

[6] Claudio Morana. A semiparametric approach to short-term oil price forecasting. *Energy Economics*, 23(3):325–338, 2001.

[7] Leann Myers and Maria J Sirois. Spearman correlation coefficients, differences between. *Encyclopedia of statistical sciences*, 12, 2004.

[8] Igor V Tetko, Pavel Karpov, Ruud Van Deursen, and Guillaume Godin. State-of-the-art augmented nlp transformer models for direct and single-step retrosynthesis. *Nature communications*, 11(1):1–11, 2020.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[10] Neo Wu, Bradley Green, Xue Ben, and Shawn O'Banion. Deep transformer models for time series forecasting: The influenza prevalence case. *arXiv preprint arXiv:2001.08317*, 2020.