

University of Stavanger

Faculty of Science and Technology
Department of Electrical Engineering and Computer Science

Project 4:

Forecasting Daily Household Energy Consumption



Group 15

Pierre Gary Benjamin Simonsen Oddbjørn Kleven
Sondre Lyngstad Halvard Erlandsen Sondre Espeland

University of Stavanger

February 22, 2026

Table of Contents

List of Figures	ii
1 Introduction	1
1.1 Research Problem	1
1.2 Dataset Description	2
1.2.1 Dataset Columns	3
1.3 Predictions for the Household Power Consumption Dataset	3
1.4 Models in the Project	5
1.4.1 Base Models	5
1.4.2 Other Models	5
2 Preprocessing	6
2.1 Importing Libraries	6
2.2 Importing Data	6
2.3 Preprocessing – NaN Handling	6
3 Analysis	8
3.1 Daily Active Power	8
3.2 Sub-metering	10
4 Feature Generation and Data Splitting	14
4.1 Feature Generation	14
4.2 Data Splitting and Feature Scaling	15
5 Baseline Models	17
5.1 Moving Average	17
5.2 Exponential Smoothing	18
5.3 ARIMA	18
6 Other Models	20
6.1 SARIMA	20
6.2 Random Forest Regression	21
6.3 XGBoost	22
6.4 LSTM	22
7 Discussion	25
7.1 Comparison of Model Performance Metrics	25
7.2 Comparing Model Predictions and Actual Data	26
7.3 Research Questions	28
7.4 Further Work	29
8 Conclusion	30
Bibliography	31

List of Figures

3.1	Daily global active power with 7-day and 30-day moving average filters applied. The smoothed curves highlight seasonal variations and overall stability in household energy consumption.	8
3.2	Average weekly pattern of global active power consumption, showing clear day-to-day variations and higher usage during weekends.	9
3.3	Average daily global active power consumption. Power usage peaks during morning and evening hours, corresponding to typical household activity periods.	9
3.4	Heat map showing the average global active power per hour and day of the week. Higher energy usage occurs during evening hours, with the most pronounced peak on Sunday around 20:00.	10
3.5	30-day moving average of daily sub-metering consumption. <code>Sub_metering_3</code> (water heater and air conditioner) dominates total energy use, showing clear seasonal variation with higher values during winter months.	11
3.6	Average daily consumption patterns for the three sub-metering components. <code>Sub_metering_3</code> shows clear morning and evening peaks, while <code>Sub_metering_1</code> and <code>Sub_metering_2</code> remain relatively stable throughout the day.	11
3.7	Heat map of <code>Sub_metering_3</code> showing average hourly energy consumption during winter months (November–February). Higher and more evenly distributed usage indicates continuous heating demand throughout the day.	12
3.8	Heat map of <code>Sub_metering_3</code> showing average hourly energy consumption outside the winter period. Power usage is more concentrated in the morning and evening, reflecting reduced heating activity.	12
4.1	Correlation matrix of all selected features. <code>Sub_metering_3</code> shows the strongest correlation with <code>Global_active_power</code> , confirming its dominant impact on total household energy consumption.	14
4.2	Visualization of the data split into training and test sets for <code>Global_active_power</code> . The training set covers approximately 80% of the data, while the remaining 20% is used for model validation.	15
4.3	Visualization of one input sequence and its target for model training. The figure shows how several input features, e.g., voltage and sub-metering values are used to predict the future/target(bottom row) <code>Global_active_power</code> over a 24-hour horizon.	16
5.1	Moving Average forecast compared to actual values for the first sequence. The model produces a constant prediction based on the mean of previous observations, effectively smoothing short-term variations but failing to capture rapid changes in demand.	17

5.2	Exponential Smoothing forecast compared to actual values for the first sequence. The model adapts to short-term variations and captures seasonal patterns more effectively than the Moving Average, but still lags slightly behind rapid fluctuations in demand.	18
5.3	ARIMA forecast compared to actual values for the first sequence. The model captures the overall trend but lags in adapting to rapid short-term changes, reflecting its linear nature and limitations in modeling strong seasonality.	19
6.1	SARIMA forecast compared to actual values for the first sequence. The model captures repeating seasonal patterns more effectively than standard ARIMA, but still smooths out short-term fluctuations in power demand.	20
6.2	Random Forest prediction for the first test sequence. The ensemble model captures the overall consumption pattern but slightly smooths local fluctuations due to averaging across multiple decision trees.	21
6.3	XGBoost prediction for the first test sequence. The model captures short-term variations more effectively than Random Forest, reflecting its sequential boosting approach that iteratively corrects previous prediction errors.	22
6.4	Training and validation loss for the LSTM model. The decreasing loss curves indicate that the model gradually learns temporal dependencies without overfitting, as the validation loss stabilizes at a low value.	23
6.5	LSTM prediction for the first test sequence. The model captures short- and medium-term fluctuations in power consumption more effectively than tree-based models, demonstrating its ability to learn temporal patterns.	24
7.1	Comparison of model performance based on Mean Absolute Percentage Error (MAPE) and Mean Squared Error (MSE). The best-performing models appear in the lower-left corner of the plot, where both MSE and MAPE are low.	25
7.2	Model predictions compared to actual test data for Sequence 0. The LSTM model closely follows the real consumption trend, while SARIMA underestimates and Random Forest tends to over-smooth fluctuations.	26
7.3	Model predictions compared to actual test data for Sequence 2. The LSTM and XGBoost models show better alignment with the observed variations than SARIMA, which remains too rigid in its forecasts.	27
7.4	Model predictions compared to actual test data for Sequence 47. LSTM continues to capture short-term variations accurately, while SARIMA and tree-based models struggle to reproduce rapid fluctuations in demand.	27

Chapter 1

Introduction

As global energy consumption continues to rise, improving energy efficiency and managing household electricity usage have become increasingly important. Smart home technologies, combined with data-driven forecasting models, enable both households and grid operators to predict energy demand and optimize energy distribution. Accurate forecasting not only helps reduce operational costs and environmental impact but also supports the integration of renewable energy sources and enhances grid stability.

In recent years, the proliferation of smart meters and Internet of Things (IoT) devices has resulted in the availability of high-resolution time series data on household electricity consumption. Leveraging this data, advanced analytical techniques can uncover usage patterns, detect anomalies, and provide actionable insights for both consumers and energy providers. However, household energy consumption is influenced by various factors, including weather conditions, occupancy patterns, and appliance usage, making accurate prediction a challenging task.

This project focuses on analyzing and forecasting hourly household energy consumption using a rolling window of 168 hours (24 hours \times 7 days) of historical power measurements. The approach involves comprehensive data preprocessing, visualization, and the application of several forecasting models to identify consumption patterns and predict energy demand for the next 24 hours.

The models explored in this report include Moving Average (MA), Autoregressive Integrated Moving Average (ARIMA), Seasonal ARIMA (SARIMA), Long Short-Term Memory (LSTM) neural networks, eXtreme Gradient Boosting (XGBoost), and Random Forest. Each method is evaluated using the same dataset to ensure fair comparison of performance metrics, and their respective strengths and weaknesses are discussed.

1.1 Research Problem

The main research problem of this study is to investigate how statistical and machine learning models can be applied to analyze household electricity consumption patterns and accurately forecast future energy demand using historical data.

To address this, the following research questions have been formulated:

1. How does data preprocessing, including resampling, outlier removal, and power factor validation, affect the accuracy and reliability of the forecasting models?
2. What correlations and temporal patterns, including daily, weekly, and seasonal trends, can be identified in the household's total and sub-metered electricity consumption?
3. How do traditional statistical models compare to machine learning approaches in predicting energy demand?

1.2 Dataset Description

The dataset, titled “*Individual Household Electric Power Consumption*”, was originally collected by EDF R&D (France) and provided by data.world, Inc. at Kaggle [1]. It contains measurements collected from a single household between December 2006 and November 2010, with a temporal resolution of one measurement per minute.

It is important to note that the dataset does not directly measure energy consumption, but rather active power P expressed in kilowatts [kW].

Active power represents the instantaneous rate at which electrical energy is converted or consumed. To estimate the corresponding energy usage over each one-minute interval, the active power is converted to watt-hours [Wh] according to:

$$E = P \cdot \frac{1000}{60}$$

where P is the active power in kilowatts [kW], multiplied by 1000 to convert to watts [W], and divided by 60 to represent watt-hours per minute.

The total household energy consumption not captured by the three submeters can therefore be estimated as:

$$(\text{global_active_power} \cdot \frac{1000}{60}) - \text{sub_metering_1} - \text{sub_metering_2} - \text{sub_metering_3}$$

This represents active energy (Wh per minute) used by unmetered equipment. The dataset includes approximately 1.25% missing values in the measurements, though all timestamps are continuous. Missing entries appear as empty fields between semicolons (e.g., April 28, 2007).

To ensure completeness and consistency, the group acquired the full version of the dataset from the UCI Machine Learning Repository [2].

1.2.1 Dataset Columns

Attribute	Description
date	Date in format <i>dd/mm/yyyy</i>
time	Time in format <i>hh:mm:ss</i>
global_active_power	Household global minute-averaged active power (<i>kW</i>)
global_reactive_power	Household global minute-averaged reactive power (<i>kVAr</i>)
voltage	Minute-averaged voltage (<i>V</i>)
global_intensity	Household global minute-averaged current intensity (<i>A</i>)
sub_metering_1	Kitchen consumption (<i>Wh</i>) - dishwasher, oven, microwave
sub_metering_2	Laundry room consumption (<i>Wh</i>) - washing machine, dryer, fridge
sub_metering_3	Electric water heater and air conditioner (<i>Wh</i>)

1.3 Predictions for the Household Power Consumption Dataset

Before beginning the assignment, we wanted to make our own predictions about the Household Power Consumption dataset. The goal of this exercise was to think critically about what patterns we might expect to see in the data before analyzing it.

By discussing and formulating hypotheses in advance, we could compare our assumptions to the actual results later on, helping us better understand the underlying relationships between household activities, time, and energy usage. This predictive reasoning step also encouraged a more structured and analytical approach to exploring the dataset, and helped us more easily identify obvious anomalies.

Global Active Power

Daily Pattern: We predict that global active power will vary with the household's occupancy patterns. Usage will be lowest during work hours and sleeping hours, and highest when residents are home and awake.

Weekly Pattern: Following the daily pattern, we expect higher usage during weekends, as residents spend more time at home.

Monthly Pattern: We predict increased consumption during the winter months, when residents are more likely to stay indoors and use heating appliances that require substantial energy.

Sub Metering 1

Appliance Context: This meter corresponds to the kitchen area. As the oven and hot plates are gas-powered, electricity consumption will mainly come from smaller kitchen appliances.

Daily Pattern: Usage should peak during meal preparation times—early in the morning and around 19:00–20:00 in the evening.

Weekly Pattern: Slightly higher usage is expected on weekends, as people tend to spend more time cooking then.

Monthly Pattern: A modest increase is expected during colder months, since people spend more time indoors and cook at home more frequently.

Sub Metering 2

Appliance Context: This meter corresponds to the laundry room.

Daily Pattern: Usage is expected to be irregular throughout the day, with minimal activity during nighttime hours.

Weekly Pattern: We anticipate higher usage on weekends, when residents are likely to catch up on laundry and other household chores.

Monthly Pattern: No strong seasonal trend is expected, as laundry frequency is generally consistent throughout the year.

Sub Metering 3

Appliance Context: This meter measures electricity usage from the electric water heater and air conditioner, the dominating consumption source.

Daily Pattern: We expect the highest usage when residents are home—particularly in the morning (as heating systems start up) and in the evening after work.

Weekly Pattern: Usage is likely to peak on weekends, aligning with increased time spent at home.

Monthly Pattern: Since heating typically consumes more energy than cooling, we predict the highest values during winter months.

1.4 Models in the Project

1.4.1 Base Models

- **Moving Average**
- **Exponential Smoothing**
- **ARIMA** (AutoRegressive Integrated Moving Average)

1.4.2 Other Models

- **SARIMA** (Seasonal AutoRegressive Integrated Moving Average)
- **Random Forest Regression**
- **XGBoost** (Extreme Gradient Boosting)
- **LSTM** (Long Short-Term Memory)

Chapter 2

Preprocessing

2.1 Importing Libraries

The project utilizes a range of Python libraries for data handling, visualization, and modeling. `Pandas` and `NumPy` are used for efficient data manipulation and numerical operations, while `Matplotlib`, `Seaborn`, and `Plotly` provide visualizations for data analysis. Time series models are implemented using the `statsmodels` and `pmdarima` libraries, which support ARIMA and Exponential Smoothing methods.

Machine learning and deep learning models are built using `scikit-learn`, `TensorFlow`, and `Keras`, with additional optimization performed via `keras-tuner`. `XGBoost` and `RandomForestRegressor` are included for ensemble-based regression tasks. The `tqdm` library was used to visualize progress bars during training loops, and standard preprocessing tools such as `StandardScaler` were imported for data normalization.

2.2 Importing Data

The dataset was first inspected for non-numeric and blank values. All detected entries were counted and examined to determine whether they occurred simultaneously. Each time a non-numerical value appears in the dataset, the entire row is non-numerical except for the `Date` and `Time` columns. This indicates that the issue is not local to individual household measurements but is likely related to the power grid or data transmission.

2.3 Preprocessing – NaN Handling

To better interpret the presence and impact of missing values, the duration of consecutive missing data spans was analyzed. The seven longest gaps in the dataset range from approximately 14 hours to 5 days. The five-day gap is most likely caused by extended communication failures, meter downtime, or scheduled maintenance rather than continuous power outages. The remaining gaps, all shorter than two hours, are more consistent with brief power interruptions or temporary transmission errors.

A new `Datetime` column was created by combining the existing `Date` and `Time` columns, which was then set as the dataset index. This allows the dataset to be treated as a true

time series, simplifying sorting, filtering, and resampling by specific time intervals such as hours, days, or months.

All missing or non-numeric entries were converted to NaN values and subsequently removed. Non-numeric rows were deleted rather than filled, in order to avoid introducing artificial or misleading data into the model.

Finally, the columns `Global_reactive_power`, `Global_intensity`, `Time`, and `Date` were removed, as they are not used in further analysis.

After cleaning and selecting the relevant columns, the dataset was resampled into hourly, daily, and monthly intervals. For each new period, `Global_active_power` and `Voltage` were calculated as mean values, while the sub-metering variables were aggregated as the sum of all readings within each period.

Chapter 3

Analysis

3.1 Daily Active Power

We start by analysing and plotting the global active power. Figure 3.1 shows global active power with and without a 7-days/30-days moving average filter.

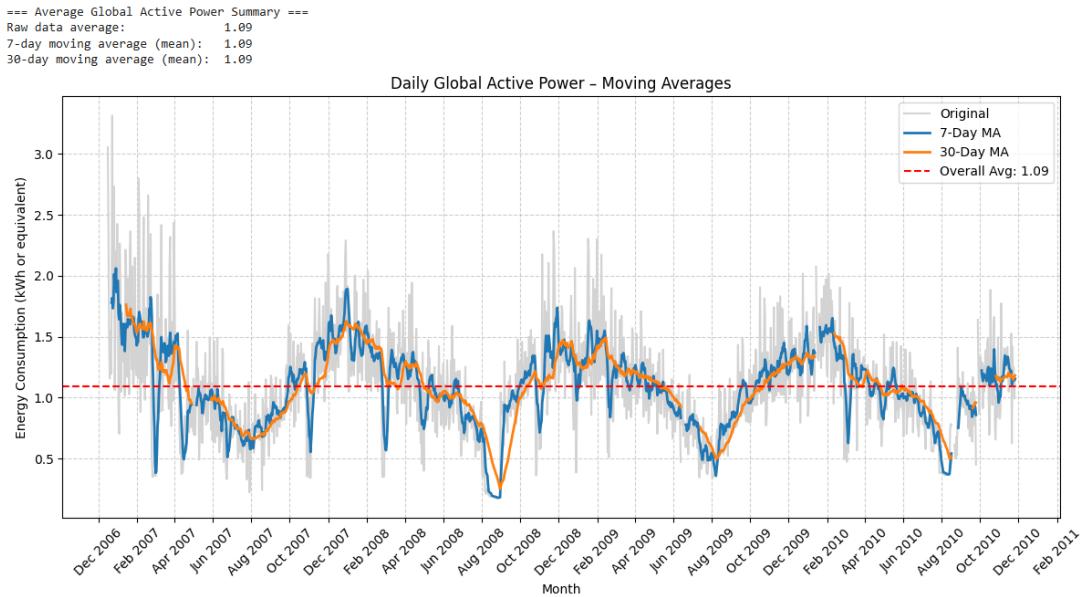


Figure 3.1: Daily global active power with 7-day and 30-day moving average filters applied. The smoothed curves highlight seasonal variations and overall stability in household energy consumption.

The results show that household energy consumption has remained relatively stable over time. The raw data average of 1090 Wh is the same as the smoothed values, with the 7-day and 30 day moving average also at 1090 Wh. You can clearly observe a seasonal pattern in the global active power, with higher energy consumption during the winter months (October to February) compared to the rest of the year. The higher consumption during winter is likely a result of increased energy demand caused by colder temperatures. One can also observe that the dataset is missing significant amounts of data in the last 25% of the dataset.

To better observe the dynamics of the power consumption, we can plot and analyze the average daily and average hourly global active power. This is shown in figure 3.2 and 3.3

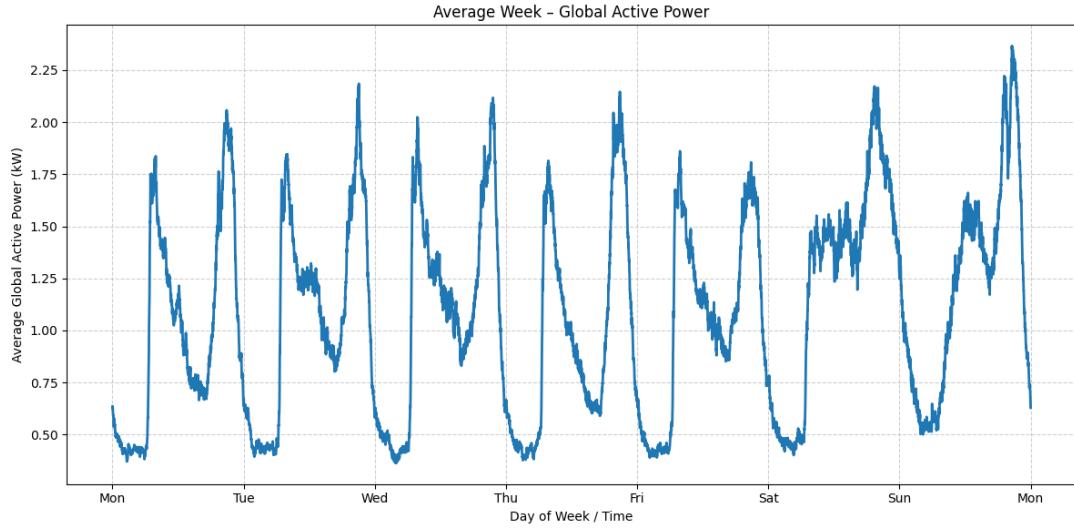


Figure 3.2: Average weekly pattern of global active power consumption, showing clear day-to-day variations and higher usage during weekends.

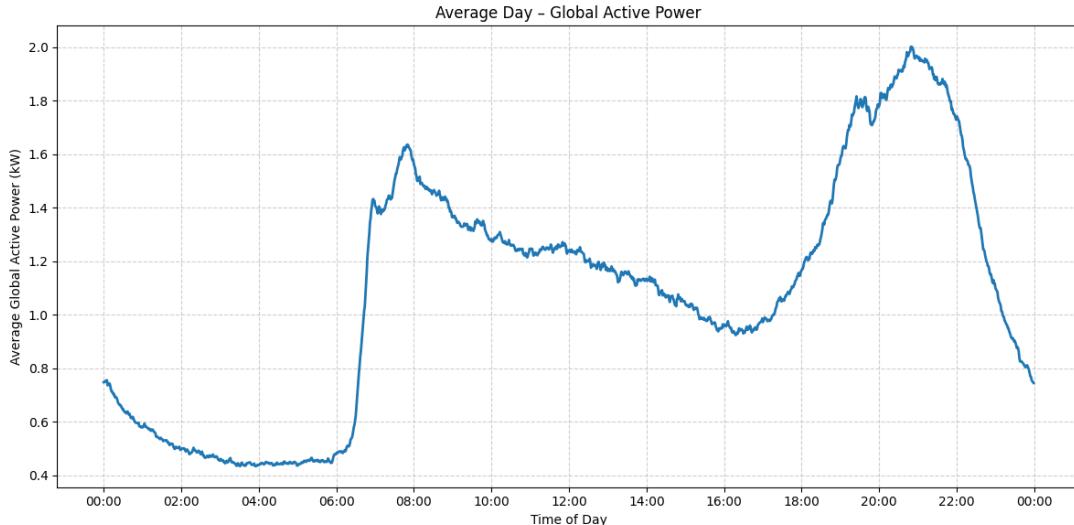


Figure 3.3: Average daily global active power consumption. Power usage peaks during morning and evening hours, corresponding to typical household activity periods.

For both figure 3.2 and 3.3, we can see that the consumption is higher during "rush hour", which are mornings from 07:00 - 09:00, and evenings from 18:30-22:30, approximately. During the day, the most amount of power is used during the evening. From the first plot, showing the average week, one can observe that the power consumption is higher during weekends, and the most significant peak shown, is on Sunday. So the highest power consumption will be at Sunday at 20:30. To summarize these two plots in one, a heat map showing the global active power per hour and day of the week is shown in figure 3.4.

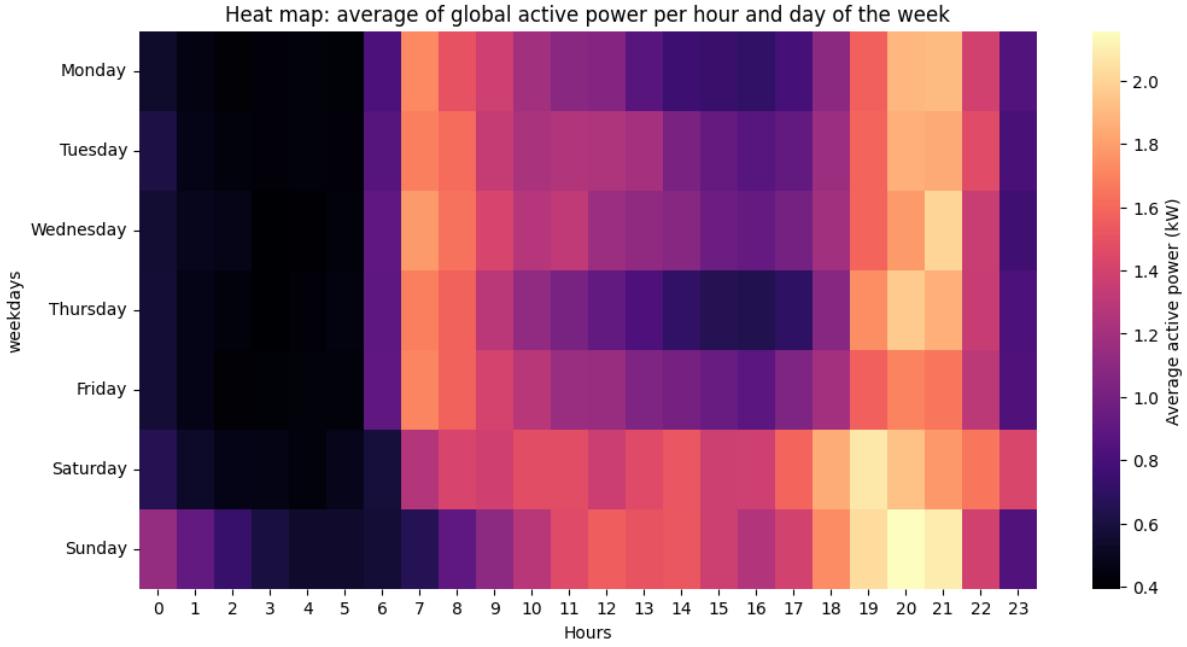


Figure 3.4: Heat map showing the average global active power per hour and day of the week. Higher energy usage occurs during evening hours, with the most pronounced peak on Sunday around 20:00.

The plotted heat map confirms these observations, showing that the largest amount of energy is used on Sunday around 20:00.

3.2 Sub-metering

To gain deeper insight into which household activities drive this pattern, we analyze the three sub-metering components (`Sub_metering_1`, `Sub_metering_2`, and `Sub_metering_3`) in the following section. `Sub_metering_1` corresponds to the kitchen, containing mainly a dishwasher, an oven, and a microwave (the hot plates are gas-powered). `Sub_metering_2` corresponds to the laundry room, containing a washing machine, a tumble dryer, a refrigerator, and a light. `Sub_metering_3` corresponds to an electric water heater and an air conditioner.

The 30-day moving averages for the three sub-metering components are shown in figure 3.5.

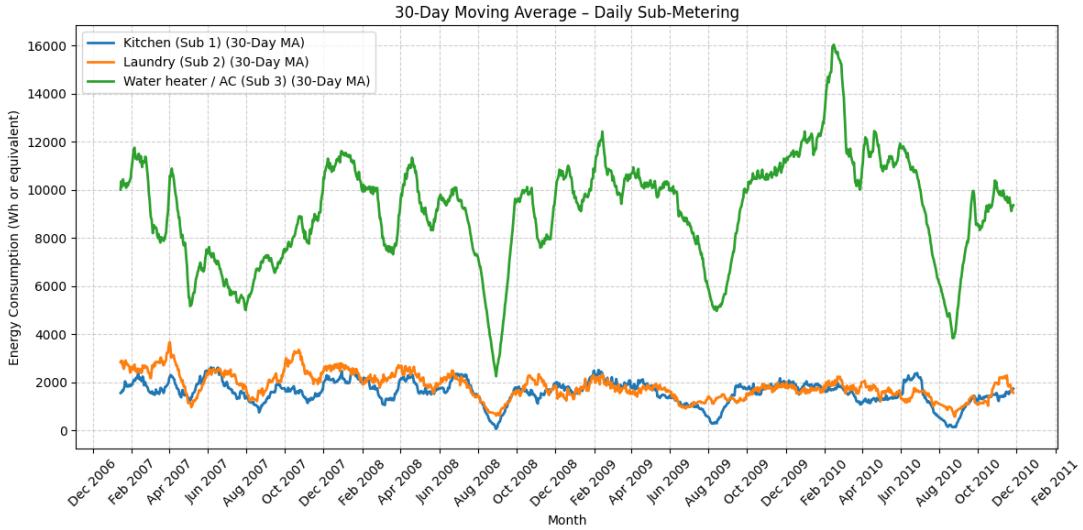


Figure 3.5: 30-day moving average of daily sub-metering consumption. `Sub_metering_3` (water heater and air conditioner) dominates total energy use, showing clear seasonal variation with higher values during winter months.

To better observe the patterns, a 30-day moving average filter is applied. As expected, the water heater and air conditioner consume three to four times more power than the kitchen and laundry room. It can also be observed that the laundry room, on average, uses more power than the kitchen.

From a seasonal perspective, `Sub_metering_3` (water heater and air conditioner) peaks during the winter months (November to February), while the power usage in the kitchen and laundry room remains relatively stable. This indicates that the global active power is mainly influenced by the dynamics of `Sub_metering_3`.

The average daily consumption patterns for the three sub-meters are shown in figure 3.6.

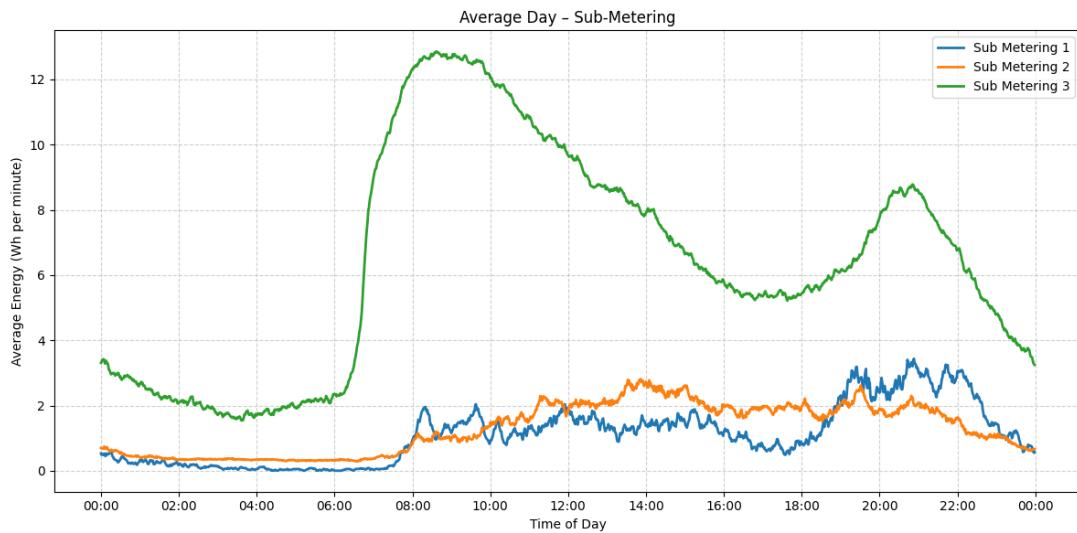


Figure 3.6: Average daily consumption patterns for the three sub-metering components. `Sub_metering_3` shows clear morning and evening peaks, while `Sub_metering_1` and `Sub_metering_2` remain relatively stable throughout the day.

For `Sub_metering_3`, one can observe that the most energy used by the water heater and air-conditioner is in the morning, then it gradually decreases before another high in the evening. For the `Sub_metering_2`, its relatively flat from the morning to the evening, and really low during the night. This makes sense, because `Sub_metering_2` contains lights and laundry, which generally will not be used during night. For the last one, its also really low during night, then relatively flat from 08:00 to 18:00, then increases in the evening. Probably because the kitchen gets used more in the evening after work.

Because of the high influence `Sub_metering_3` has on the global active power, we want to plot a heat map of `Sub_metering_3` during winter months (nov-feb) and compare it to the rest of the year. The comparison is shown in figure 3.7 and 3.8.

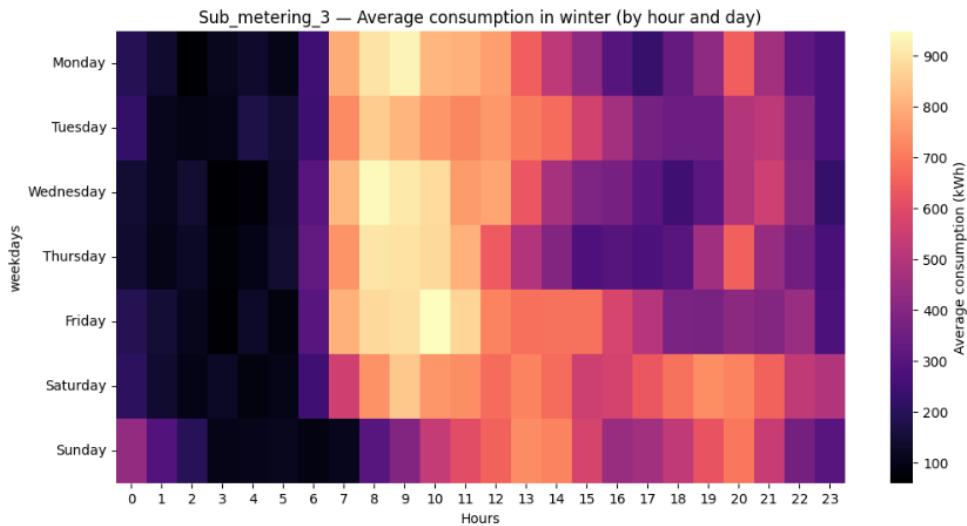


Figure 3.7: Heat map of `Sub_metering_3` showing average hourly energy consumption during winter months (November–February). Higher and more evenly distributed usage indicates continuous heating demand throughout the day.

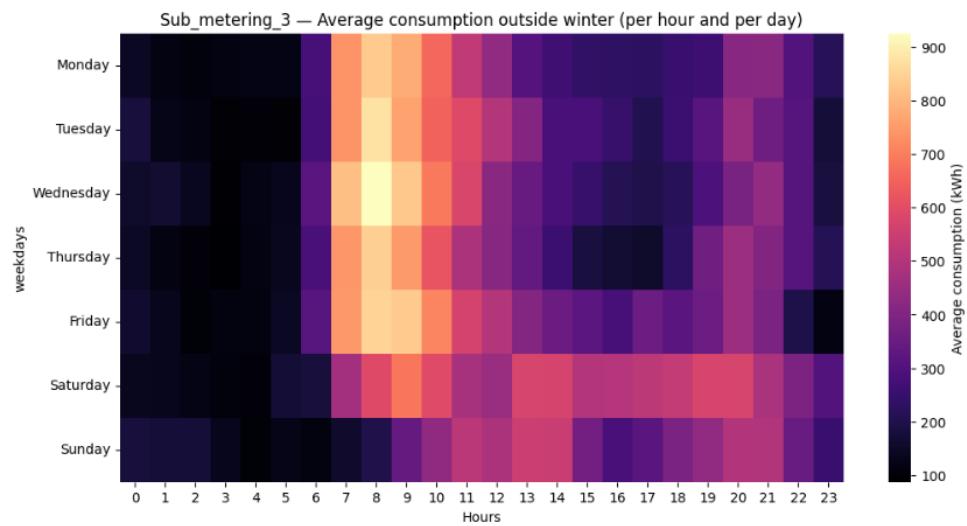


Figure 3.8: Heat map of `Sub_metering_3` showing average hourly energy consumption outside the winter period. Power usage is more concentrated in the morning and evening, reflecting reduced heating activity.

The most noticeable difference between the two heat maps, is that the values is more evenly distributed throughout the day in the winter months. This is most likely because of the air conditioner, which is used as a cooler during the hot months and as a heater during the winter, whereas the latter most likely uses the most power. Looking at the heat maps during morning hours, we can say that the power used most likely comes from the water heater (morning showering), because the two heat maps is similar in these particular hours, meaning that there is not a big difference if between winter month and rest, again meaning that this comes from the water heater. From the winter heat map, we can see that the heater uses more energy in the starting prosess, but also uses energy to keep it warm during the day, explaing the brighter areas of the heat map later in the day.

Feature Generation and Data Splitting

4.1 Feature Generation

This part is to show the different correlations of different features in a correlation matrix, giving a good foundation for the model-making, later in the report. The corelation matrix, shown in figure 4.1, allows us to check and quantify our previous analysis.

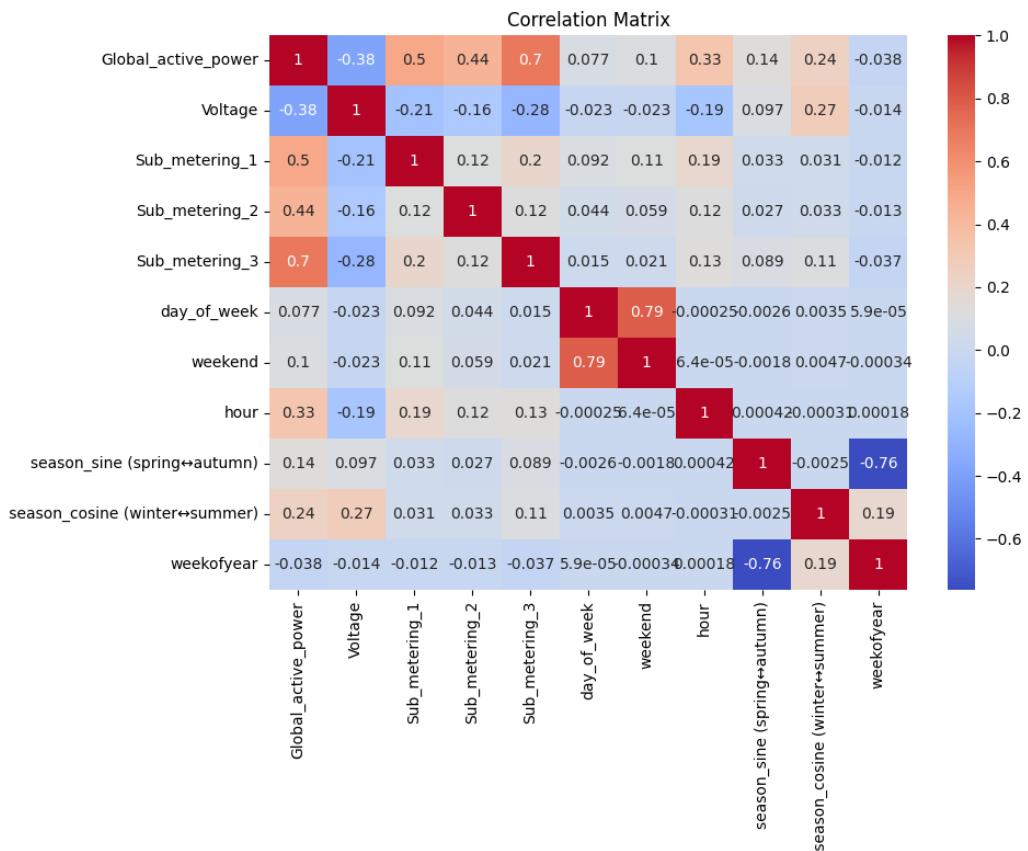


Figure 4.1: Correlation matrix of all selected features. `Sub_metering_3` shows the strongest correlation with `Global_active_power`, confirming its dominant impact on total household energy consumption.

- The correlation between `Sub_metering_3` and `Global_active_power` is 0.7. This confirms our earlier observation that `Sub_metering_3` has a strong impact on total consumption (highest correlation value). As `Sub_metering_3` increases, so does `Global_active_power`, indicating a positive correlation.
 - The correlations between `Sub_metering_1/Sub_metering_2` and `Global_active_power` are 0.5 and 0.44, respectively. These values align with our previous analysis, showing that they have a smaller effect on total consumption than `Sub_metering_3` ($0.5/0.44 < 0.68$). As both `Sub_metering_1` and `Sub_metering_2` increase, the total active power also rises, reflecting a positive relationship.
 - The correlation between `Hours` and `Global_active_power` is 0.33. This indicates a moderate influence of the time of day on energy consumption, consistent with previous findings. The correlation with `Month` (particularly winter months) is slightly lower at 0.24, while `Weekend` also shows an influence with a correlation value of 0.11. The difference between autumn and spring also impacts overall active power (0.14).
- These values therefore confirm our previous analyses.

4.2 Data Splitting and Feature Scaling

The dataset is divided into training and testing segments to ensure that the forecasting models are evaluated on data they have not previously seen. In time-series forecasting, it is essential to preserve the chronological order of observations; therefore, the split is performed sequentially rather than randomly. As illustrated in Figure 4.2, the first 80% of the data is used for training, allowing the models to learn patterns such as daily and weekly cycles, seasonal behavior, and the relationships between the target variable and auxiliary features (e.g., voltage and sub-metering values). The remaining 20% is held out as the test set, providing an unbiased estimate of how well the models generalize to future periods.

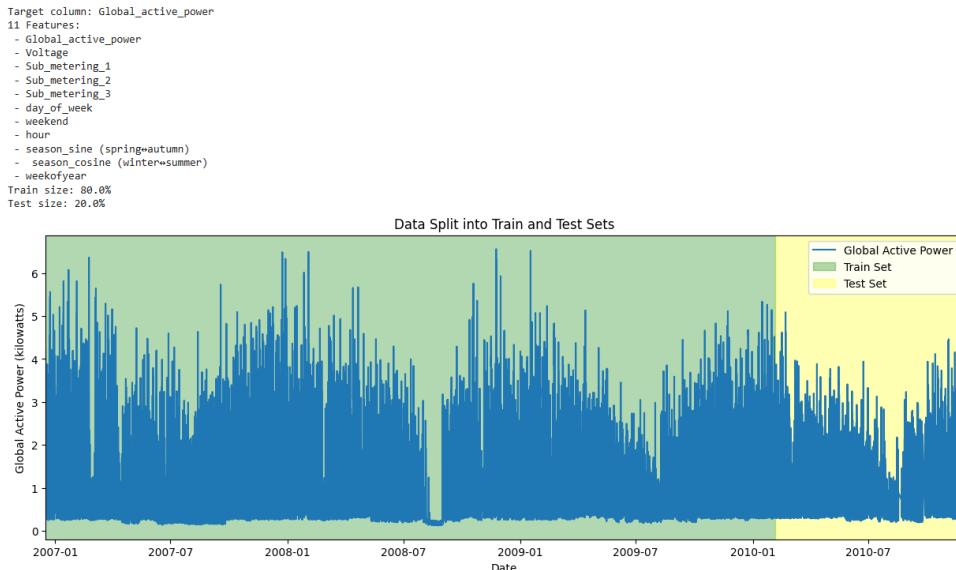


Figure 4.2: Visualization of the data split into training and test sets for `Global_active_power`. The training set covers approximately 80% of the data, while the remaining 20% is used for model validation.

Visualizing the split confirms that the models are always trained on past observations and evaluated on truly later timestamps. After the chronological train–test division, the data is further transformed into supervised learning sequences. As illustrated in Figure 4.3, power-consumption forecasting relies on sliding windows that convert continuous time-series data into fixed-length input/output pairs. Each input window contains one week of hourly measurements ($24 \times 7 = 168$ points), and the models are tasked with predicting the next 24 hours. This structure mirrors realistic forecasting scenarios in which recent historical behavior is used to estimate short-term future consumption.

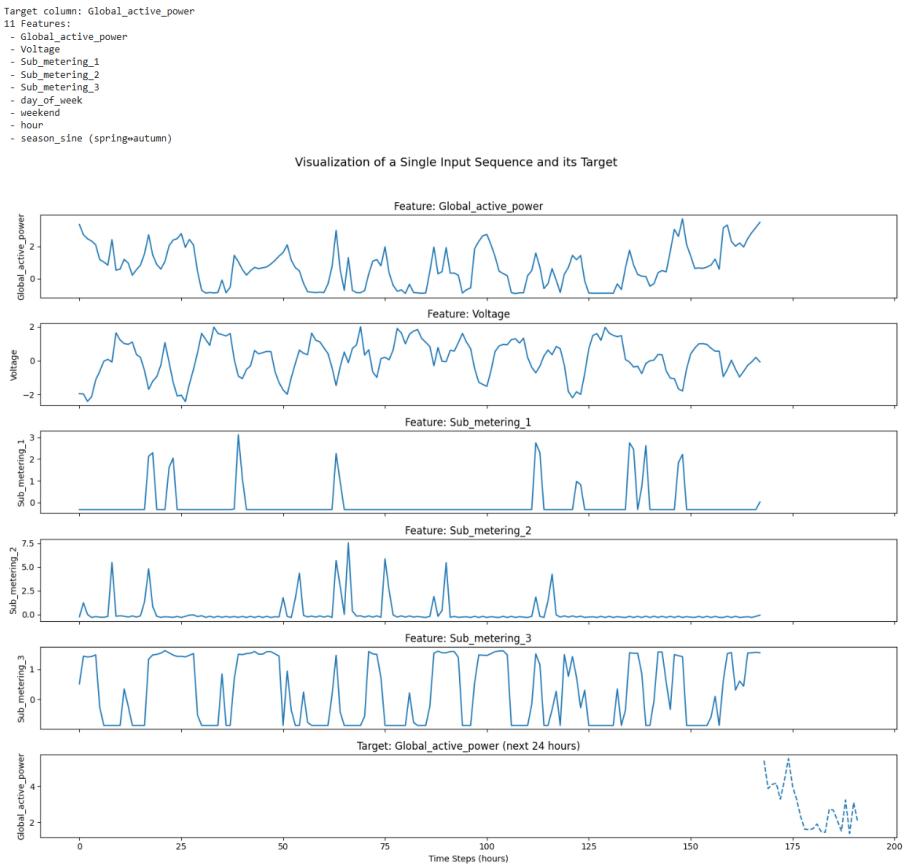


Figure 4.3: Visualization of one input sequence and its target for model training. The figure shows how several input features, e.g., voltage and sub-metering values are used to predict the future/target(bottom row) `Global_active_power` over a 24-hour horizon.

Feature scaling is applied prior to model training to ensure that all input variables are on a comparable scale. Standardization is performed using statistics computed from the training set only, which prevents variables with large numerical ranges from dominating the learning process and avoids data leakage from the test set. This consistent scaling accelerates model convergence and improves overall performance.

Separate sequence datasets are generated for scaled inputs (used by machine learning models) and unscaled values (used by statistical models such as ARIMA and SARIMA). This allows different model families to operate on data in the form they require while maintaining consistent window and horizon definitions. Printing the final sequence shapes verifies that the transformation produces the intended number of samples and aligns each input window with its corresponding future target segment.

Chapter 5

Baseline Models

5.1 Moving Average

The Moving Average model forecasts future values as the mean of recent observations. It smooths short-term fluctuations and highlights the underlying trend, making it a simple baseline for comparison with more advanced forecasting models. In this implementation, the prediction for each forecast horizon is the mean of the last observed values within the input sequence, as shown in figure 5.1.

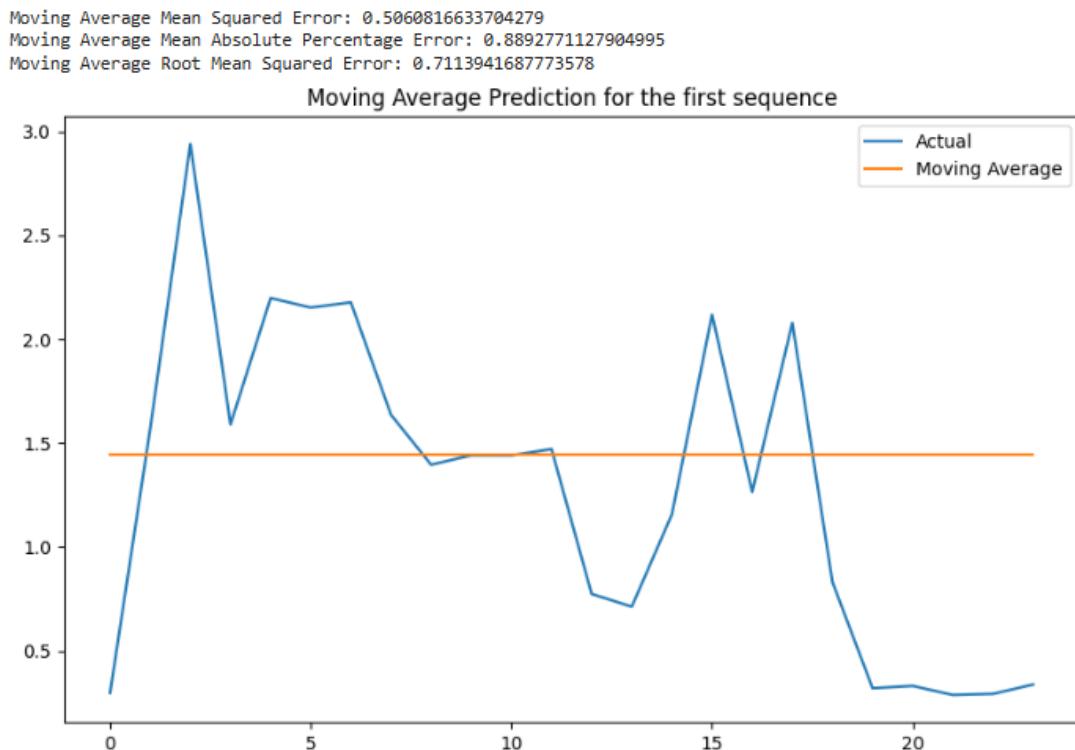


Figure 5.1: Moving Average forecast compared to actual values for the first sequence. The model produces a constant prediction based on the mean of previous observations, effectively smoothing short-term variations but failing to capture rapid changes in demand.

5.2 Exponential Smoothing

Exponential Smoothing forecasts future values by assigning greater weight to recent observations while exponentially decreasing the influence of older data [3]. This allows the model to adapt to short-term trends and seasonal variations. In this implementation, the model is refitted sequentially for each test sequence, using a seasonal additive component with a 24-hour period to capture daily consumption patterns, as shown in figure 5.2.

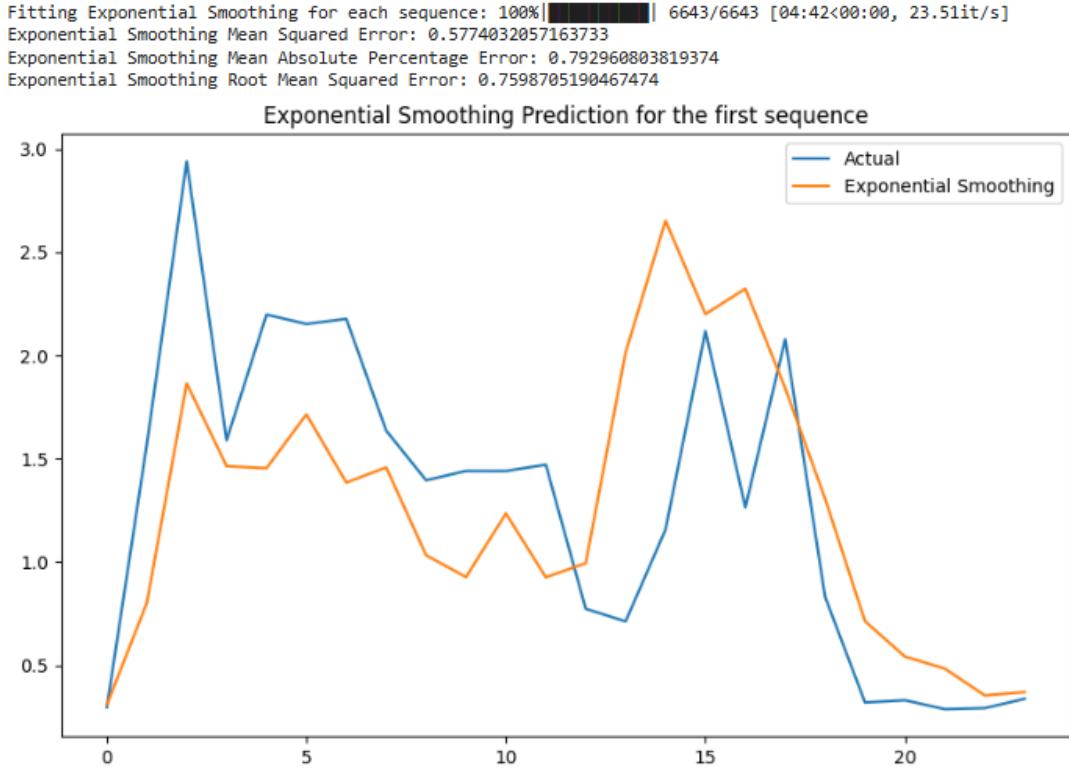


Figure 5.2: Exponential Smoothing forecast compared to actual values for the first sequence. The model adapts to short-term variations and captures seasonal patterns more effectively than the Moving Average, but still lags slightly behind rapid fluctuations in demand.

5.3 ARIMA

The model used in this section is an Auto-Regressive Integrated Moving Average (ARIMA) model. This is a classical statistical forecasting approach widely used for modeling time series data [4].

ARIMA captures three key components of a time-dependent signal:

- Autoregression (AR): how the current value depends on previous values.
- Integration (I): the degree of differencing required to remove trends and make the data stationary,
- Moving Average (MA): how the current value depends on previous forecast errors.

The ARIMA model receives a one-dimensional sequence of past consumption values as input and produces a forecast of the next consumption values as output.

To select the best model configuration, we used the `auto_arima` function from the `pm-darima` library. This function automatically determines the optimal parameters (p, d, q) by testing multiple combinations and selecting the one that minimizes the Akaike Information Criterion (AIC). Once the optimal configuration was identified, the model was refitted on each test sequence, and predictions were generated for the following time steps, as shown in fig 5.3.

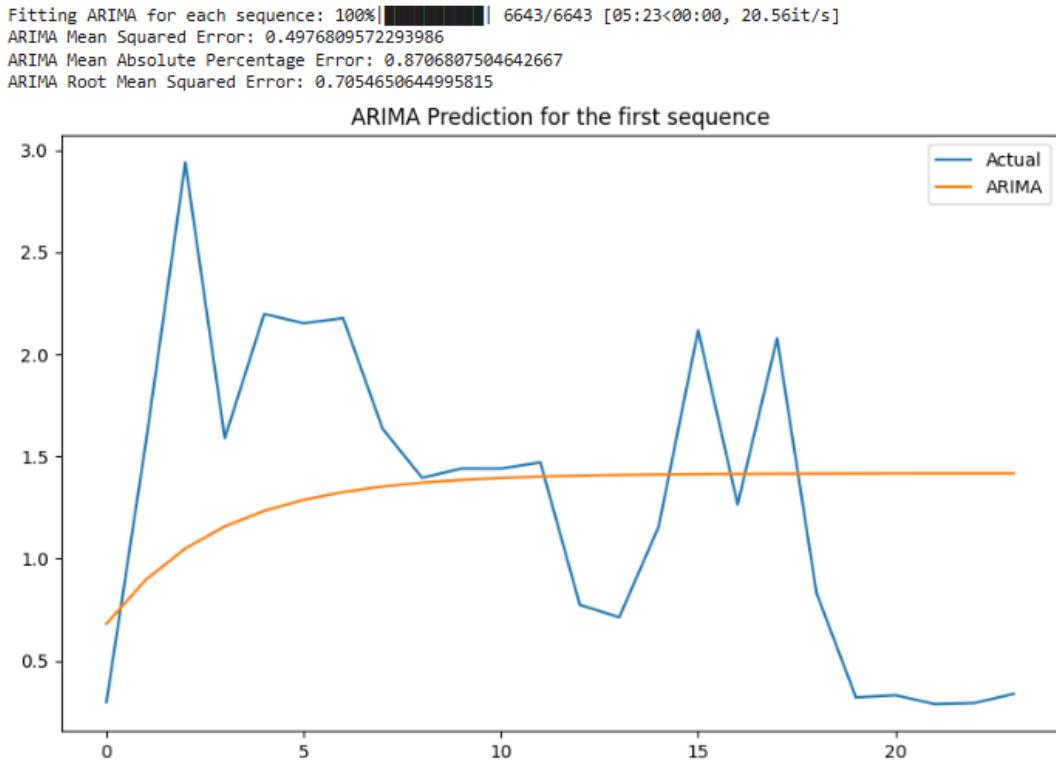


Figure 5.3: ARIMA forecast compared to actual values for the first sequence. The model captures the overall trend but lags in adapting to rapid short-term changes, reflecting its linear nature and limitations in modeling strong seasonality.

Training an ARIMA model is computationally inexpensive compared to machine learning models such as neural networks. It involves fitting a few statistical parameters through optimization rather than gradient-based learning. However, because the model was fitted individually for each sequence, the total computation time scales linearly with the number of sequences analyzed. ARIMA provides a solid statistical baseline that performs well on data with linear temporal relationships but may struggle with strong seasonality or nonlinearity.

Chapter 6

Other Models

6.1 SARIMA

The Seasonal Auto-Regressive Integrated Moving Average (SARIMA) model extends the standard ARIMA model by adding parameters that capture seasonal effects in the data [5]. While ARIMA models short-term dependencies and trends, SARIMA introduces seasonal autoregressive and moving average terms to represent recurring patterns such as daily or weekly cycles. The model is defined by the parameters;

$$(p, d, q) \times (P, D, Q, m),$$

where the uppercase terms (P, D, Q) and m represent the seasonal part of the model. As with ARIMA, the `auto_arima` function was used to determine optimal non-seasonal and seasonal parameters. The input remains a one-dimensional time series, but the model now accounts for periodic structures in the sequence, improving accuracy when the data exhibit strong seasonal behavior, as illustrated in figure 6.1.

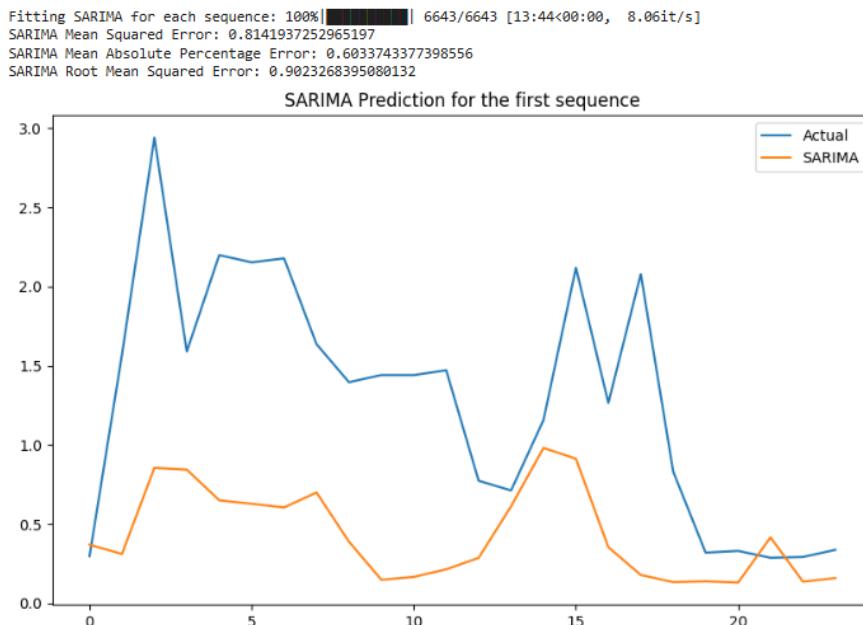


Figure 6.1: SARIMA forecast compared to actual values for the first sequence. The model captures repeating seasonal patterns more effectively than standard ARIMA, but still smooths out short-term fluctuations in power demand.

Training a SARIMA model is like training the ARIMA, computationally efficient, compared to most machine learning models. The main additional computational expense comes from estimating the seasonal parameters, but this remains far less demanding than training complex neural networks. SARIMA is particularly effective for forecasting stationary time series with clear seasonality, making it a suitable and interpretable choice for modeling cyclical energy consumption patterns.

6.2 Random Forest Regression

The model used in this section is a *Random Forest Regressor*, a supervised learning method based on a set of decision trees [6]. It is a *non-linear* statistical model that is particularly robust and effective for tabular data. Each tree in the forest learns from a subset of the data and variables, and the final prediction is obtained by averaging the outputs of all the trees. This approach greatly reduces the risk of overfitting and improves the stability of the model, as illustrated in figure 6.2.



Figure 6.2: Random Forest prediction for the first test sequence. The ensemble model captures the overall consumption pattern but slightly smooths local fluctuations due to averaging across multiple decision trees.

In our case, the data represents the electricity consumption of a house in the form of time series. Since a Random Forest can only process two-dimensional tables, each sequence is "flattened", i.e., transformed into a vector containing all the successive values. The model therefore receives an input of the form `(n_samples, n_features)`, where each row corresponds to a consumption sequence transformed into a vector. It then learns to associate these sequences with the future consumption value that we want to predict. The output is therefore a *vector of values*, representing the future evolution of consumption over the targeted time horizon.

Training a Random Forest is relatively *inexpensive*: complexity increases with the number of trees and their depth, but remains much lower than that of a neural network. In our configuration, the model has *200 trees* with a *maximum depth of 30*, which provides a reasonable compromise between performance and computational cost.

6.3 XGBoost

The second model we used is the *XGBoost model*, which works slightly differently from Random Forest [7]. It is a *statistical algorithm* based on a set of *decision trees constructed sequentially*. Each new tree is trained to correct the errors made by the previous trees, resulting in a powerful, non-linear model that performs particularly well on tabular data.

For input and output, it is essentially the same as Random Forest, with a flattened input (a single large vector) and an output value vector, as illustrated in figure 6.3.

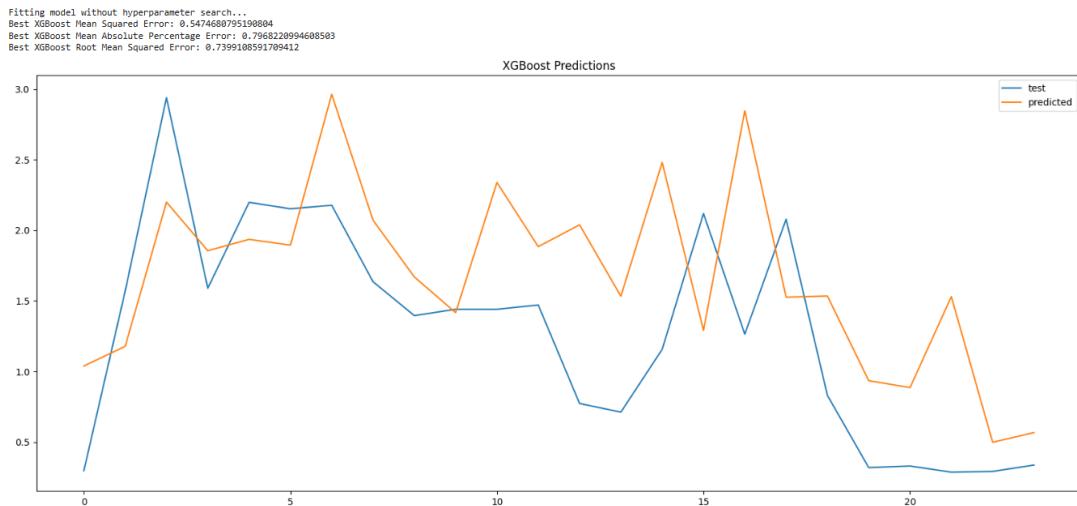


Figure 6.3: XGBoost prediction for the first test sequence. The model captures short-term variations more effectively than Random Forest, reflecting its sequential boosting approach that iteratively corrects previous prediction errors.

Finally, training an XGBoost model is *more expensive* than Random Forest, mainly due to the sequential nature of boosting, but it remains *very efficient and optimized*, especially for moderate data sizes.

6.4 LSTM

In this section, the model used is an LSTM model (Long Short-Term Memory), a *recurrent neural network* designed to process sequential data [8]. Unlike the statistical models used previously (Random Forest and XGBoost), LSTMs are able to take into account the temporal dimension of the data by storing useful information over several time steps thanks to their internal “memory cell” structure.

The architecture of an LSTM model is highly flexible: one can vary the number of layers, include dropout regularization, or adjust the number of hidden units. The architecture presented below was obtained after extensive experimentation and was found to deliver the best performance. The development process involved iteratively designing candidate architectures, training them with early stopping to reduce overfitting, and comparing their performance across multiple evaluation metrics. Once a promising architecture was identified, `keras_tuner` was employed for hyperparameter optimization. This tool automatically fine-tunes model parameters—such as the number of units, learning rate,

epochs, and batch size—within user-defined bounds to maximize performance.

The model receives as input a tensor of shape `(n_samples, n_window_size, n_features)`, where `n_window_size` corresponds to the size of the time window analyzed, and `n_features` to the number of variables available at each moment. The first block of the network is a 128-unit LSTM encoder, responsible for scanning the time window step by step and extracting a compact representation: a *state vector* summarizing the evolution of the sequence.

This representation is then duplicated using a *RepeatVector* to generate an artificial sequence of length `n_forecast_size`, corresponding to the desired prediction horizon. This new sequence is processed by a 512-unit LSTM decoder, which gradually produces a series of vectors describing the possible future states of the system. Finally, several dense layers refine these representations and produce a final output consisting of one value per predicted time step, resulting in an output of shape `(n_samples, n_forecast_size)`.

The network structure is summarized in the table in figure 6.4. In the same figure, the graph illustrates the evolution of the loss for both the training and test datasets. The training loss decreases steadily, dropping from approximately 0.78 to 0.43 by the end of training. The validation loss remains stable around 0.36–0.37, indicating that the model generalizes well to unseen data without clear signs of overfitting. This stable behavior suggests that the chosen architecture is effective at capturing the temporal dynamics of the data while maintaining strong predictive performance.

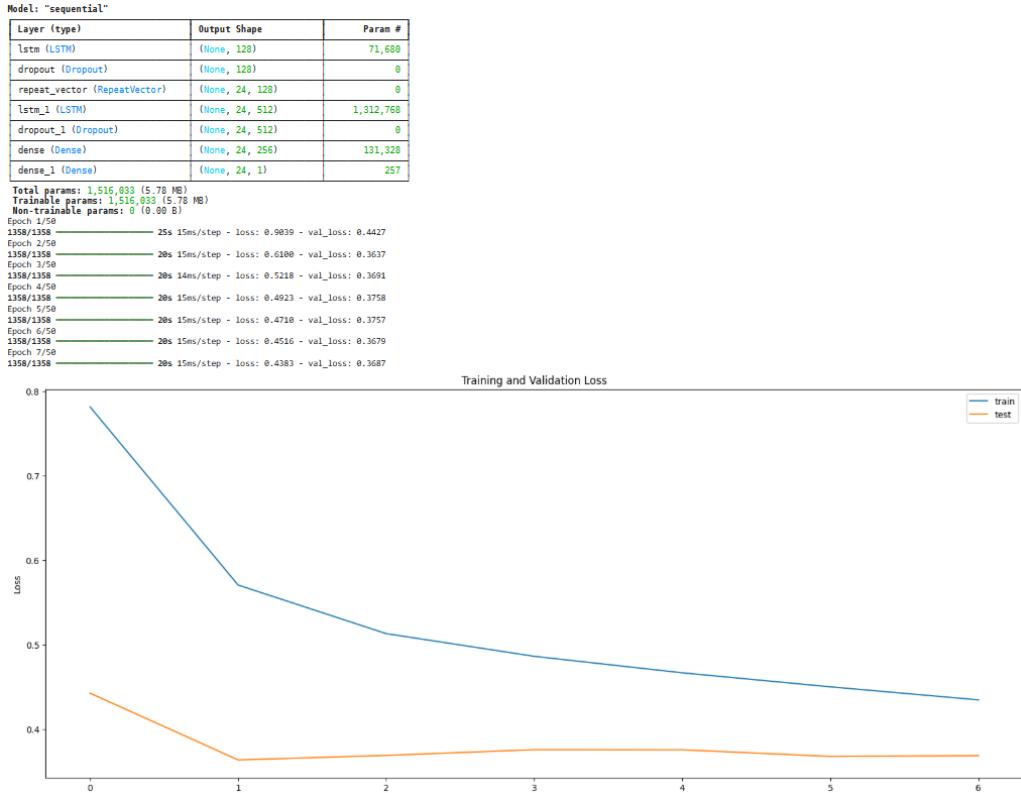


Figure 6.4: Training and validation loss for the LSTM model. The decreasing loss curves indicate that the model gradually learns temporal dependencies without overfitting, as the validation loss stabilizes at a low value.

Thus, the prediction of the global active power on the first test sequence obtained by this LSTM model is shown in figure 6.5 below.

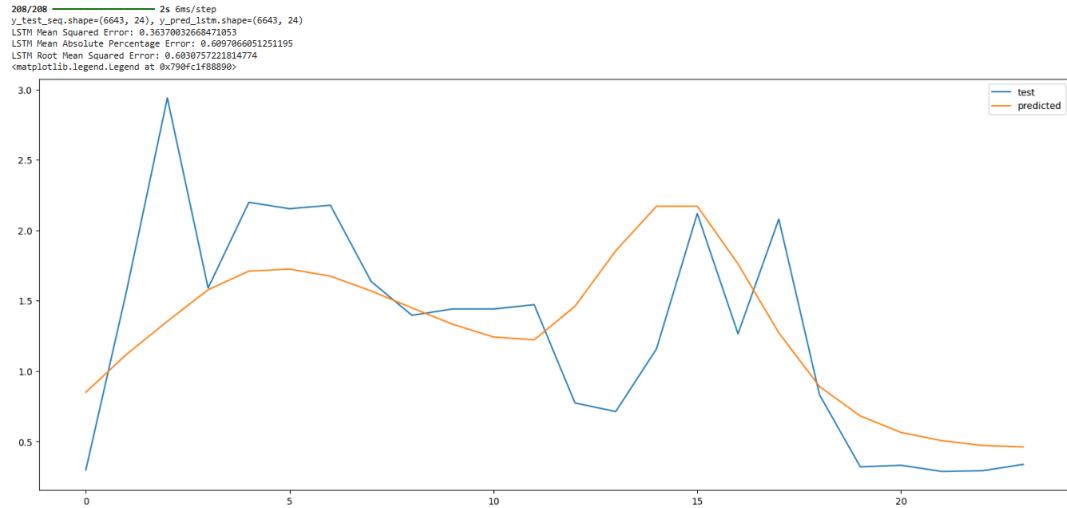


Figure 6.5: LSTM prediction for the first test sequence. The model captures short- and medium-term fluctuations in power consumption more effectively than tree-based models, demonstrating its ability to learn temporal patterns.

Being based on deep neural networks, this model is significantly more computationally expensive than decision tree-based approaches. It requires numerous parameters, sequential information propagation, and often longer training times, especially when the model includes large LSTM layers as it does here. However, this computational complexity is justified by the LSTM's ability to effectively model temporal dynamics and provide consistent predictions on evolving data.

Chapter 7

Discussion

7.1 Comparison of Model Performance Metrics

Figure 7.1 compares the performance of all forecasting models based on their Mean Absolute Percentage Error (MAPE) and Mean Squared Error (MSE). Models positioned closer to the bottom-left corner indicate better overall performance, as they combine low prediction error with reduced variance.

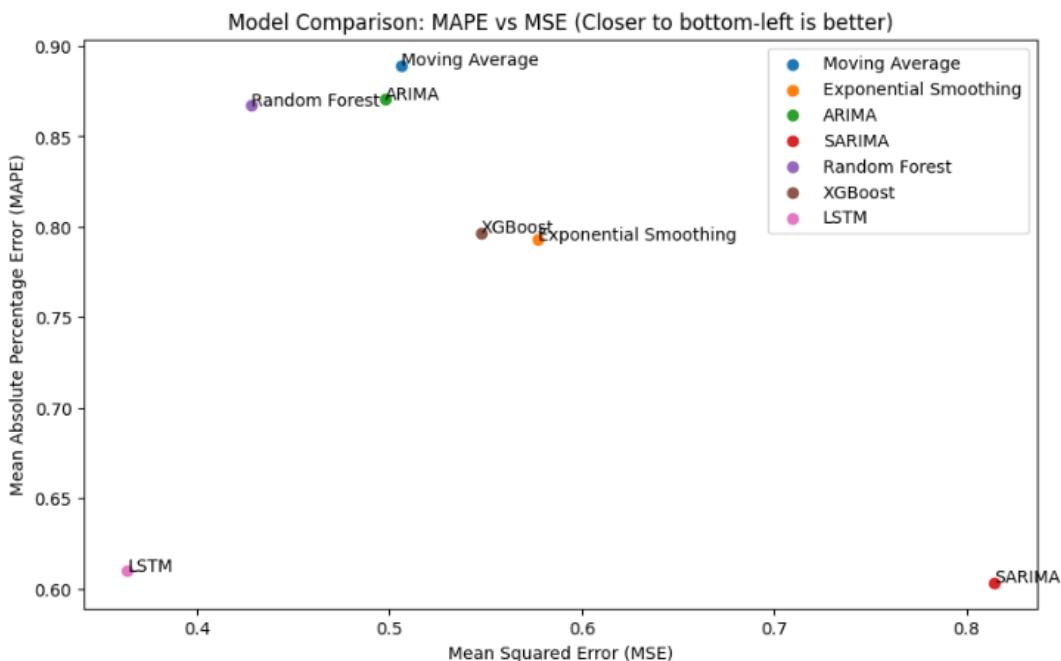


Figure 7.1: Comparison of model performance based on Mean Absolute Percentage Error (MAPE) and Mean Squared Error (MSE). The best-performing models appear in the lower-left corner of the plot, where both MSE and MAPE are low.

The figure above compares the performance of different models in terms of Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE). The MSE measures absolute accuracy by heavily penalizing large deviations, while the MAPE expresses the average relative error in percentage, allowing fair comparison across models regardless of data scale.

The best-performing models appear in the lower-left corner of the plot, where both MSE and MAPE are low. The LSTM clearly stands out with the lowest MSE (≈ 0.36) and a MAPE of (≈ 0.61), demonstrating its ability to effectively capture complex temporal dependencies. The SARIMA model, although showing a relatively low MAPE (≈ 0.60), has a higher MSE (≈ 0.81), indicating occasional large prediction errors. Other models such as ARIMA, Exponential Smoothing, Random Forest, and XGBoost achieve intermediate or less consistent results.

Overall, the LSTM provides the best balance between absolute and relative accuracy, making it the most effective model for this forecasting task.

7.2 Comparing Model Predictions and Actual Data

Figures 7.2–7.4 visualize how the different models track actual household consumption for selected sequences. Each sequence represents a 24-hour forecast window, allowing a direct comparison between observed and predicted power values.

In figure 7.2, the LSTM model clearly follows the real consumption pattern, closely matching both peaks and downs throughout the day. In contrast, SARIMA consistently underestimates demand, particularly during high-consumption periods, while Random Forest over-smooths the data, failing to capture rapid fluctuations.

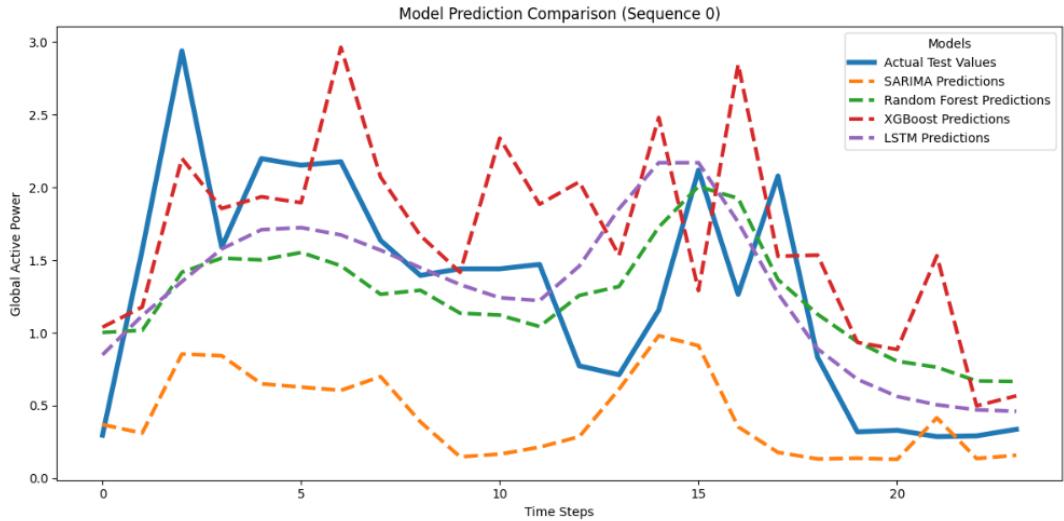


Figure 7.2: Model predictions compared to actual test data for Sequence 0. The LSTM model closely follows the real consumption trend, while SARIMA underestimates and Random Forest tends to over-smooth fluctuations.

Figure 7.3 further reinforces this pattern. Here, both LSTM and XGBoost provide a better fit to the observed data, effectively capturing variations in load behavior. SARIMA again appears too constrained, producing forecasts that fail to adjust quickly to short-term changes in consumption. This rigidity is a direct result of its reliance on fixed seasonal cycles, which limits responsiveness to day-to-day variations.

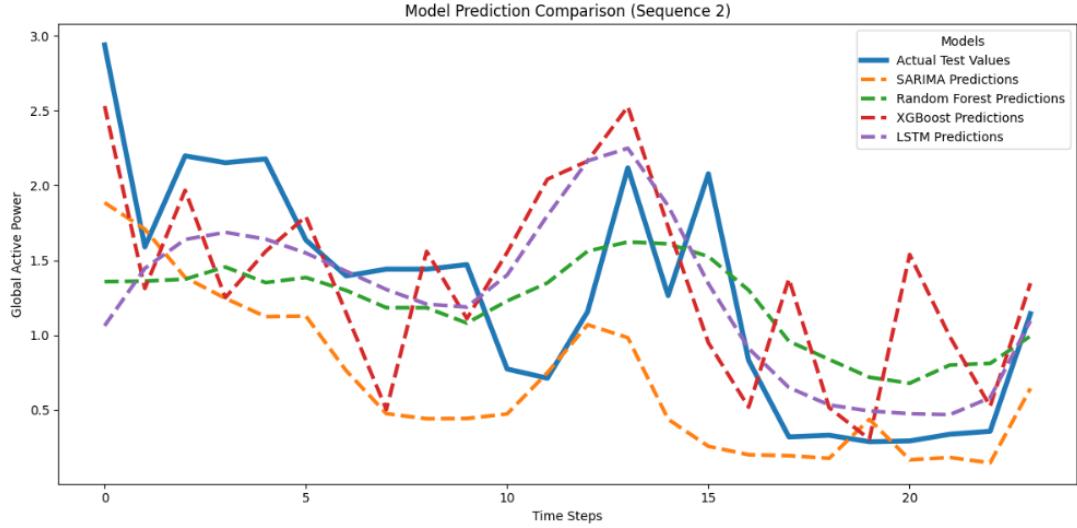


Figure 7.3: Model predictions compared to actual test data for Sequence 2. The LSTM and XGBoost models show better alignment with the observed variations than SARIMA, which remains too rigid in its forecasts.

Finally, figure 7.4 shows that LSTM maintains its predictive strength even for more complex consumption sequences. It captures both amplitude and timing of peaks with reasonable accuracy, while tree-based models (Random Forest and XGBoost) tend to lag slightly in transitions. SARIMA continues to underperform, producing overly smoothed curves that diverge from the actual pattern.

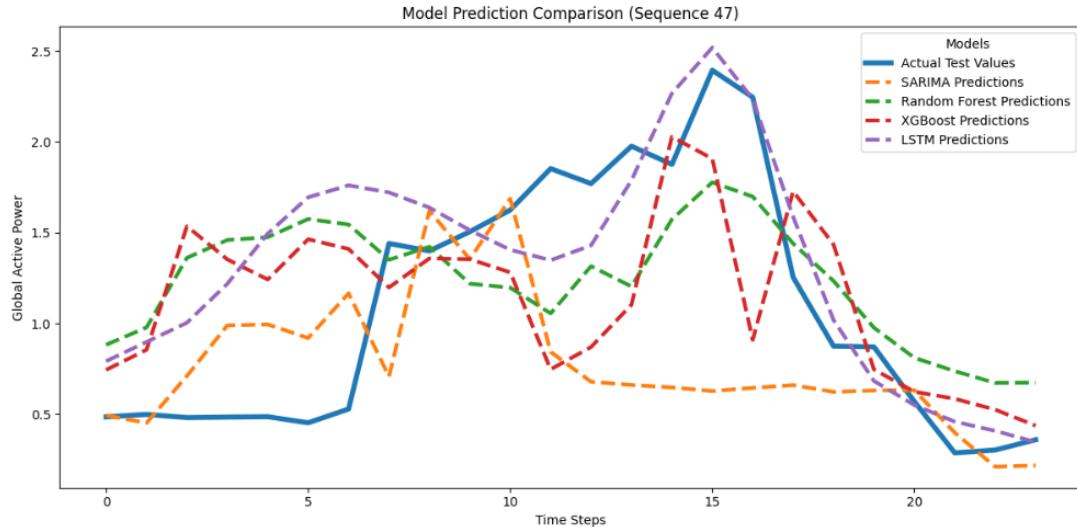


Figure 7.4: Model predictions compared to actual test data for Sequence 47. LSTM continues to capture short-term variations accurately, while SARIMA and tree-based models struggle to reproduce rapid fluctuations in demand.

Across all test sequences, the LSTM model stands out for its ability to capture nonlinear and time-dependent behavior in the data, leading to more realistic and dynamically responsive forecasts. The visual comparison aligns well with the quantitative performance metrics discussed earlier, confirming that while traditional models can approximate general trends, only the LSTM architecture successfully models the complex temporal dynamics of household power consumption.

7.3 Research Questions

1. How does data preprocessing, including resampling and outlier removal affect the accuracy and reliability of the analysis and the forecasting models?

Working with a data set of this size, it is important to ensure the quality of data we put into our system, to secure quality of the output data. The preprocessing and data cleaning phase must therefore be the first step. The main objective is to make sure the values contained in the data set is workable. There are several approaches to this step, there are several ways to interpolate the missing data found in the dataset, but this would in practice be false data. The argument must also be made that the nature of the missing data actually reflects the nature of the industry, as power outages and communication errors frequently occur. Therefore, we chose to exclude the 1.25% missing data rather than filling them with “trash data”. The same goes for outlier removal, as the outliers seem to be sequential rather than random, assumptions that there’s a logical cause must be made.

A big part of the project was to identify consumption patterns, and visualizing trends and seasonality, we aggregated the data to hourly, daily and monthly periods. These intervals allowed us to get a firm grasp of the households rhythm throughout the entirety of the year. Using these intervals we’ve been able to visualize when the occupants are home, and also their daily routines throughout. The output of our data shows a predictable output that reflects the repetitiveness of daily life.

2. What correlations and temporal patterns, including daily, weekly, and seasonal trends, can be identified in the household’s total and sub-metered electricity consumption?

Clear correlations and seasonal trends were identified in the household’s electrical parameters. The analysis showed that the energy consumption was generally higher during the winter months, reflecting increased use of heating and indoor appliances. Distinct daily patterns were also observed, with consumption peaking during morning hours (07:00-09:00) and evening hours (18:30-22:30), corresponding to typical household activity hours. Consumption was higher on weekends and reached its maximum on Sundays around 20:00. From the sub-metering analysis, it was found that the water heater and air conditioner were the main contributors to total power usage, consuming roughly three to four times more energy than the kitchen and laundry room. The water heater and air conditioner showed seasonal variation, with a higher consumption during the winter months, while the kitchen and laundry room remained relatively stable throughout the year. This indicates that the overall seasonality of the household power consumption is primarily driven by the usage patterns of the water heater and air conditioner.

3. How do traditional statistical models compare to machine learning approaches in predicting energy demand?

The comparative evaluation of forecasting models revealed important differences between traditional statistical approaches and modern machine learning techniques. Statistical models such as Moving Average, Exponential Smoothing, ARIMA, and SARIMA provided interpretable baselines and performed well in capturing regular temporal patterns

and seasonality. However, their predictive accuracy was limited when faced with complex, nonlinear relationships and interactions among multiple features.

Machine learning models, including Random Forest, XGBoost, and LSTM, demonstrated superior performance in forecasting future energy demand, especially when leveraging a broader set of input features and longer historical windows. These models were able to capture intricate dependencies and adapt to subtle changes in consumption behavior, resulting in lower forecasting errors and improved generalization to unseen data. Nevertheless, machine learning approaches required more extensive data preprocessing, feature scaling, and computational resources, and their interpretability was generally lower compared to statistical models.

7.4 Further Work

Future work and possible extensions of this study may include:

- **Model Optimization:** Investigate alternative tuning techniques or automated optimization frameworks to further improve model performance.
- **Feature Engineering:** Incorporate additional environmental or temporal features (e.g., temperature, holidays, or seasonality effects) to enhance forecasting accuracy.
- **Real-Time Implementation:** Explore integration of predictive models into smart home or grid-control systems for live forecasting and demand response applications.
- **Uncertainty Analysis:** Quantify confidence intervals in forecasts and evaluate probabilistic methods for more reliable decision-making.

Chapter 8

Conclusion

This study explored how statistical and machine learning models can be applied to both analyze and forecast household electricity consumption using historical data. Through a combination of data analysis, preprocessing and model evaluation, several key findings were obtained.

The thorough analysis reflects the repetitiveness of daily life well. As a first step, the group discussed what we expected and made a set of predictions. The output of our analysis matches our predictions well which validates our approach. Figures 3.6 shows a realistic and expected graph of `sub_metering_3`, as heating requires more energy initially and declines when maintaining the temperature. This presumption is confirmed in the heat map in figures 3.7 and 3.8. However, the figure shows unexpected curves for `sub_metering_1` and `sub_metering_2` for the time span 10-18 that did not match our predictions. Discussing probable causes for this, we concluded that this is probably somewhat shifted when residents are off work (i.e. vacations, weekends). Figure 3.5 shows a sine wave plot for `sub_metering_3` which oscillates with the seasons, also matching our predictions. The figure shows an abnormal spike in February 2010 which reflects the cold wave that hit Europe in 2010, which validates the output further [9].

In summary, this study highlights the importance of thorough data preprocessing, the value of exploratory analysis in understanding consumption patterns, and the strengths and limitations of different modeling approaches. For household energy forecasting, combining robust preprocessing with advanced machine learning models yields the most accurate and reliable predictions, while statistical models remain valuable for their simplicity and interpretability. These insights can inform future work on smart energy management and the development of data-driven solutions for optimizing household electricity usage.

Bibliography

- [1] Ortiz, J. (2024). *Individual Household Electric Power Consumption Dataset*. Retrieved from Kaggle, on 27 October 2025: <https://www.kaggle.com/datasets/imtkaggleteam/household-power-consumption>
- [2] UCI Machine Learning Repository. (2025). *Individual Household Electric Power Consumption Dataset*. Retrieved from UCI Machine Learning Repository, on 6 November 2025: https://archive.ics.uci.edu/ml/machine-learning-databases/00235/household_power_consumption.zip
- [3] GeeksforGeeks. (2023). *Exponential Smoothing for Time Series Forecasting*. Retrieved from GeeksforGeeks, on 3 November 2025: <https://www.geeksforgeeks.org/artificial-intelligence/exponential-smoothing-for-time-series-forecasting/>
- [4] GeeksforGeeks. (2025). *ARIMA*. Retrieved from GeeksforGeeks, on 3 November 2025: <https://www.geeksforgeeks.org/data-science/model-selection-for-arima/>
- [5] GeeksforGeeks. (2025). *SARIMA (Seasonal Auto-Regressive Integrated Moving Average)*. Retrieved from GeeksforGeeks, on 3 November 2025: <https://www.geeksforgeeks.org/machine-learning/sarima-seasonal-autoregressive-integrated-moving-average/>
- [6] GeeksforGeeks. (2025). *Random Forest Regression in Python*. Retrieved from GeeksforGeeks, on 4 November 2025: <https://www.geeksforgeeks.org/machine-learning/random-forest-regression-in-python/>
- [7] GeeksforGeeks. (2025). *XGBoost Algorithm for Machine Learning*. Retrieved from GeeksforGeeks, on 3 November 2025: <https://www.geeksforgeeks.org/machine-learning/xgboost/>
- [8] GeeksforGeeks. (2025). *Introduction to Long Short-Term Memory (LSTM) Networks*. Retrieved from GeeksforGeeks, on 3 November 2025: <https://www.geeksforgeeks.org/deep-learning/deep-learning-introduction-to-long-short-term-memory/>
- [9] Wikipédia. (2025). *Hiver 2009-2010 en Europe*. Retrieved from Wikipédia, on 12 November 2025: https://fr.wikipedia.org/wiki/Hiver_2009-2010_en_Europe