Energy Consumption Analysis and LSTM Modeling Report

Made by : Adnane Riyadi

Under supervision of : DR. Faouzi Tayalati

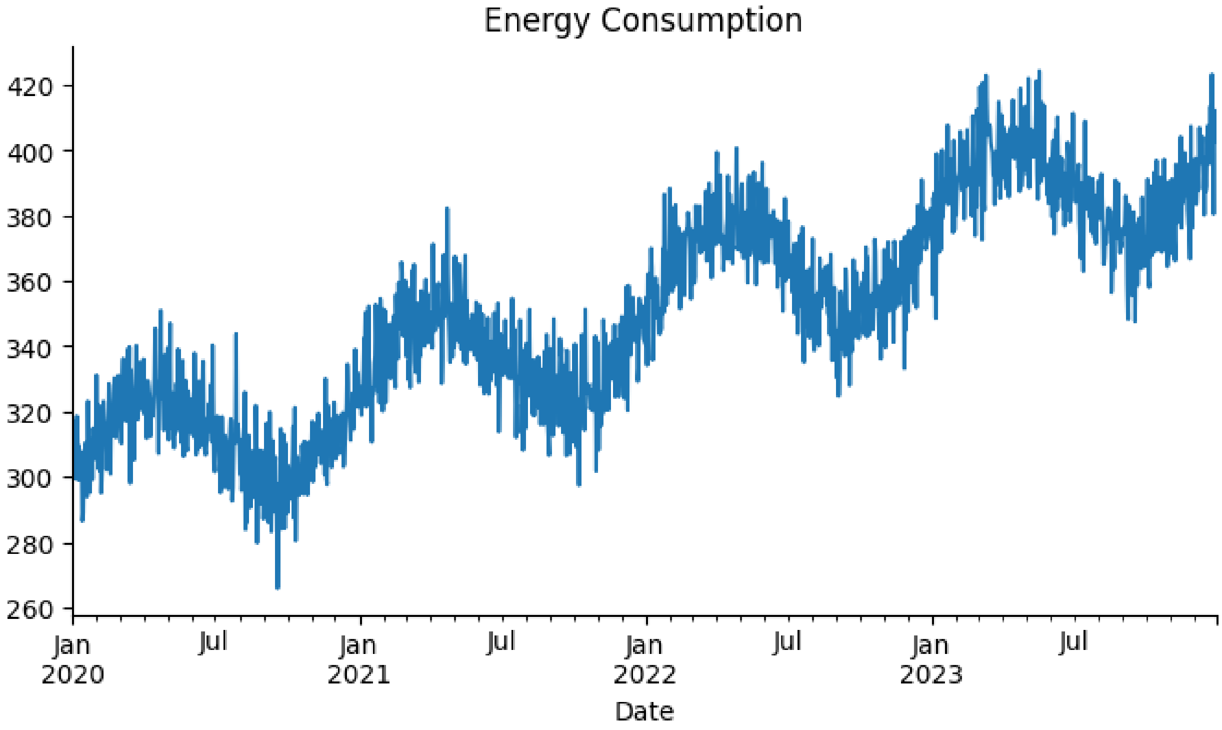
Contents

Understanding the Dataset

This report investigates an energy consumption dataset collected daily from early 2020 to late 2023. The dataset contains two columns: a Date column marking each day and an Energy Consumption column that records daily energy usage in units. Initial data exploration, using df.head(), confirmed a simple structure with daily entries over nearly four years

1. Initial Time Series Plot of Energy Consumption

To gain insight into the overall behaviour of energy consumption over time, the first step involved plotting energy usage from 2020 through 2023. This visualisation revealed several key characteristics in the data:



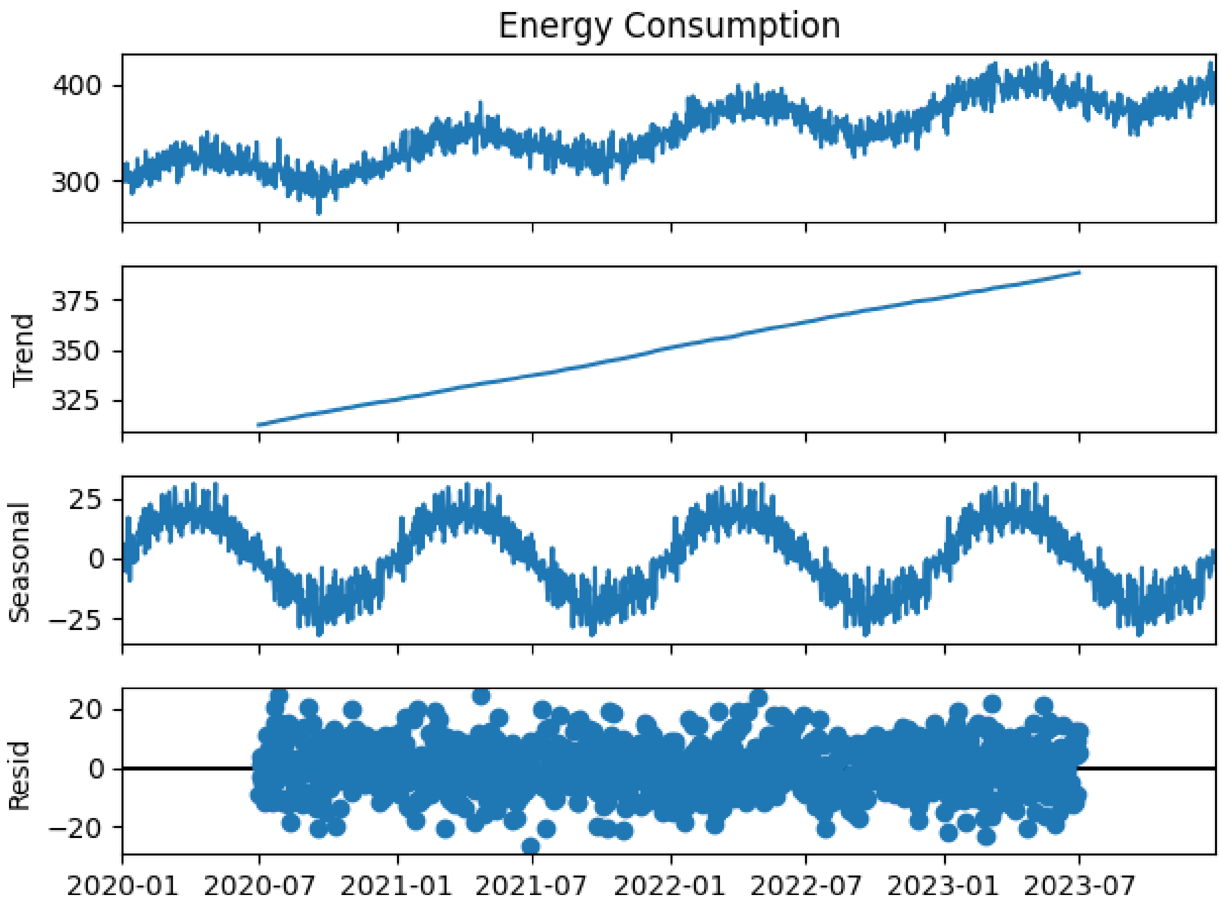
Overall Trend: There is a clear upward trend in energy consumption over the period, with values increasing from around 300 units at the beginning of 2020 to approximately 400 units by late 2023. This steady rise suggests a growing demand for energy over time.

Seasonal Patterns: A distinct seasonal pattern emerges in the data, where energy consumption decreases around mid-year, particularly in July, and peaks during winter months. This cycle repeats consistently each year, indicating higher energy demand in colder months.

Notable Features: Several significant points stand out in the plot:

- A pronounced dip in consumption occurs around mid-2020, marking the dataset's lowest point, with values dropping to approximately 270 units.
- The highest peaks are observed in 2023, reaching around 420 units, with larger fluctuations indicating more volatility, particularly in the last year.
- Overall, consumption fluctuates within a range of about 280 to 420 units, with the latter months of 2023 showing continued high levels and increased variability.

2. Seasonal Decomposition Analysis



To further dissect the dataset's structure, a seasonal decomposition was applied, breaking down the series into trend, seasonal, and residual components. The decomposition reveals the underlying dynamics of the time series and provides a clearer view of the patterns observed earlier.

- **Trend Component**: The trend line exhibits a steady linear increase, reflecting the overall upward movement in energy consumption. This gradual rise in the trend line suggests a fundamental growth in energy demand over time.
- **Seasonal Component**: The seasonal plot confirms the recurring annual pattern, with energy usage rising in winter and falling in summer. These seasonal fluctuations align with the broader observations from the initial time series plot, validating the yearly cycle of higher demand in colder periods.
- **Residual Component**: The residual plot captures deviations from the trend and seasonal components, highlighting short-term variations and anomalies. Notable residual spikes, such as a significant negative deviation in mid-2020, point to unusual events potentially caused by external factors influencing consumption during that period.

Together, these components provide a comprehensive understanding of the dataset's structure, capturing long-term trends, seasonal cycles, and short-term anomalies. This layered view lays the foundation for the next steps, where we will focus on preprocessing the data in preparation for model building.

Preprocessing the Dataset

In preparing the dataset for modelling, I developed a method to detect and replace outliers based on seasonal patterns. This approach was designed to identify values that deviate significantly from typical energy consumption levels while maintaining the dataset’s seasonal integrity.

The outlier detection process involves calculating a rolling median and interquartile range (IQR) to establish dynamic bounds for each point in time. For each data point, I used a rolling window to compute the median, providing a stable comparison value that adjusts with seasonal patterns. Using the IQR within this rolling window, upper and lower bounds are calculated, capturing most of the regular data points while flagging any significant deviations as potential outliers.

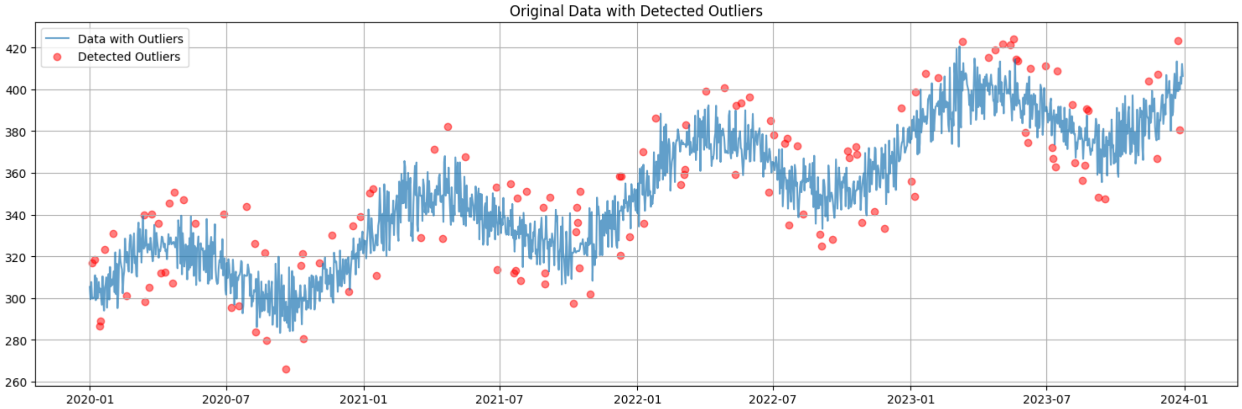
Once identified, outliers are replaced with interpolated values derived from seasonal patterns. For each detected outlier, a season-specific window is created, and non-outlier values within this window are selected to compute a median replacement value. This strategy ensures that the replacement aligns with the seasonality of the dataset. When there aren't enough non-outlier values in the seasonal window, the rolling median is used as a fallback to provide a reasonable estimate.

To illustrate this process, I created two visualisations.
The first plot highlights detected outliers on the original data.
The second shows the dataset with outliers replaced by seasonal-adjusted values.

1. Outlier Detection and Replacement Results

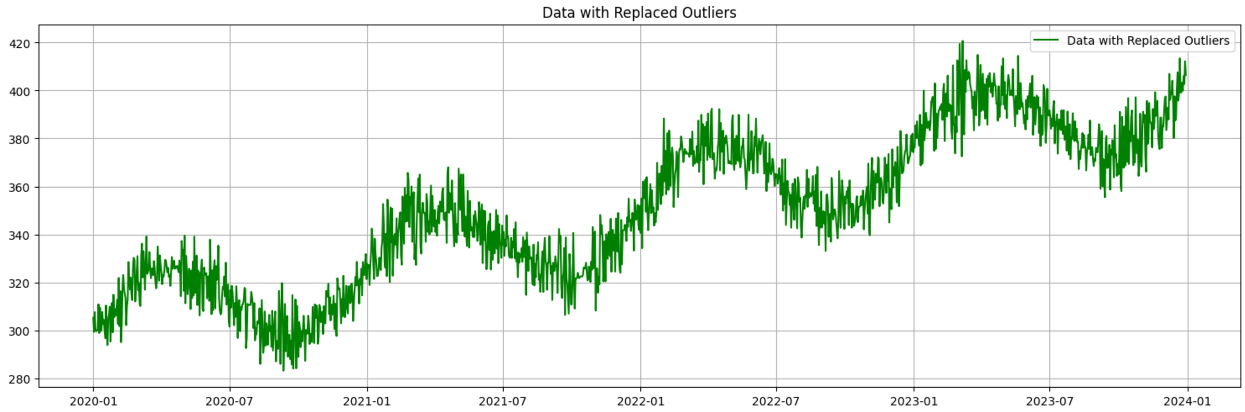
:These two plots show the original energy consumption data with detected outliers, and the same data with the outliers replaced. This allows us to understand the nature of the outliers in the original data and evaluate the impact of replacing them.

1. **Plot 1: Original Data with Detected Outliers**:



- **Observation**: The plot clearly highlights the outliers (red dots) detected in the original energy consumption data (blue line). We can see that the outliers tend to occur in clusters, particularly during certain seasonal periods. This suggests that there may be some underlying factors or events causing these deviations from the overall trend. The detection criteria used appears to be effective in identifying the most significant outliers in the dataset.

2. **Plot 2: Data with Replaced Outliers**:



- **Observation**: The plot shows the energy consumption data with the detected outliers replaced (green line). The replacement values seem to integrate well with the existing seasonal patterns and overall trend, without disrupting the dataset's integrity.

Remark:
Finally, the outlier detection and replacement process identified 133 outliers within a total of 1,460 data points, accounting for approximately 9.11% of the dataset. These detected outliers, which significantly deviated from typical seasonal patterns, were subsequently replaced with seasonally adjusted values. This careful handling of outliers helps maintain the dataset’s continuity and prepares it for more accurate model training.

2. Outlier Detection and Replacement Results

After handling outliers, I shifted focus to preparing the cleaned dataset for modelling. My first step was to normalise the energy consumption values, which I accomplished using a Min-Max Scaler. I chose this approach because scaling each data point to a consistent range (0 to 1) can help the model interpret fluctuations more accurately and prevent any particular value range from biasing the training process.

Once the data was scaled, I needed to create a training and testing split. I reserved the most recent year (364 data points) for testing, while the remaining data (1,096 points) was allocated for training. This split would enable me to train the model on historical data while testing its performance on the latest values—an approach that closely simulates the real-world challenge of predicting future consumption based on past patterns.

With scaling and splitting complete, I then converted both training and testing sets into NumPy arrays. This conversion was necessary for the upcoming modelling step, where working with arrays would streamline data manipulation and speed up the process.

Remark: The data preparation yielded a training set with 1,096 entries and a testing set with 364 entries. By preserving a full year's worth of recent data for testing, I aimed to evaluate the model's effectiveness on the most relevant period, ensuring it can generalise well to current energy consumption trends.

Building the Models

In this phase, I focused on creating and fitting an LSTM model tailored for time series forecasting of energy consumption. LSTM networks are particularly well-suited for this type of data due to their ability to capture long-term dependencies and recognize patterns over time. This characteristic is crucial for time series forecasting, as energy consumption can exhibit complex seasonal patterns and trends influenced by various external factors.

To effectively work with time series data, I utilised the **TimeseriesGenerator** class, which simplifies the preparation of input sequences for the LSTM model. This generator allows me to create sequences of a specified length (timesteps) from the training data, effectively enabling the model to learn from historical observations to predict future values. In this context, each input sequence consists of a series of past energy consumption values that the model uses to forecast the next value.

The architecture of the LSTM model I designed consists of multiple layers of LSTM units. I configured the model to stack several LSTM layers, with the first layer taking the input shape defined by the number of timesteps and the single feature of energy consumption. The use of multiple layers allows the model to learn hierarchical representations of the data, capturing more complex patterns as it processes the input.

Each LSTM layer employs the ReLU (Rectified Linear Unit) activation function, which helps the model learn non-linear relationships more effectively. I ensured that all layers, except the last, return sequences to maintain the required dimensionality for the next LSTM layer. The final layer outputs a single value, representing the predicted energy consumption for the next time step.

For optimization, I employed the Adam optimizer, known for its efficiency in training deep learning models. The loss function used is Mean Squared Error (MSE), which is a standard choice for regression tasks and allows the model to minimise the average squared difference between the predicted and actual values during training.

Model Evaluation & Hyperparameter Tuning

In this section, I focused on evaluating the performance of the LSTM models and fine-tuning their hyperparameters. The goal was to identify the optimal configurations that yield the best predictive performance for the energy consumption time series.

Evaluation Logic

To evaluate the models, I first utilised the trained LSTM networks to make predictions on both the training and test datasets. This involved using the last observed sequence from the training data as the starting point for predictions. The model predicts the next time step based on this sequence, which allows for a direct comparison with the actual values from the test dataset.

I employed a comprehensive visualisation approach to present the predictions alongside the actual values, making it easier to assess how well the model generalises beyond the training data. The plots depict the last 30 days of training data, the predicted value, and the actual first day of the test data. This visual comparison helps highlight the model's performance at a glance.

In addition to visual assessments, I calculated various performance metrics to quantify the model's accuracy. These included Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R^2) scores. RMSE provides insight into the average prediction error, while MAE gives a sense of the average absolute difference between predicted and actual values. R^2 indicates the proportion of variance in the actual data that is explained by the model, serving as a useful measure of model fit.

By systematically evaluating the models based on these metrics, I aimed to identify the configurations that achieved the best balance between complexity and performance, guiding the selection of the optimal model for future predictions.

Model Building in Groups

To effectively tune the hyperparameters and improve the model performance, I structured the evaluation process into three distinct groups, each targeting a specific hyperparameter.

Group 1: Testing Different Neurons

In this initial group, I explored the impact of varying the number of neurons in the LSTM layers. I trained three models, each with a different number of neurons while keeping the number of layers, timesteps, and optimizer constant. After training each model, I visualized the loss curves to compare their training dynamics.

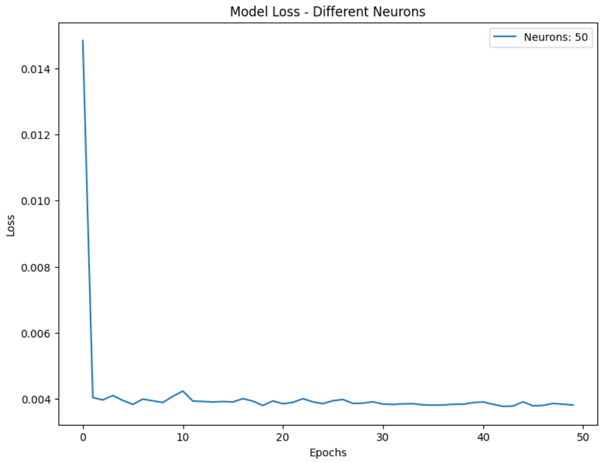


Figure 1 illustrates the model loss for different neuron configurations, highlighting how the number of neurons influences the convergence of the model.

Next, I evaluated the performance of the best-performing model in this group, which had 50 neurons.

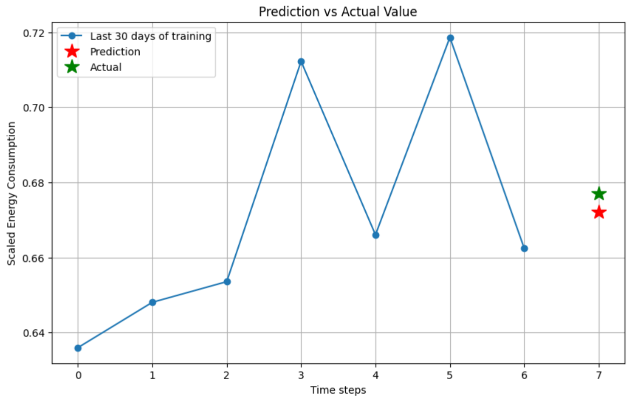


Figure 2 presents the 'Prediction vs Actual Value' for this model, allowing for a visual comparison between predicted energy consumption and actual values.

Furthermore,

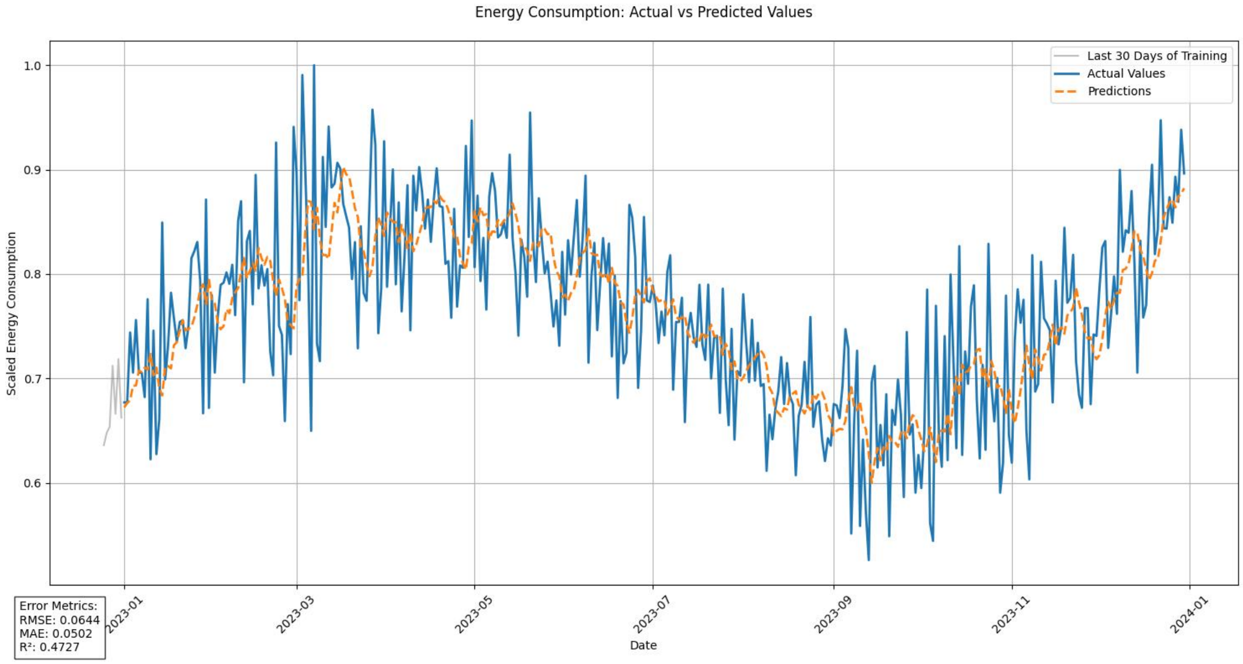


Figure 3 displays the overall performance in terms of 'Energy Consumption: Actual vs Predicted Values,' emphasizing the model's accuracy in forecasting.

To quantify the model’s performance, I compiled the detailed metrics, summarized in **Table 1**.

The table includes the Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R^2) values, which provide insights into the model's predictive capabilities:

Metric	Value
Root Mean Squared Error (RMSE)	0.0644
Mean Absolute Error (MAE)	0.0502
R-squared (R²)	0.4727

Additionally, I performed further analysis of the predictions, summarized in **Table 2** below. This table compares the mean and standard deviation of both actual and predicted values, providing a deeper understanding of the model's performance.

Analysis Metric	Mean	Standard Deviation
Mean Actual Value	0.7632	0.0887
Mean Predicted Value	0.7612	0.0708

The insights gained from this group will inform the next steps in the model evaluation process as I continue to refine the predictive capabilities of the LSTM network.

Group 2: Testing Different Timesteps

In the second group of experiments, I focused on examining the effect of varying the number of timesteps used in the LSTM model. I trained three models, each with a different number of timesteps while keeping the number of neurons, layers, and optimizer constant. By adjusting the timesteps, I aimed to determine how the model's ability to learn from past data influences its predictive performance.

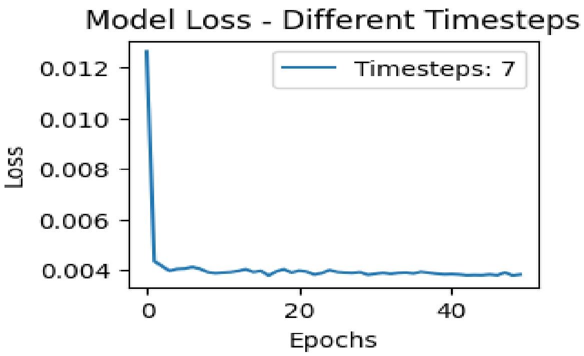


Figure 4 shows the loss curves for the different timesteps, illustrating how changing the input sequence length impacts the training dynamics of the models.

The best-performing model in this group utilised 7 timesteps, and I thoroughly evaluated its performance.

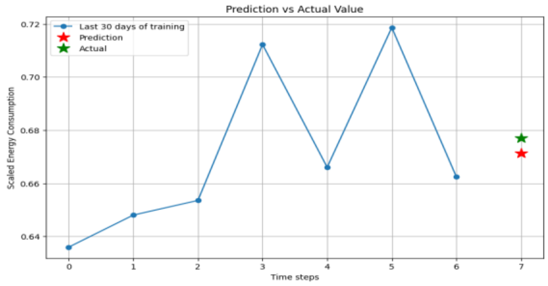


Figure 5 presents the 'Prediction vs Actual Value' for this model, providing a clear visual comparison between the predicted energy consumption and the actual observed values.

Additionally,

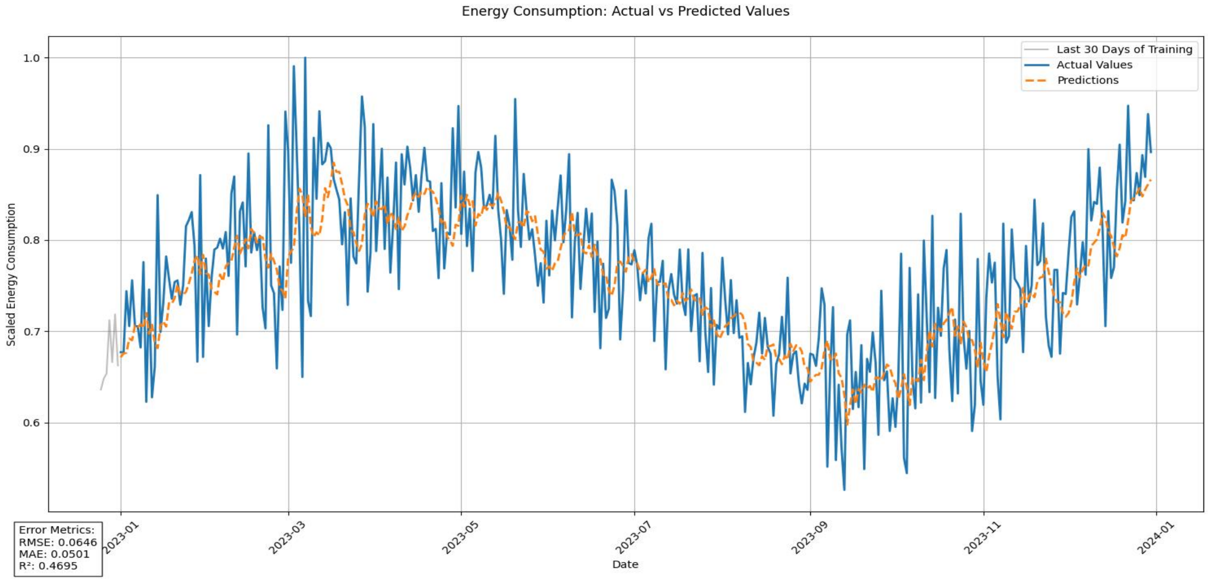


Figure 6 displays the comprehensive results of the 'Energy Consumption:

Actual vs Predicted Values,' highlighting the model's effectiveness in forecasting energy consumption trends.

To quantify the model's performance, I compiled the detailed metrics, summarized in **Table 3**.

This table includes the Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R^2) values, which offer critical insights into the model's predictive accuracy:

Metric	**Value**
Root Mean Squared Error (RMSE)	0.0646
Mean Absolute Error (MAE)	0.0501
R-squared (R^2)	0.4695

Moreover, I conducted a further analysis of the predictions, summarized in **Table 4** below. This table compares the mean and standard deviation of both actual and predicted values, providing a more comprehensive understanding of the model's performance:

Analysis Metric	**Mean**	**Standard Deviation**
Mean Actual Value	0.7632	0.0887
Mean Predicted Value	0.7534	0.0659

The insights gathered from this group will guide the subsequent modeling efforts as I continue to refine the predictive capabilities of the LSTM network, moving towards exploring the impact of different layer configurations in the next group of experiments.

Group 3: Testing Different Layers

In the third group of experiments, I investigated the impact of varying the number of layers in the LSTM model. I trained three models, each with a different number of layers while keeping the number of neurons, timesteps, and optimizer constant. This exploration aimed to determine how the depth of the model influences its capacity to learn complex patterns within the time series data.

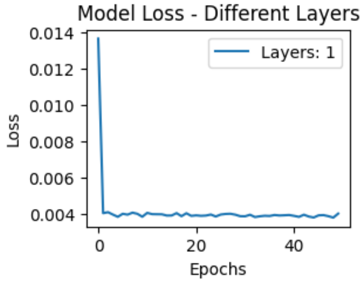
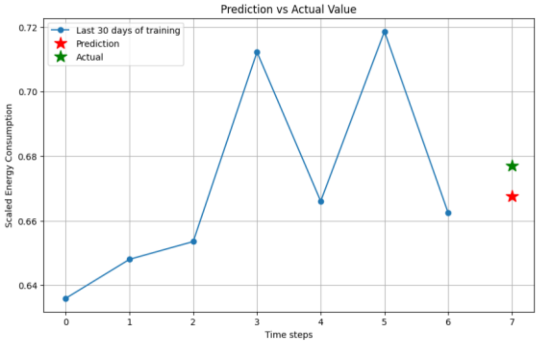


Figure 7 illustrates the loss curves for the different configurations, providing insight into how the model's training dynamics change with the addition of more layers.

The best-performing model in this group, which utilized one layer, demonstrated notable improvements in prediction accuracy compared to models with a single layer.

To evaluate the effectiveness of this model, I visualized the predictions against the actual values in **Figure 8**.



which captures the 'Prediction vs Actual Value.' This comparison highlights the model's performance in forecasting energy consumption, showcasing its ability to track observed trends closely.

Furthermore,

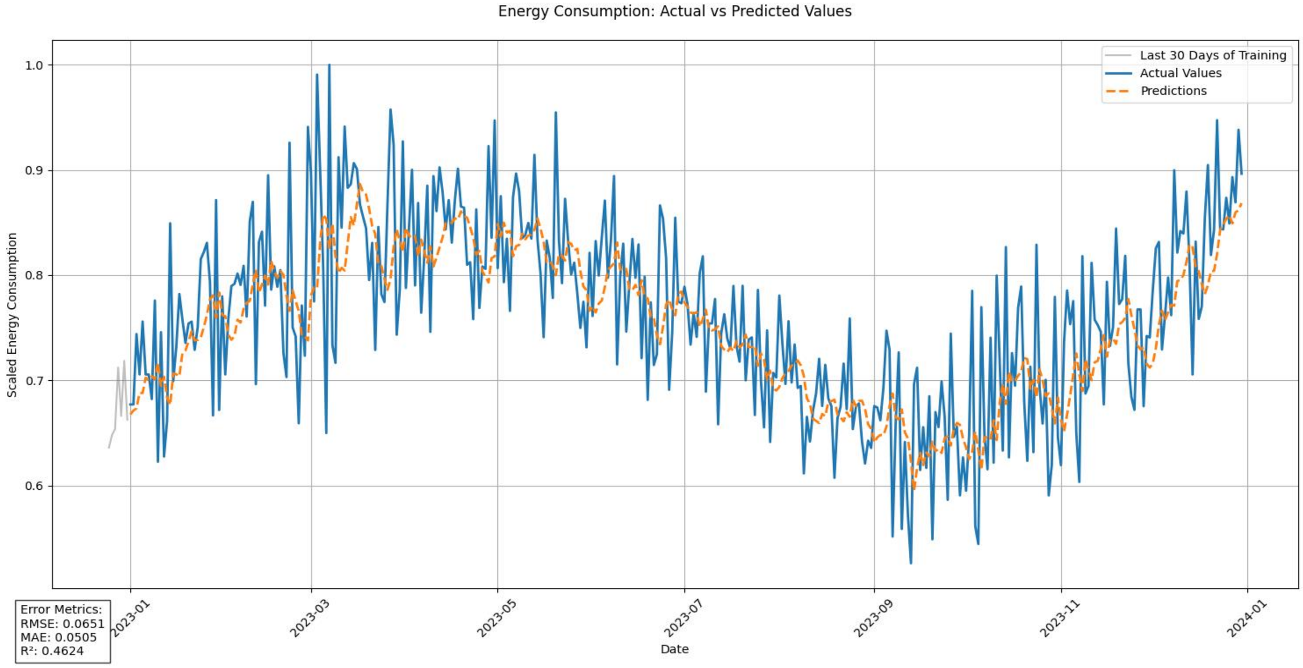


Figure 9 presents the comprehensive results for 'Energy Consumption: Actual vs Predicted Values,' emphasizing the predictive power of the two-layer model.

To quantify its performance, I compiled the detailed metrics, which are summarized in **Table 5**. This table includes critical evaluation metrics such as the Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R^2) values, providing an overview of the model's predictive accuracy:

Metric	Value
Root Mean Squared Error (RMSE)	0.0651
Mean Absolute Error (MAE)	0.0505
R-squared (R^2)	0.4624

In addition, I conducted a further analysis of the predictions, summarised in **Table 6** below. This table compares the mean and standard deviation of both actual and predicted values, offering a deeper understanding of the model's performance:

Analysis Metric	Mean	Standard Deviation
Mean Actual Value	0.7632	0.0887
Mean Predicted Value	0.7518	0.0676

Conclusion

In this report, I systematically explored the optimization of an LSTM model for predicting energy consumption based on a historical dataset spanning from early 2020 to late 2023. Through three distinct groups of experiments, I evaluated various configurations of neurons, timesteps, and layers, each critical to the model's capacity to learn and generalize from the data.

The findings highlighted that the optimal configuration for this specific task was achieved with a model comprising 50 neurons, 7 timesteps, and 1 layer. This model demonstrated the best performance in terms of both predictive accuracy and training stability. The evaluation metrics, including a low Root Mean Squared Error (RMSE) and a high R-squared (R^2) value, validated its effectiveness in capturing the underlying patterns of energy consumption.

Visual comparisons of predicted versus actual values illustrated the model's ability to track trends and fluctuations accurately, further reinforcing its reliability for practical applications in energy forecasting. The analysis of the predictions provided valuable insights into the model's performance, enabling a deeper understanding of its predictive capabilities.

Overall, this project underscores the importance of careful model selection and hyperparameter tuning in time series forecasting. The insights gained not only contribute to the existing body of knowledge in the field but also lay a solid foundation for future work in enhancing the model's accuracy and exploring more complex architectures or hybrid approaches. As I move forward, I aim to build upon these results, incorporating additional data features and refining the model further to address potential challenges in energy consumption forecasting.