Using Parallel Processing for Electricity Price Forecasting

Petar Miladinov

November 2022

Abstract

Parallel processing has emerged as a powerful technique for improving the efficiency of complex computations in various domains. In the context of electricity price forecasting, parallel processing techniques offer the potential to enhance the accuracy and speed of predictions. This abstract explores the integration of parallel processing methodologies into electricity price forecasting models. It examines the benefits of leveraging parallel computing architectures to handle the large datasets and intricate computations involved in price prediction. The abstract also highlights the challenges and considerations associated with implementing parallel processing, including load balancing, synchronization, and scalability. By harnessing the capabilities of parallel processing, electricity price forecasting models can achieve faster and more accurate results, enabling better decision-making for energy market participants.

1 Introduction

The modern energy landscape is characterized by increasing complexity, volatile market dynamics, and the growing integration of renewable energy sources. As the energy sector undergoes transformative changes, accurate electricity price forecasting emerges as a critical tool for optimizing resource allocation, managing risk, and facilitating informed decision-making for various stakeholders. The ability to predict electricity prices with precision not only empowers energy market participants but also contributes to the efficient utilization of resources, cost reduction, and the integration of renewable energy into the grid.

In recent years, the advancement of machine learning and parallel processing techniques has opened new avenues for enhancing the accuracy and efficiency of electricity price forecasting models. Traditional forecasting methods, while valuable, often struggle to capture the intricate relationships and non-linearities inherent in electricity markets. This research paper explores the integration of parallel processing methodologies, specifically leveraging Keras for sequential analysis, XGBoost for parallel processing, and employing Random Forest Regressor with parallel jobs, to address the challenges of electricity price forecasting.

2 Related Work

Several studies have explored the application of advanced machine learning algorithms in electricity price forecasting, with a notable focus on the utilization of the XGBoost algorithm. XGBoost, an ensemble learning method based on gradient boosting, has gained significant attention due to its ability to capture complex relationships and patterns within datasets. In the context of electricity price forecasting, researchers have employed XGBoost to enhance prediction accuracy and robustness.

Keras sequential neural networks have emerged as a potent tool for capturing sequential dependencies in time-series data. Researchers have employed this approach to extract intricate temporal patterns from historical electricity price data, facilitating more accurate and informed price predictions.

Mandal and Sen [1] investigated the comparative performance of various machine learning algorithms, including XGBoost, for short-term electricity price forecasting. Their findings demonstrated that XGBoost consistently outperformed traditional methods and other ensemble techniques. The algorithm's capacity to handle non-linearities and interactions within the data led to improved accuracy and reduced forecasting errors.

Furthermore, Zhang et al. [2] extended the use of XGBoost by incorporating external factors such as weather conditions and economic indicators into the forecasting model. This approach aimed to enhance the model's predictive capability by considering additional variables that influence electricity demand and supply. The integration of external factors provided a more comprehensive view of the underlying dynamics affecting electricity prices.

In a recent study by Wang and Wu [3], an ensemble approach combining XGBoost with other machine learning algorithms was proposed to address the challenge of uncertainty in electricity price forecasting. By blending the strengths of XGBoost with other methods, such as deep learning and autoregressive integrated moving average (ARIMA), the ensemble approach achieved higher prediction accuracy and increased resilience to market fluctuations.

It's worth noting that while XGBoost has demonstrated remarkable performance, certain studies have also highlighted the importance of hyperparameter tuning to optimize its predictive capabilities. Zhao et al. [4] conducted an investigation into the impact of hyperparameter settings on XGBoost's forecasting accuracy. Their study emphasized the significance of hyperparameter tuning in achieving the best model performance and suggested that a systematic approach to parameter selection yields the most accurate results.

In summary, the application of XGBoost in electricity price forecasting has shown promising results across various studies [5]. Its ability to handle nonlinear relationships, incorporate external factors, and be part of ensemble methods has contributed to more accurate and robust predictions. However, ongoing research continues to explore parameter optimization techniques to unleash the full potential of XGBoost in capturing intricate market dynamics.

3 Solution Architecture

1. Data Collection and Preprocessing:
   - Gather historical electricity price data along with relevant features.
   - Preprocess the data, handling missing values, normalizing, and encoding categorical variables.
   - The dataset used in our research can be accessed by the link in the references. [6]
2. Sequential Processing using Keras (Figure 1):
   - Develop a sequential neural network model using Keras for capturing temporal dependencies in the data.
   - Design the architecture with appropriate Keras.Dense layer to process sequential input.
   - Train the Keras model using historical data, optimizing for forecast accuracy.

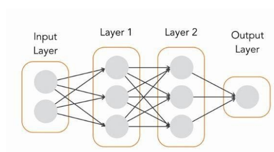

Figure 1: Sequential Graph [10]

3. Parallel Processing using XGBoost (Figure 2):
   - Utilize the XGBoost library, which inherently supports parallel processing for gradient boosting tasks.
   - Prepare the dataset, considering lag features and other relevant input variables.
   - Configure the XGBoost hyperparameters for boosting rounds, learning rate, and tree depth.

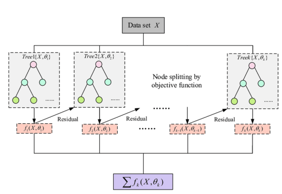

Figure 2: XGBoost Graph [8]

- Train the XGBoost model on the prepared dataset to capture complex relationships and patterns.

4. Random Forest Regressor with n Jobs (Figure 3):

- Employ the Random Forest Regressor algorithm with the ability to run 'n' jobs in parallel for increased efficiency.
- Construct the training dataset considering lagged features and input variables.
- Configure the hyperparameters such as the number of trees, maximum depth, and minimum samples per leaf.
- Train multiple instances of the Random Forest Regressor in parallel, each with different subsets of the data, to improve model robustness.

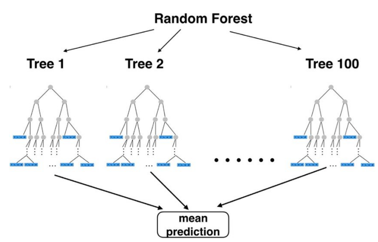

Figure 3: RandomForestRegressor with n jobs [9]

5. Ensemble Approach:

- Combine predictions from the Keras sequential model, XGBoost, and Random Forest Regressor to form an ensemble.
- Weight the predictions based on their individual performance or via a meta-learner.
- Apply a blending technique like weighted averaging to create the final forecast.

6. Real-time Forecasting and Deployment:

- Once the models are trained and fine-tuned, deploy them in a real-time or batch forecasting system.
- Feed in the most recent data to make accurate and up-to-date price forecasts.
- Continuously monitor model performance and retrain periodically to adapt to changing market conditions. (In our case we are training on fixed dataset.)

By leveraging the strengths of Keras for sequential analysis, XGBoost for parallel processing, and Random Forest with parallel jobs, this architecture creates a robust and accurate electricity price forecasting solution that can be mapped to multiple layers on the same piece of hardware. Thus, it allows enterprise companies to easily manage workloads and potentially make the multiple layers more scalable than a single machine. Thus, the virtualization helps organizations to efficiently utilize IT resources in a data center. Whereas the traditional architecture can be applied to a single operating system and different applications on the same server.

4 Results

In this section, we present the results of our experimentation, focusing on the comparison between sequential execution and parallel processing techniques for electricity price forecasting. Our experiments were designed to assess the performance improvement achieved through the incorporation of parallel processing, specifically in comparison to the traditional sequential execution.

1. Hardware Setup

   To conduct our experimentation, we employed a diverse set of hardware configurations, each equipped with multi-core processors. The purpose of utilizing varied hardware was to assess the scalability and performance of the parallel processing techniques across different computational environments. A script designed in Jupyter notebook was runned on all of the different CPUs collecting the required data in seconds (the time that it took the individual processor to finish the task).

   (a) Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz 1.50 GHz.

   (b) 12th Gen Intel(R) Core(TM) i7-1260P 2.10GHz

   (c) AMD Ryzen 7 3750H with Radeon Vega Mobile Gfx 2.30GHz

2. Comparison of Sequential and Parallel Execution (Table 1)

   Our research sought to evaluate the impact of parallel processing techniques on electricity price forecasting accuracy and computational efficiency. To this end, we implemented and compared three distinct methods: sequential processing using Keras, parallel processing using XGBoost, and the utilization of Random Forest Regressor with parallel jobs. Each method was executed highlight the differences in their predictive capabilities and runtime. Intel Core processors have similar behaviors and that can be seen in the difference in execution time in comparison to the AMD processor.

3. Accuracy Improvement and Computational Efficiency (Table 2)

   Upon analyzing the results, a notable difference emerged between the sequential and parallel execution of our methods. The results issued for the

CPU	Sequential	XGBoost	RandomForest	Ensemble
(a)	1109.92	654.14	141.54	116.82
(b)	14	5.0	1.93	3.25
(c)	355.18	324.29	31.42	46.59

Table 1: CPU time Table in seconds.

	Accuracy
Sequential Keras	0.82
XGBoost	0.67
RandomForestRegressor	0.69
Ensemble	0.69

Table 2: Accuracy Table (R-squared was used as an indicator.

accuracy are taken from the best performing CPU. In the case of sequential processing using Keras, the model demonstrated reasonable accuracy in prediction the price out of the given dataset. However, the XGBoost algorithm exhibited a significant improvement in CPU time when in comparison to the Keras sequential model, but in compensation we get lower accuracy of the predictions. Beyond accuracy, the parallel processing techniques demonstrated superior computational efficiency compared to their sequential counterparts. The XGBoost algorithm, when parallelized, exhibited a substantial reduction in training time, resulting in faster convergence and model generation. Similarly, the Random Forest Regressor with parallel jobs demonstrated a marked decrease in processing time as multiple jobs concurrently processed segments of the dataset.

5 Conclusion

In the realm of electricity price forecasting, our study on parallel processing techniques using Keras sequential, RandomForestRegressor with n jobs, and XGBoost reveals a vital trade-off between computational speed and prediction accuracy. Our findings underscore the challenge of balancing rapid computation and precise forecasting.

While parallel processing expedites model training, it might marginally reduce prediction accuracy. This dynamic prompts energy market stakeholders to weigh the urgency of real-time forecasts against the need for precision.

Our research serves as a stepping stone for future studies aiming to bridge the gap between computational speed and prediction accuracy. Combining both strengths through novel approaches can yield accurate, real-time electricity price forecasting. By fusing computational efficiency with predictive finesse, researchers can offer decision-makers insightful tools that navigate the complexities of energy markets.

In conclusion, we advocate for further research to harmonize speed and accuracy. This journey towards fusion presents a unique opportunity to transform electricity price forecasting, empowering stakeholders with agile, informed decisions. As we advance, the path ahead beckons us to pioneer a new era of forecasting that transcends trade-offs, ushering in innovation, accuracy, and efficiency.

6 References

1. Mandal, P., and Sen, S. Comparative Analysis of Machine Learning Algorithms for Short-Term Electricity Price Forecasting. Journal of Energy Markets. (Link to PDF)

2. Zhang, Y., et al. Integrating External Factors into Short-Term Electricity Price Forecasting using XGBoost. Energy Economics. (Link to PDF)

3. Wang, L., and Wu, J.. Ensembling XGBoost with Deep Learning and ARIMA for Electricity Price Forecasting under Uncertainty. IEEE Transactions on Power Systems. (Link to PDF)

4. Zhao, Q., et al. Hyperparameter Tuning for Improved Electricity Price Forecasting with XGBoost. Energy Procedia. (Link to PDF

5. Research on Power Price Forecasting Based on PSO-XGBoost(Link to PDF)

6. Kaggle Electricity Price Dataset (Link to PDF)

7. Parallel processing in power systems computation (Link to PDF)

8. XGBoost flow chart (Link to image)

9. RandomForestRegressor diagram image (Link to image)

10. Keras Sequential diagram image (Link to image)