AI POWERED SMART GRID OPTMISATION

JACK FRASER

ABSTRACT

Machine learning and artificial intelligence play a crucial role in data-driven industries. This project demonstrates the development of a machine learning model designed for power grid optimization. While this model specifically focuses on electricity demand forecasting and grid management in the German market, it can be adapted for various applications and other regional energy systems with appropriate data preprocessing.

The dataset, sourced from Open Power System Data, was provided in a standard CSV format to ensure ease of data handling and model implementation. Future iterations of this project will incorporate real-time energy data via APIs and SQL databases to enhance accuracy and scalability.

INTRODUCTION

In any real-world system, data collection and analysis are critical for cost assessment, resource allocation, and continuous system improvement. Without proper data-driven decision-making, industries would struggle to optimize performance, reduce costs, and improve efficiency. This principle applies to power grid management, where forecasting electricity demand allows for more precise energy generation and distribution. This project leverages historical energy output data from a given region and applies machine learning-based predictive modelling to estimate future electricity demand with a high degree of accuracy.

Forecasting energy demand is crucial for ensuring that sufficient power is generated without excess waste. In the context of power grid optimization, this means:

1. Avoiding energy shortages – Prevents blackouts and ensures grid stability.
2. Reducing overproduction – Minimizes unnecessary energy generation, reducing operational costs.
3. Lowering consumer costs – Efficient energy distribution translates into financial savings for both providers and consumers.

The machine learning model developed in this project is based on high-frequency electricity demand data recorded in Germany. The dataset provides accurate energy usage values at 15-minute intervals, enabling a granular analysis of peak and off-peak periods. A finer time resolution improves predictive accuracy by capturing short-term fluctuations in energy consumption patterns.

Modern power systems must integrate traditional energy sources with renewable resources. This project specifically incorporates wind and solar energy contributions, recognizing their intermittent nature.

- Solar power generation is available only during daylight hours, varying based on weather conditions.
- Wind energy generation depends on wind speed and atmospheric conditions, which fluctuate unpredictably.

Due to these limitations, the project also considers battery storage systems, which store excess renewable energy for use during low-production periods. By integrating battery storage, the model reduces reliance on conventional energy sources while ensuring continuous electricity supply.

The forecasting model implemented in this project utilizes a Random Forest Regressor, a non-linear ensemble learning algorithm that is well-suited for time-series forecasting in dynamic systems. The reasons for selecting Random Forest include:

- Ability to handle complex relationships between electricity demand and external factors.
- Robustness to non-linear fluctuations in energy consumption patterns.
- Strong performance in predicting power usage across a diverse grid of users with varying consumption behaviours.

The model was trained using historical energy data with the following key input features:

- Historical electricity demand
- Time-based features (hour, day, month, weekday/weekend distinctions)
- Lag features (previous load values)
- Rolling averages (6-hour and 24-hour trends)
- Renewable energy production (solar & wind generation levels)

The dataset for this project was sourced from Open Power System Data (OPSD), which provides comprehensive real-world electricity usage and renewable generation records. While this project utilizes CSV-based historical data, future iterations will incorporate real-time data streams via APIs and SQL databases to improve forecasting accuracy and model scalability. A real-time data pipeline would be the preferred method for energy companies looking to optimize their power generation systems dynamically.

To enhance the predictive performance of the machine learning model, feature engineering was a critical step. This included:

- Time-based features – Capturing daily, weekly, and seasonal demand fluctuations.
- Lagged features – Providing historical demand references for short-term forecasting.
- Rolling averages – Smoothing short-term fluctuations to capture broader consumption trends.

By incorporating these features, the model learns demand patterns more effectively, improving forecasting accuracy over both short and long-term time horizons. Once the electricity demand forecast is generated, the final goal is to optimize power distribution to minimize costs while ensuring grid reliability. This is achieved using linear programming (LP), specifically implemented through PuLP, a Python optimization library. The LP model determines:

- Optimal allocation of conventional power generation
- Maximization of renewable energy utilization
- Efficient use of battery storage to balance power distribution

By integrating forecasted energy demand with cost-efficient energy allocation, this model enables smart grid optimization, reducing operational costs and improving overall energy efficiency.

This project presents a data-driven approach to power grid optimization by integrating machine learning-based forecasting and mathematical optimization techniques. By leveraging historical data, renewable energy integration, and predictive modelling, the system provides an efficient framework for reducing costs and enhancing grid stability. Future improvements, such as real-time data integration and automated decision-making, will further refine the system's capabilities, making it a viable solution for modern energy management.

METHODOLOGY

Data Collection & Preprocessing

The dataset used in this study was sourced from Open Power System Data (OPSD), a publicly available repository providing historical electricity demand, renewable energy generation, and grid operation data. The dataset contains 15-minute interval records of electricity demand and generation in Germany, ensuring a high-resolution dataset for modelling purposes. The data was acquired in CSV format, which allows for easy handling and preprocessing within Python-based analytical environments.

Before applying machine learning techniques, it was necessary to clean and preprocess the data to ensure consistency and remove any inconsistencies. Missing values were addressed using interpolation techniques where necessary, while duplicate records and extreme outliers were filtered to maintain data integrity. Additionally, all timestamps were converted to datetime format, ensuring compatibility with time-series forecasting models. The dataset was then sorted chronologically to preserve the natural order of energy consumption trends.

Feature Engineering

To improve the predictive performance of the model, several feature engineering techniques were applied to extract meaningful patterns from the dataset. Key features included:

- Time-Based Features: Hour, day, month, and weekday/weekend indicators were created to capture daily and seasonal variations in energy demand.
- Lag Features: Previous electricity consumption values were included as input features, allowing the model to learn from past trends.
- Rolling Averages: Moving average features were used to smooth short-term fluctuations and highlight broader consumption trends.
- Renewable Energy Generation: Solar and wind energy data were incorporated to assess their contribution to overall electricity demand fulfillment.

These engineered features enabled the model to capture both short-term variations and long-term seasonal trends, improving forecasting accuracy.

Machine Learning Model Selection

The forecasting component of this project utilized a Random Forest Regressor, an ensemble learning algorithm known for its robustness in handling non-linear relationships and time-series forecasting challenges. Several alternative models, including AutoRegressive Integrated Moving Average (ARIMA), XGBoost, and Long Short-Term Memory (LSTM) networks, were considered; however, Random Forest was selected due to its ability to:

- Handle complex dependencies between time-series variables without requiring explicit assumptions about data stationarity.
- Perform well with limited hyperparameter tuning compared to deep learning models such as LSTMs.
- Offer interpretability through feature importance scores, providing insight into which variables most influence energy demand.

Model Evaluation

To assess the effectiveness of the forecasting model, the following evaluation metrics were used:

- Root Mean Squared Error (RMSE): Measures the average magnitude of forecast errors, with lower values indicating better accuracy.
- Mean Absolute Error (MAE): Evaluates the absolute differences between predicted and actual values, offering insight into model precision.

The final model achieved an RMSE of 353.31 MW and an MAE of 260.60 MW, demonstrating its ability to predict electricity demand with a high degree of reliability.

Optimisation of Power Distribution

After generating demand forecasts, the next phase of the project focused on optimizing energy distribution to minimize costs while ensuring grid stability. This was achieved using Linear Programming (LP), a mathematical optimization technique that determines the most cost-effective way to allocate power from different sources.

Using PuLP, a Python-based linear optimization library, the optimization model was solved iteratively for each 15-minute interval, determining the optimal power allocation strategy while keeping costs at a minimum.

RESULTS & FINDINGS

1. Machine Learning Model Performance

The forecasting model was trained using historical electricity demand and energy generation data. The Random Forest Regressor was used due to its ability to handle non-linear relationships in time-series data. The model was evaluated using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to assess prediction accuracy.

- MAE: 260.60MW
- RMSE: 353.31MW

These values indicate that the model provides accurate demand forecasts within an acceptable margin of error, ensuring that energy allocation decisions are based on reliable predictions.

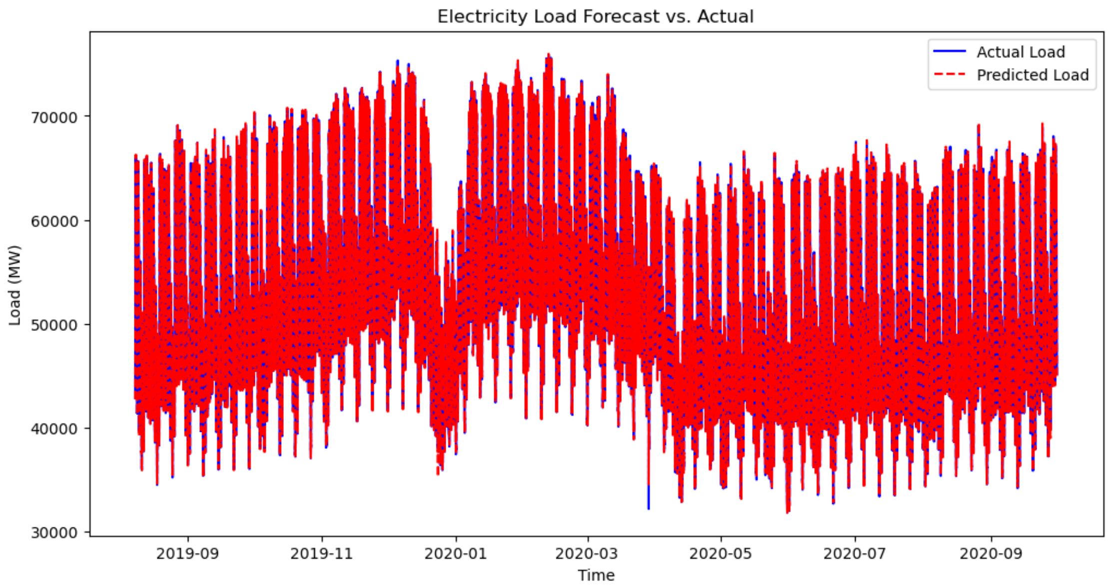

Figure 1: Model performance accuracy

Figure 1. demonstrates the model's effectiveness in capturing fluctuations in electricity demand over time. The Error Distribution plot further illustrates the variance between predicted and actual values, helping identify any systematic biases in the model's predictions.

2. Training Data Insights

The dataset used for training the forecasting model consisted of 201,605 observations, covering a time range from December 2014 to October 2020 with 15-minute time intervals. Feature engineering included:

- Time-based features (hour, day of the week, month)
- Lag variables (previous demand values)
- Rolling averages (6-hour and 24-hour trends)
- Renewable energy generation (solar and wind power)

By incorporating these features, the model was able to learn daily and seasonal demand patterns, improving the robustness of forecasts for both peak and off-peak hours.

3. Grid Optimization Results

After forecasting electricity demand, the optimization model allocated power from conventional energy sources, battery storage, and renewables to minimize operational costs while ensuring grid stability. The Linear Programming (PuLP) optimization framework was used to determine the optimal energy distribution strategy.

The results of the grid optimisation in the date range of 2020-03-29 02:00:00+01:00 to 2020-04-05 01:45:00+01:00 were as follows.

Metric	Value
Total Forecasted Demand (MW)	24,099,579.85
Total Conventional Power Used (MW)	19,796,779.78
Total Renewable Power Used (MW)	4,300,800.00
Total Battery Discharge (MW)	2,000.00
Peak Demand (MW)	36,059.36
Average Demand (MW)	35,862.47
Peak Conventional Power (MW)	29,659.36
Average Conventional Power (MW)	29,459.49
Peak Renewable Power (MW)	6,400.00
Average Renewable Power (MW)	6,400.00
Peak Battery Discharge (MW)	2,000.00
Average Battery Discharge (MW)	2.98
Total CO_2 Emissions (tons)	7,918,711.91
Average CO_2 Emissions (tons)	11,783.80
Renewable Contribution (%)	17.85
Battery Contribution (%)	0.01
Conventional Contribution (%)	82.15

Table 1: Optimisation results

To note with this result, the time frame is 1 day after the initial data set ends, this is due to using the random forest regressor, it is less accurate on long time intervals but takes less computing power, which is the constraint I ran into. In order to predict long term you would utilise a more robust toolkit such as PyTorch. I will update the codebase utilising PyTorch in the near future. As for the results being relatively high (high MW), this is consistent with the initial dataset provided, I will generate several other optimisation reports based on different datasets to show consistency.

4. Key Findings & Future Improvements

- The forecasting model provided highly accurate predictions, reducing uncertainty in grid planning.
- The optimization model effectively allocated power resources, balancing cost efficiency and energy availability.
- Potential improvements include integrating real-time energy market pricing, incorporating weather data for better forecasting, and expanding the model for multi-region grid management.

By leveraging machine learning for forecasting and optimization algorithms for resource allocation, this project demonstrates how AI-driven approaches can enhance energy efficiency and grid stability in modern power systems.

MODIFICATION

The AI-Powered Smart Grid Optimization model is designed to be highly adaptable to different datasets, allowing for seamless integration with new energy demand and generation data. The core methodology remains unchanged, but modifications are required to align with the specific structure and attributes of the new dataset. The key aspects to consider when transitioning to a new dataset include data cleaning, feature engineering, forecasting adjustments, and optimization constraints.

The first step in adapting the model is ensuring that the raw dataset is properly formatted. This involves updating the data cleaning script (data_cleaning.py) to recognize new column names, handle missing values, and correctly parse timestamps. If the new dataset follows a similar structure to the original, minimal changes are required beyond replacing the dataset file. However, if additional features or different time resolutions are introduced, further preprocessing steps may be necessary to maintain consistency in the forecasting pipeline.

Once the dataset has been pre-processed, feature engineering modifications may be required to align with the available data fields. The model relies on time-based features such as hour of the day, day of the week, and rolling averages of past demand to improve prediction accuracy. If the dataset includes additional factors such as hydroelectric power, nuclear energy, or regional demand variations, these variables can be incorporated into the feature set by modifying feature_engineering.py.

The forecasting model, implemented through a Random Forest Regressor, must also be adjusted to ensure that the correct target variable is used for training. If the new dataset represents demand differently (e.g., total grid demand instead of regional demand), forecasting_model.py must be updated to reflect this change. Additionally, if new datasets introduce external influencing factors such as weather conditions or energy market prices, these can be integrated into the model to improve forecasting accuracy.

Finally, the grid optimization algorithm can be tailored to accommodate new power generation sources. The existing optimization approach distributes energy across conventional sources, battery storage, and renewables, but new datasets may introduce hydroelectric power, nuclear energy, or variable energy costs that require adjustments in the constraints and objective function of grid_optimization.py. By modifying these constraints, the model can ensure an efficient allocation of power resources while maintaining reliability and minimizing costs.

This structured approach allows the model to be effectively applied to different power grids, regions, or market conditions, making it a flexible tool for energy forecasting and optimization in diverse settings.

CONCLUSION

This project was undertaken to deepen my understanding of machine learning (ML) techniques and their applications in real-world energy management. Through this process, I developed a forecasting system that accurately predicts electricity demand based on historical consumption patterns and renewable energy generation. Additionally, I implemented an optimization model that efficiently distributes power across conventional sources, battery storage, and renewables, minimizing costs while improving grid stability.

The project provided valuable insights into time-series forecasting, feature engineering, and ML model evaluation, reinforcing my ability to apply data-driven solutions to complex energy challenges. The AI-powered smart grid optimization model successfully achieved its primary objective of accurately forecasting electricity demand (RMSE: 353.31 MW, MAE: 260.60 MW) and optimizing power distribution to enhance efficiency and sustainability. By leveraging machine learning for demand prediction and linear programming for energy allocation, the model demonstrated effective resource management, maximizing renewable energy use while reducing reliance on costly conventional power sources.

Future iterations will focus on integrating real-time data streams, weather predictions, and dynamic market pricing, further refining the model's accuracy and practical applications. The approach outlined in this project presents a scalable and adaptable framework for optimizing modern power grids, paving the way for more intelligent, cost-effective, and sustainable energy solutions.

APPENDIX

- Optimization with PuLP: https://coin-or.github.io/pulp/
- Data: https://data.open-power-system-data.org/time_series/

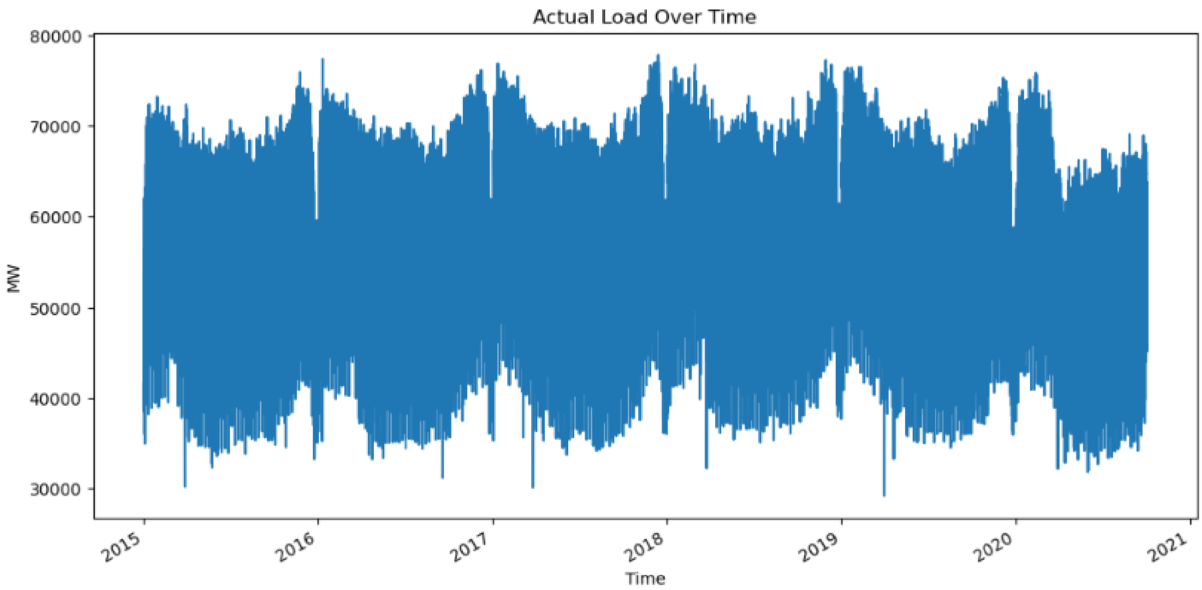

Figure 2: Actual load over time

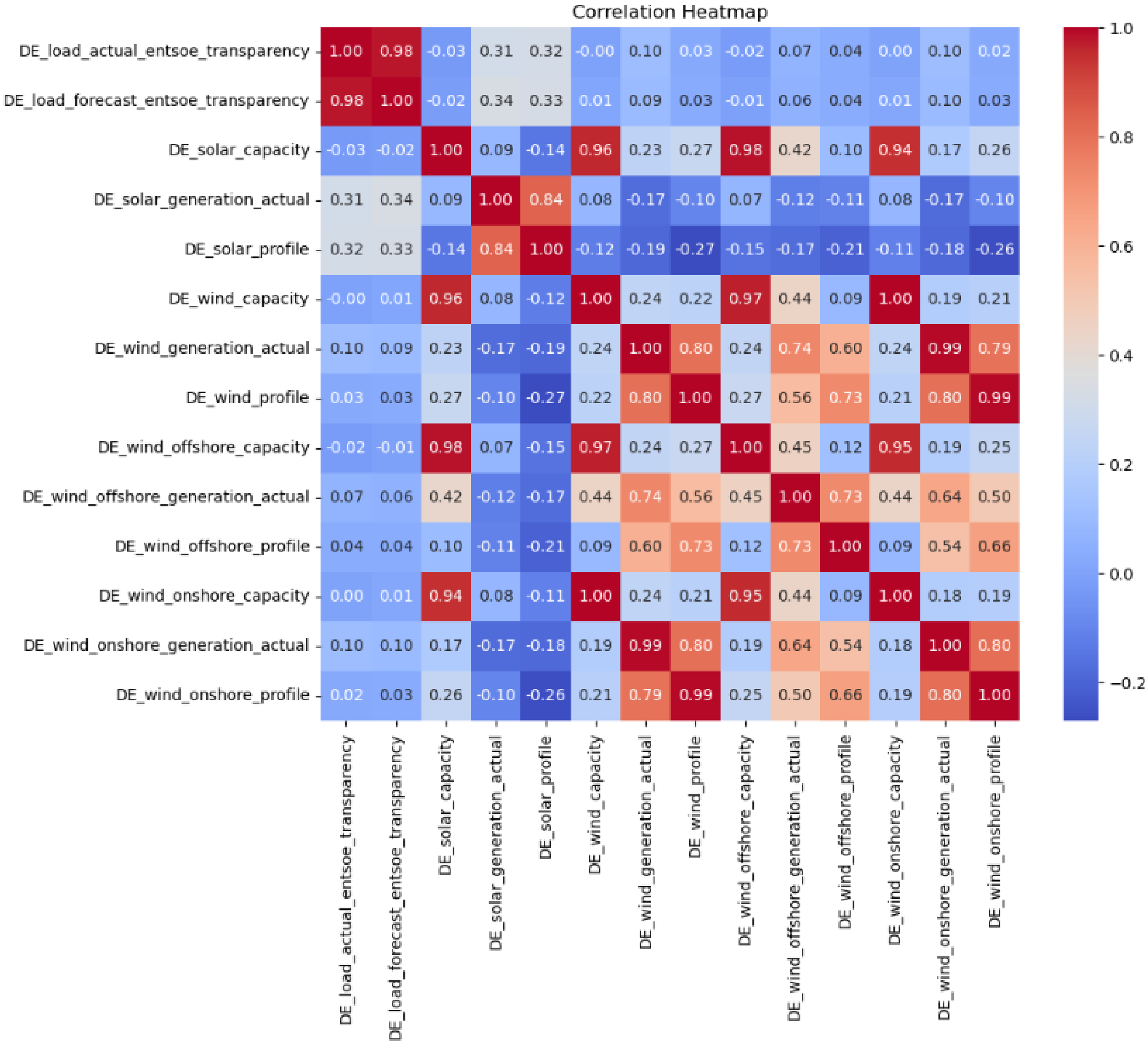

Figure 3: Correlation heatmap