Machine Learning Approaches for Electricity Price Forecasting: A Comparative Study

Aswathy Gopalakrishnan
Department of AI Engineering
Jönköping University
Jönköping, Sweden
goas24ee@student.ju.se

Jini Saroja
Department of AI Engineering
Jönköping University
Jönköping, Sweden
saji24ff@student.ju.se

Punnoose Kozhuppakalam Thomas
Department of AI Engineering
Jönköping University
Jönköping, Sweden
ktpu24fc@student.ju.se

Sneha Mathummal Parapurath
Department of AI Engineering
Jönköping University
Jönköping, Sweden
masn24nf@student.ju.se

Yonge Li
Department of AI Engineering
Jönköping University
Jönköping, Sweden
liyo23ac@student.ju.se

***Abstract*—Electricity consumption prediction is crucial for industrial companies to enhance energy efficiency and optimize their costs. This study explores four machine learning models—Polynomial Regression, Random Forest Regression, XGBoost Regression, and Artificial Neural Networks (ANN)—to predict the hourly electricity price for the company Habo Plast for the next 7 days. The models were trained on historical electricity prices, incorporating weather factors and electricity exports from Sweden. XGBoost demonstrated the best among the four models, outperforming others in terms of mean absolute error (MAE), root mean square error (RMSE) and R square score. The results show that XGBoost has the highest predictive accuracy as well as it is easy to deploy, which is good for Habo Plast to adjust its operations efficiently, and reduce electricity expenses.**

***Index Terms*—Electricity price prediction, Machine Learning, XGBoost, Polynomial, Random Forest, Regression, ANN**

I. Introduction

Electricity costs are a significant operational expense for manufacturing companies, and optimizing energy consumption is essential for cost reduction and sustainability. With fluctuating electricity prices influenced by various factors such as demand, weather conditions, and grid supply, businesses need reliable forecasting models to make informed decisions. Habo Plast–a manufacturing company which owns 16 injection moulding machines and 2 extrusion machines, and currently deliver over 1400 unique items aims to anticipate electricity costs for the upcoming week to adjust production schedules accordingly and minimize expenses.

Machine learning (ML) techniques have emerged as effective tools for time-series forecasting, offering enhanced accuracy compared to traditional statistical methods. This study evaluates four ML models—Polynomial Regression, Random Forest Regression, XGBoost Regression, and Artificial Neural Networks (ANN)—to determine the most suitable approach for predicting HaboPlast's electricity costs. These models were trained using Sweden's historical weather data(Air temperature, Precipitation, Snow depth, Sunshine time), day-ahead electricity price data, Sweden's electricity imports and exports, and additional time features—weekends, holidays, and lag features.

The experimental results indicate that XGBoost performs best in forecasting short-term electricity costs due to its ability to capture complex patterns and interactions in the data. By integrating this model into HaboPlast's decision-making process, the company can proactively adjust operations to optimize energy usage. This report presents a comparative analysis of the selected models and highlights the potential of machine learning in industrial energy cost forecasting.

II. Related Works

There exists many studies in Electricity Price Prediction using Machine Learning. Also, this has been a critical research area due to its impact on energy management, market efficiency and cost savings for industries and consumers. This section mainly focuses on an overview of existing works.

A. Related Work 1

IEEE paper for Electricity Price Prediction using Machine Learning.

This paper by K. Likitha Chowdary et al. examined four regression models—Random Forest Regression, Logistic Regression, Support Vector Regression (SVR), and Artificial Neural Network (ANN) Regressor [1]. They are using a dataset containing attributes like weather conditions, national load, and electricity demand. This study concluded that ANN performed best with the lowest Mean Absolute Error (MAE). This research explores the significance of machine learning in accurately predicting electricity prices.

B. Related Work 2

Forecasting Spot Electricity Market Prices Using Time Series Models.

In this paper Dawit Hailu Mazengia [2] focuses on forecasting electricity market prices using time series models. It also pinpoints the importance of price for market participants, including generators and consumers, to optimize their bidding strategies and operational planning. This study uses the multiple linear regression (MLR) to model electricity prices, considering historical price data and external factors. They use the seasonal variations like hydro reservoir levels, temperature as the key variables. Also, this study compares multiple linear regression models such as ARIMA models, Transfer Function models, and Dynamic Regression models, and shows that linear regression is effective in stable markets but it struggles with volatile ones.

C. Related Work 3

Wind and Electricity Prices in Sweden – A Statistical Analysis.

This paper authored by Rickard Sandberg [3], focused on the affects of wind conditions and temperature on electricity prices in Sweden across different bidding zones (SE1-SE4) using the Time series analysis. They have employed Time-Series Regression models using data ranging from September 2022 to August 2023 to examine these relationships and also compares it with both linear and dynamic models. The study concludes that the wind power has twice the impact on electricity prices compared to temperature. Moreover, the dynamic models explain up to 80% of price variations, making them highly effective for forecasting.

D. How Our Approach differs

This study aims to improve electricity price prediction by integrating diverse machine learning models and expanding feature selection. Unlike previous work by K. Likitha Chowdary et al., which identified ANN as the best model, we focus on models such as Polynomial Regression, Random Forest, XGBoost, and ANN. Our approach integrates three key datasets: Sweden's historical weather data, electricity price zones, and import/export market data. We also use advanced feature engineering techniques, including lag features, rolling averages, and weekend/holiday indicators, along with an in-depth correlation analysis to assess feature importance.

The study focuses on Sweden's SE3 region and analyzes data from 2019 to 2024, emphasizing long-term trends over short-term fluctuations. A unique aspect of this research is the incorporation of electricity import/export data, which explores how international energy trade affects domestic prices.

To ensure model robustness, Time Series Cross-Validation is applied, and hyperparameter tuning is used to optimize XGBoost and ANN models. Additionally, this study is specifically tailored to Habo Plast AB in Sweden's SE3 zone, differentiating it from studies focusing on markets like Nord Pool or Ontario.

So, this approach extends previous work by integrating diverse machine learning models, richer feature engineering, and customized Swedish market analysis. While previous studies focused on linear models or deep learning, this methodology balances accuracy, efficiency, and interpretability, making it more suitable for industrial applications like Habo Plast.

III. Background

The project utilizes four distinct machine learning models – Polynomial regression, random forest, XGBoost, and Artificial Neural Networks (ANN), for accurate forecasting of electricity prices.

A. Model Selection Criteria

The selection of the models was based on essential requirements that had to be fulfilled. Firstly, the nature of the input requires models that can effectively handle the dataset's continuity and complexity. The features were gathered from different data sources and exhibited indirect associations. Hence, the models had to capture non-linear relationships effectively. In addition to these, interpretability, computational efficiency, and robustness to overfitting are also crucial factors in understanding the model's reliability in predictions.

Polynomial Regression (Baseline model) : It extends linear regression and offers a robust modelling method to capture non-linear patterns. It can hence analyse the impact of weather conditions and accommodate seasonal variations on electricity prices.

Random Forest Regression (Ensemble model) : The ability of Random Forest to follow the tree structure and identify important features enhances its interpretability of the prediction. The fact that it is robust to overfitting and can handles different datatypes is an added advantage.

XGBoost Regression (High-performance model) : With high computational efficiency, ability to detect feature interactions effectively and handle overfitting, XGBoost is selected as a choice to experiment the dataset on a high performing model.

Artificial Neural Networks (ANN) (Complex model) : ANNs handles complex non-linear relationships within the data, and can effectively integrate data from multiple sources, while remaining scalable to changes in the dataset. This model was chosen to analyse how the dataset performs in a complex model. Understanding the impact of features on predictions, and how well the model can perform well on unseen data, were essential for meeting the project's objective. By analysing the results on different models with varying complexities, a thorough comparison could be made to identify the best performing model.

IV. Data Exploration

The primary step in constructing a model is acquiring high-quality and reliable data, and identifying it's influence on the model's predictive power. In the exploration stage, an in-depth analysis of the features, their characteristics, and underlying patterns need to be investigated. Prior to these, any discrepancies in the data, including the missing values must be handles.

The cleaned dataset then undergoes required reshaping and scaling, followed by effective analysis through visualizations.

This helps to understand the patterns and trends in the data, and also the correlation among the features to identify which has a stronger effect on the target feature, price.

A. Data Understanding

Three datasets in csv format were utilized in the project. The first consists of Sweden's historical weather data, which was generated by executing a provided code file, for the years 2012 to 2024. This includes daily and hourly data for meteorological parameters, sunshine time, air temperature, precipitation, and snow depth which provides essential insights on how weather changes affect the electricity demand and pricing. The second dataset comprises of day-ahead electricity price for all four electricity price zones of Sweden, spanning from 2016 to 2024. This comprised of the timestamped electricity prices in euros per megawatt hour. The third dataset consists of Sweden's electricity import and export data via Energy chart API. The market dynamics highlighting the exchanges with the neighbouring countries have an influence on the electricity price which makes this data vital for the prediction.

The analysis focused on data from 2019 to 2024, as relying on outdated data may introduce irrelevant patterns. Additionally, the data for SE3, Sweden electricity zone 3, was considered as the company Habo Plast AB is located in the respective area. All the data for the SE3 from the respective years were combined to form the final data set for the models.

The combined dataset created a robust foundation for the data analysis step where the intricate relationships between the features were explored and patterns were identified.

B. Data Cleaning

Handling missing values and unnecessary noise data from the dataset improves the performance of the model. Upon examination of the data, missing values were discovered in several columns, particularly towards the dates towards the end of the dataset. These incomplete entries were removed to maintain the integrity of the dataset. For weather parameters, sunshine was recorded in hours, while the others were reported daily. To ensure the continuity mean imputation was applied. The data cleaning helped in creating a refined dataset which then underwent feature engineering.

C. Feature Engineering

By creative appropriate inputs, feature engineering improves the model's ability to capture relationships within the data, which helps in more reliable prediction. Key steps involved in this are identifying relevant features, creating new features derived from existing data, and transforming them to optimize the model.

In response to further research and analysis, additional features were incorporated in advanced analysis to enhance the predictivity of the models. These include indicators of weekends and holidays as these affect the manufacturing dynamics and electricity consumption patterns.

Additionally, lag features were introduced, a one-hour lag and a seven-day rolling average, which allows the model to capture temporal dependencies and trends effectively. The one-hour lag accounts for immediate prior price influences, and fits the data better in the prediction. On the other hand, seven-day rolling statistics smoothens out minor fluctuations and highlights long term trends.

A correlation analysis was done to understand the effect of various feature on the target feature, price. The analysis of weather parameters indicated that sunshine time has a weak correlation with prices, while air temperature shows a slight positive correlation, suggesting higher temperatures can increase electricity prices due to cooling demands. Precipitation and snow depth demonstrated limited influence on pricing. Export data showed a negative correlation indicating increased electricity exports could lead to lower prices.

Fig. 1. Correlation Matrix of Features and Price without lag features

In the correlation analysis with the time features, the weekends and holiday indicators capture good seasonal trends and the lag feature and rolling statistics showed a significance impact on the prediction of price in such a way that they are overpowered in the model over other parameters.

After the extraction of key features like hours, days, weekends, holidays and even lag features from datetime information, the unused features like date and time were dropped to reduce unnecessary noise.

D. Data Analysis

Data analysis is the systematic approach to examine the dataset to uncover valuable insights, patterns, trends, and meaningful conclusions. We have relied on visualizations for the exploratory data analysis.

A detailed price analysis over the years was done, to examine how electricity prices have changed from 2016 to 2024. Furthermore, Sweden's historical weather data was analysed alongside energy production metrics to examine how various energy sources contribute throughout the year. The analysis highlighted the fluctuations in prices, particularly during 2022

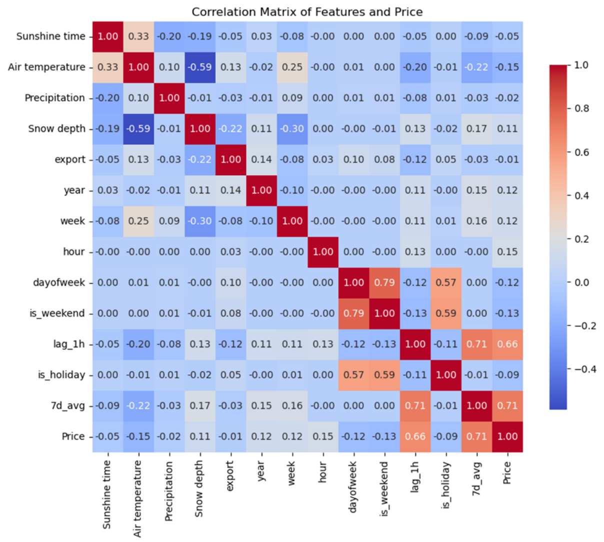

Fig. 2. Correlation Matrix of Features and Price with Lag features

and 2023. This was driven by geopolitical events such as the Ukraine war, which disrupted supply and heightened the prices. Seasonal patterns were also evident. For instance, higher prices were observed in winter months due to increased heating demand. Additionally, spikes were observed from July to September, which corresponded with increased energy exports to Germany, particularly as their nuclear power plants were temporarily shut down.

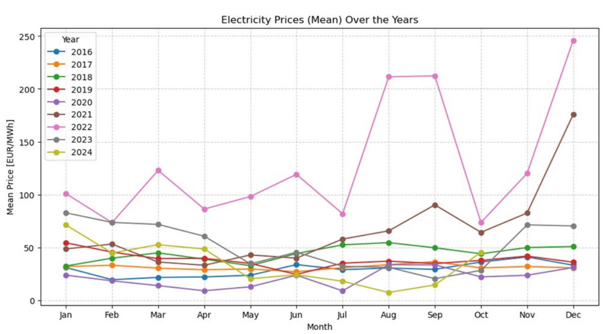

Fig. 3. Electricity Prices(Mean) Over the Years

The impact of energy exports on pricing was also assessed. During peak demand periods in neighbouring regions, domestic supply can be reduced and, consequently, elevate prices.

The comprehensive analysis of electricity prices, weather data, and energy production patterns revealed key factors influencing the electricity price market behaviour. This integrated approach, combined with feature engineering, established a foundation for developing accurate predictive models.

V. Approach Description

This section details different approaches used throughout the project to arrive at the final implementation, and conclusions.

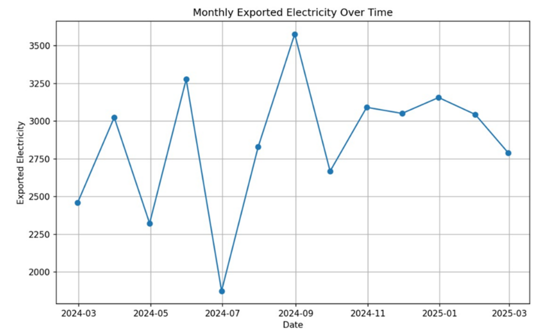

Fig. 4. Monthly Exported Electricity Over Time

The subchapters cover many procedures that are employed in the final implementations.

A. Tools

Jupyter Notebook, Visual Studio Code, and Excel were integral tools throughout the project. Jupyter Notebook enabled the creation of comprehensive documents that included code, visuals, and explanations. Visual Studio Code provided a flexible environment for coding tasks. Excel was employed for organizing and storing data. The entire project was developed using the Python programming language.

B. Libraries

To enhance the data analysis and modeling processes, various Python libraries were utilized. These libraries were essential in simplifying the development of a robust program capable of managing large datasets, making predictions, and deriving meaningful insights.

Pandas: A Python library that facilitated working with structured data like DataFrame and Series structures, enabling efficient data cleaning, transformation, and exploration. It is a key tool for preparing data before modeling.

NumPy: It is used mainly for numerical computing and for handling large arrays and matrices.

Scikit Learn / sklearn: It offers various algorithms for tasks like classification, regression, and clustering. It also includes tools for data preprocessing, model evaluation, and tuning parameters.

Tensorflow: An open-source machine learning framework used for building deep learning models, such as neural networks and it also supports large-scale computations.

Holidays: Provides holiday data for different countries, which is useful in time-series forecasting to include holiday effects in models.

Matplotlib: A data visualization library widely used for creating plots, particularly for data exploration and presentation. It offers a wide range of functionalities for creating various types of plots, including line plots, scatter plots, bar plots, and histograms.

Seaborn: A Python data visualization library built on matplotlib. It simplifies creating attractive statistical graphics with minimal code, making it popular for data exploration and presentation.

C. Standardization

Standardization is a preprocessing technique used to transform the features to have a mean of zero and a standard deviation of one. In the project, StandardScaler was used to apply this transformation, ensuring all features contribute equally to the model's predictions. The features in the dataset belonged to different scales, and StandardScaler prevents features with larger scales from dominating the model's learning process, thereby improving learning efficiency and convergence.

D. Train-Test Split

The dataset was split into training and testing sets using the Scikit-learn library. This technique is important for evaluating the model's ability to generalize on unseen data. The dataset was split in such a way that 80% of the data was used for training, allowing the model to learn patterns, while the remaining 20% served as the testing set, which was unseen by the model during training. This helps assess the model's performance on real-world, unseen data.

E. Hyperparameter Tuning

Polynomial Regression, Random Forest, XGBoost, and Artificial Neural Networks (ANN) were the machine learning techniques used for modeling. Polynomial Regression was used as a baseline model due to its simplicity and interpretability, providing a starting point for comparison with more complex models. Polynomial Regression model can handle both linear and non-linear relationships, but complex models like XGBoost and Artificial Neural Networks offer flexibility and are best suited for modeling complex, high-dimensional interactions within the data. Additionally, the Random Forest model, an ensemble learning method, was used due to its ability to handle non-linear relationships and its robustness against overfitting by averaging multiple decision trees. To enhance model performance, hyperparameter tuning was performed using GridSearchCV, which exhaustively searches for the best combination of hyperparameters for each model. This process helped to optimize the settings of all algorithms, improving their predictive accuracy and making them more effective for the price prediction task.

GridSearchCV is a hyperparameter tuning technique that helps to find best combination of parameter for a given model by conducting an exhaustive search over a specified hyperparameter grid. In this project, GridSearchCV was used with different cross-validation techniques to ensure the model's robustness and to prevent overfitting.

Fig. 5 shows unique hyperparameters used for each model.

F. Models

Several models were developed for this project to forecast the hourly prices of electricity for the upcoming week.

Polynomial Regression

This baseline model can capture both linear and non-linear relationships in electricity price trends. For the model to learn more complex patterns increasing the degree can help. However, in that case, the model might cause overfitting, which would reduce its ability to capture abrupt changes in electricity prices. Despite this limitation, this regression technique provides a useful reference point for comparing the performance of more advanced models.
Best parameter after hyperparameter tuning : poly__degree = 2

Model	Parameter	Values
Polynomial Regression	poly__degree	[2, 3, 4, 5]
Random Forest	n_estimators	[100,200]
	max_depth	[3, 6, 9, None]
	min_samples_split	[2, 5, 10]
	min_samples_leaf	[1, 2, 4]
	max_features	["sqrt", "auto"]
XGBoost	n_estimators	[100, 200]
	max_depth	[3, 6, 9]
	learning_rate	[0.01, 0.1, 0.2]
	subsample	[0.8, 1.0]
	colsample_bytree	[0.8, 1.0]
Artificial Neural Network	optimizer	adam
	activation	relu

Fig. 5. Hyperparameters used in all models

Random Forest

Random forest works on the divide and conquer approach and is based on the random subspace method. Here, multiple decision trees are built, and each decision tree is trained by selecting any random sample of attributes from the predictor attributes. Each tree matures up to a maximum extent based on the attributes or parameters present and makes a numerical prediction. The final decision tree formed for the prediction is mainly based on weighted averages of all the predictions made. This technique helps to reduce the chance of overfitting.
Best parameter after hyperparameter tuning: {'max_depth': None, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 200}

Extreme Gradient boosting

It is a machine learning technique that builds a predictive model by successively combining several weak learners. Iteratively, it enhances the model's predictions. This approach is effective for predicting electricity prices because it can capture complex interactions and prevent overfitting through regularization techniques. Hyperparameters like learning rate, maximum tree depth, and column sampling were changed using GridSearchCV to improve prediction performance and ensure the model could adapt to fluctuating electricity prices.
Best parameter after hyperparameter tuning: {'colsample_bytree': 1.0, 'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 100, 'subsample': 1.0}

Artificial Neural Networks (ANN)

ANN is a complex model composed of multiple layers of interconnected neurons that learn complex patterns through weight adjustments during training. It can process large amounts of historical data, detect hidden trends, and adapt to nonlinear dependencies. So, this model technique is capable of electricity price prediction. The activation function used in the model was ReLU was used to introduce non-linearity, while the Adam optimizer was chosen for efficient gradient-based learning. It was fine-tuned to enhance forecasting accuracy while avoiding overfitting.

VI. Experiment Design

This section covers the structure of the experiment, including the training and testing procedures and the metrics used for evaluation.

A. Lag Features and rolling statistics

Lag features are variables derived from past values in the dataset that are shifted by a certain number of time steps as required. Rolling statistics are the moving averages used to smooth out short-term fluctuations and highlight longer-term trends. They are useful in handling noisy data and identifying patterns over time. So, these features help to understand how previous observations at earlier times can help predict future outcomes. In this way, the model can learn these relationships and make future predictions based on old data. In time-series forecasting, lag features are necessary for the model to learn trends in the data and to increase its predictive power.

Below lag features are added to the dataset while doing time-series cross-validation:
lag_1h : Stores the electricity price from the previous time step (the previous hour).
7d_avg : Stores the 7-day rolling average of the electricity price.

B. Training and Testing Procedure

The train test split is used to evaluate the model's ability to generalize unseen data. After training on the training set, the model is evaluated on the testing set. This helps ensure that the model can make predictions on data that hasn't been seen before. Different shuffling procedures are used for various cross-validation approaches to properly divide the data.

Time-Series Cross-Validation with Lag Features

Time-series cross-validation is a tailored approach for validating models on time-series data. Unlike traditional cross-validation methods that randomly split the data into training and test sets, time-series cross-validation preserves the chronological order of the data to prevent data leakage. This is crucial in forecasting tasks, where future data points should never be used to predict past or current ones. These models use lag features to learn trends and to make accurate predictions.

TimeSeriesSplit: In time-series forecasting, it is critical to maintain the chronological order of the data to prevent data leakage. Unlike other traditional cross-validation methods that involve a random shuffling of the dataset, TimeSeriesSplit ensures that training is conducted on past data, with testing performed on future data. For this experiment, the split number was set to 5 which will split the dataset into 5 folds. So, each fold uses the data up to a certain date for training and tests it on future dates. In this way, data leakage risk is eliminated.

Each model's performance is calculated with this approach and analysed.

	Model	MAE	MSE	RMSE	R2 Score
0	Polynomial	16.360872	397.684581	19.942031	0.748548
1	ANN	16.566198	924.523735	25.680394	0.780174
2	Random Forest	9.559670	209.550642	14.475864	0.867503
3	XGBoost	6.280150	141.549685	11.897465	0.910499

Fig. 6. Model's performance comparison with Time-series Cross-Validation

K-Fold Cross-Validation without Lag features

K-Fold cross-validation technique is used to assess the performance of the model. The dataset is split into 10 equal parts and then the first nine sets are used for training and the last remaining set is used for testing. This process is repeated 10 times, and this ensures that each data point is used for both data testing and training. After all iterations, performance metrics for each fold are calculated and averaged.

The model was trained and tested without lag features, so the model is forced to rely on more direct features, such as weather data and import-export data. This experiment is carried out to compare the effect of lag features on different cross-validation techniques.

The performance of each model is evaluated using this approach and analyzed in detail.

	Model	MAE	MSE	RMSE	R2 Score
0	Polynomial	36.244464	3834.699015	61.924947	0.245056
1	ANN	14.426013	598.397016	24.430360	0.886924
2	Random Forest	8.545941	319.528055	17.875348	0.937094
3	XGBoost	9.141240	293.996739	17.146333	0.942120

Fig. 7. Model's performance comparison with K-fold cross validation

When comparing the results between time-series cross-validation and K-Fold cross-validation, it is evident that although the R^2 score is higher in the case of 10-Fold cross-validation, the Mean Absolute Error (MAE) is lower with the time-series cross-validation approach. This suggests that, despite the higher R^2 in K-Fold, the time-series cross-validation model provides more accurate predictions with fewer large errors. Therefore, the time-series cross-validation model is considered more reliable for this task, as it better handles the temporal dependencies in the data and avoids potential data leakage.

C. Feature Importance Analysis

We have two different feature analysis insights for the XGBoost Model – without lag features, and with lag features. We have opted to perform analysis only on the XGBoost model as this is the most promising model achieved from the training and testing phase. The analysis of both have been presented below.

From the figure 8, the top influential features in the analysis

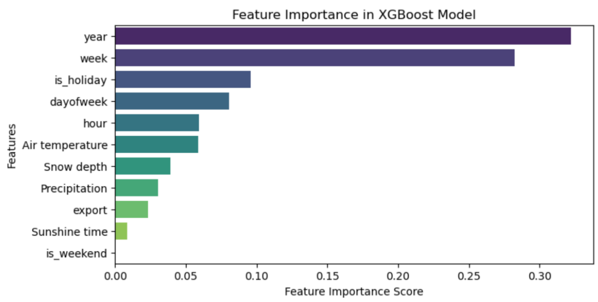

Fig. 8. Feature importance without Lag Features

proved to be the year of the input data, with an importance score of 0.32, seconded by the week of the input data, with an importance score of 0.28. This is followed by moderately important features like holidays and day of the week which have importance scores of 0.1 each. Other features that barely make this category include the Hour of the input data, and the ambient Air Temperature with importance values of 0.6 each. Lastly, we have the low important features like Snow Depth, Percipitation, Sunshine time and is_weekend, all having importance values less than 0.05.

From the figure 9, the top influential features in the analysis

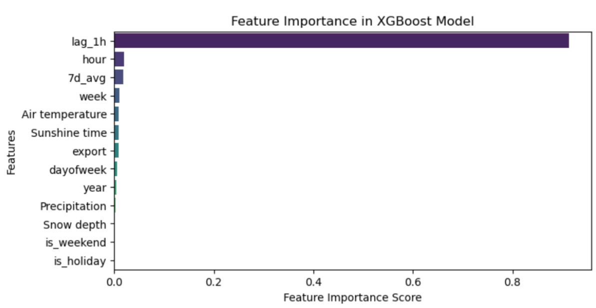

Fig. 9. Feature importance with lag features

proved to be the lag_1h, with an importance score of 0.93, representing the target value from 1 hour prior. This feature completely dominates the feature set. This is followed by features with secondary influences like hour, 7d_avg, week and Air temperature which all have marginal importance scores. Lastly, we have the features with minimal/negligible importance like Sunshine Time, export, dayofweek, year, Percipitation, Snow depth, is_weekend and is_holiday. Their low importance implies that the model finds very little predictive signals from these variables.

D. Evaluation Metrics

The performance of each model was evaluated using the following metrics, which are standard for regression tasks:

- **Mean Absolute Error (MAE):**

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{1}$$

  - Where y_i is the actual electricity export, x_i is the predicted export, and n is the number of data points.
  - MAE measures the average magnitude of errors in the predictions.

- **Mean Squared Error (MSE):**

$$\text{MSE} = \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N} \tag{2}$$

  - Where y_i is the actual electricity export, x_i is the predicted export, and n is the number of data points.
  - MSE penalizes larger errors more heavily due to the squaring of the differences. A lower MSE indicates better model performance.

- **Root Mean Squared Error (RMSE):**

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N}} \tag{3}$$

  - RMSE gives a higher weight to larger errors, providing a more sensitive measure of prediction accuracy.

- **R-squared (R²):**

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} \tag{4}$$

  - Where $\bar{y}$ is the mean of the actual electricity export values.
  - R^2 represents the proportion of the variance in the dependent variable that is predictable from the independent variables. A higher R^2 indicates a better fit.

E. Model Performance Comparison using metrics

The performance of the four models will be compared based on the evaluation metrics calculated on the test set. This will allow for the identification of the most effective model for predicting Sweden's hourly electricity prices within the SE3 distribution zone. The model that produces the lowest MAE and RMSE, and the highest R² will be considered the best model.

VII. Result Analysis

A. Results

After experimenting with different approaches, a comparative analysis on the performance of all four models was conducted.

Model	MAE ↓ (Lower is better)	RMSE ↓ (Lower is better)	R² ↑ (Higher is better)
Polynomial Regression	Moderate	High	Low
Random Forest	Moderate	Moderate	Moderate
XGBoost	Lowest	Lowest	Highest
ANN	Comparable to XGBoost	Higher than XGBoost	Slightly lower than XGBoost

Fig. 10. Model comparison based on Evaluation Metrics

The XGBoost Model has the best prediction among all models with the lowest Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and the highest R^2 Score. Random Forest performed well but not as precise as XGBoost. Polynomial Regression, implemented as baseline model, did not show a commendable performance when it came to complex electricity price fluctuations. ANN is powerful but needs more training time. Even then, it's performance didn't match the predictive performance of XGBoost.

The following Graph shows how each model performs:

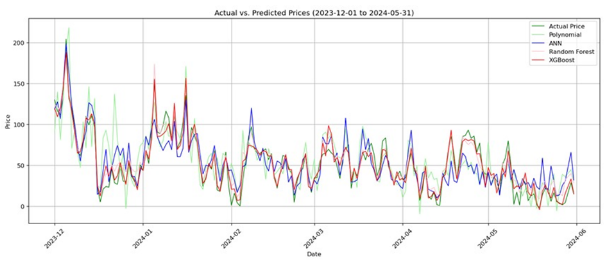

Fig. 11. Models performance

The graph represents the actual price compared to the predicted price from December 2023 to May 2024, using the four models: Polynomial Regression, Random Forest, XGBoost and Artificial Neural Networks (ANN). From the graph we can see that all models closely follow the trends of actual price, but the level of accuracy varies::

- The Polynomial Model (green line) shows a fluctuation of trend in the higher price changes, and finds it difficult to capture sudden changes.
- The ANN model (blue line) shows a smooth graph, and aligns well with actual prices, but shows slight lag with the sudden highs and lows.
- The Random Forest Model (yellow line) follows the actual prices reasonably well, but its predictions appear less consistent during the latter part of the period.
- The XGBoost Model (red line) shows a strong alignment with actual prices, especially during periods of steady trends, however it occasionally underestimates during sharp spikes.

B. Analysis

From the observed results, we can infer the following characteristics:

1) Lower Prediction Errors XGBoost consistently makes fewer mistakes when predicting the target variable. This means its estimates are much closer to the actual values compared to other models like polynomial regression or artificial neural networks (ANN).
2) Better Generalization A good model should not just memorize patterns in the training data but also perform well on unseen data. XGBoost is showing the least amount of error spread, meaning it generalizes better and avoids overfitting.
3) More Stable Performance While ANN and Random Forest also perform well, they may sometimes be more sensitive to changes in data or require more tuning. XGBoost, on the other hand, uses boosting techniques that help it refine predictions gradually, making it more robust.
4) Handles Complexity Well Compared to Polynomial Regression, which can struggle with complex relationships in data, XGBoost is designed to capture intricate patterns without adding too much noise.
5) Efficiency in Learning XGBoost is optimized for speed and accuracy. It builds trees sequentially, learning from mistakes in previous steps, which helps it improve results over time while still maintaining efficiency.

When selecting a model for forecasting, the choice depends on balancing accuracy, complexity handling, and computational efficiency. Polynomial Regression models are suitable for simpler, linear relationships but struggle with complexity and generalization. Random Forest provides a robust and stable alternative, particularly for handling varied datasets, though it requires significant computational resources. XGBoost emerges as a standout performer, offering a balance between lower prediction errors, generalization, and stability, excelling in complex scenarios. ANN, while powerful for highly intricate patterns, demands careful tuning and computational power, making it ideal for specialized cases.

Ultimately, XGBoost is recommended for most forecasting tasks due to its versatility, adaptability, and strong performance across diverse datasets. However, each model's selection should align with the specific requirements and constraints of the project.

VIII. Conclusion

In this project, we evaluated four machine learning models—Polynomial Regression, Random Forest, XGBoost, and Artificial Neural Networks (ANN)—for time series regression on a dataset with approximately 50,500 entries and a target variable ranging from 0 to 130.

Initially, all models were trained and evaluated using standard features, and performance was measured using MAE, MSE, RMSE, and R^2 score. Among the initial results, XGBoost demonstrated superior performance with the lowest MAE (6.28), MSE (141.55), RMSE (11.89), and the highest R^2 score (0.91), indicating strong predictive capability and generalization.

To enhance model accuracy, lag features were introduced to capture temporal dependencies in the data. This led to significant performance improvements across all models:

- **Polynomial Regression**: Improved R^2 score from 0.25 to 0.75 and reduced MAE from 36.24 to 16.36.
- **ANN**: Enhanced R^2 from 0.88 to 0.78 (minor drop), suggesting sensitivity to lag feature engineering.
- **Random Forest**: R^2 decreased from 0.93 to 0.87 with consistently low error metrics.
- **XGBoost**: Maintained top performance with an R^2 of 0.94 post-lag features, reaffirming its robustness and accuracy.

Hyperparameter tuning was conducted via grid search, optimizing each model for performance (parameters shown above). XGBoost remained the most consistent and accurate across evaluations, making it the preferred model for this time series prediction task. After this finalisation, we added a generic Mean Error (ME) metric to the best XGBoost model, inspired by a suggestion with the HaboPlast representative so as to understand to what extent the sum of the errors area pointing. And as per our evaluations, we have reached a mean error of -0.7792, indicating that the model was under-predicting the prices more often than over-predicting the prices for the test set.

A. Key Takeaways

- Lag features greatly improved model performance, especially for Polynomial Regression.
- XGBoost outperformed all other models, offering the best balance between error minimization and variance explanation.
- Model selection should consider both predictive power and computational cost, with XGBoost providing optimal trade-offs for time series regression in this context.

B. Recommendations

Within the SE3 power zone of Sweden's electricity grid, Nuclear power contributes to 45-50% of the energy production. But since Nuclear Power isn't affected by weather, it hasn't been taken into account. For the future, we recommend adding features that take Nuclear power into consideration and model the data accordingly.

Additionally, we suggest a Language model to monitor news channels for any possible events or anomalies that may require abnormal export values or anomalies in weather forecasts. This would allow for foreseeing peaks or lows related to societal events or natural disasters.

IX. Acknowledgment

We would like to express our sincere gratitude to Dr. Florian Westphal and Dr. Alexandros Tzanetos for their valuable guidance and support throughout to achieve the project goal. Additionally, we would like to acknowledge Simon Sleman from Science Park for providing us with the opportunity to collaborate with HaboPlast company.

References

[1] K. Likitha Chowdary, Ch. Nanda Krishana, K. Srikanshka Manaswini, and B. Jithendra, *Electricity Price Prediction using Machine Learning*, Available: https://ieeexplore.ieee.org/document/10073777, [Accessed: Feb. 10, 2025].

[2] D. H. Mazengia, *Forecasting Spot Electricity Market Prices Using Time Series Models*, Chalmers University of Technology, Gothenburg, Sweden, 2008. Available: https://webfiles.portal.chalmers.se/et/MSc/DawitHailuMazengia.pdf, [Accessed: Feb. 11, 2025].

[3] R. Sandberg, "Wind and Electricity Prices in Sweden – a Statistical Analysis," Energiforsk, 2024. Available: https://energiforsk.se/media/33100/2024-975-wind-and-electricity-prices-in-sweden-a-statistical-analysis.pdf. [Accessed: Feb. 12, 2025].