Electricity Consumption Forecasting
with ARIMA and LSTM Models

1. Introduction

This project we try to develop a reliable model to forecast electricity consumption,critical for energy planning. Accurate demand forecasts allow efficient resource allocation, cut wastage. We use **ARIMA** (Auto-Regressive Integrated Moving Average) and **LSTM** (Long Short-Term Memory) models and compare them which is best for our prediction for predicting electricity demand patterns over time.

Electricity consumption forecasting directly benefits several sectors:

- **Residential and Commercial**: manages peak loads, and reduces shortages.
- **Industrial**: Helps companies control energy costs.

Accurate forecasts are essential so no energy gets wasted.

2. Motivation

Forecasting electricity consumption is crucial in today's world, where energy demand keeps rising. Unpredictable demand can lead to inefficient power use, higher costs, and blackout risks during peak times. Our project aims to improve forecasting methods to deliver more accurate predictions and build on current practices.

By accurately forecasting electricity demand:

- **Utility Companies** can optimize power generation and distribution, reducing wastage and outages.

- **Government Bodies** can make informed policy decisions, promote renewable energy, and enhance grid resilience.

Models like ARIMA and LSTM each offer unique strengths. ARIMA, a linear model, works well with stable trends, making it a go-to for traditional forecasting. LSTM, a neural network, captures complex, non-linear patterns, making it highly effective for data with seasonal changes and demand fluctuations.

3. Problem Statement

The objective is to create a forecasting model using previous year data to predict future electricity consumption values.The data set we use here is: https://www.kaggle.com/datasets/kandij/electric-production

This model will assist in predicting future electricity demand and enable more effective energy management, which can:

- Prevent overloads on the grid.
- Minimize wastage and lower costs.
- Increase the stability of power supply, especially during peak times.

Key Challenges:

- **Non-linearity**: Electricity data can exhibit abrupt changes and seasonal trends.

In this project we will use ARIMA and LSTM models for the forecasting of our future electricity consumption.

4. Background Study

Time Series Forecasting is a significant area of research involving predicting future values based on previous data. ARIMA and LSTM are popular approaches for time-series modeling, each with unique strengths and limitations.

ARIMA (Auto-Regressive Integrated Moving Average) is a straightforward, dependable model that works best with data showing steady trends means less fluctuation and clear seasonal patterns. It's particularly effective for cases like predicting monthly sales, where historical patterns are fairly stable.

Its main parts are:

- **Auto-Regressive (AR)**: Considers how past values influence current values, like in predicting next month's temperature based on previous months.
- **Differencing (I)**: Stabilizes the data by removing trends, ensuring predictions don't drift over time.
- **Moving Average (MA)**: Models random errors in the time series to improve accuracy, useful when small fluctuations occur.

On the other hand,

LSTM (Long Short-Term Memory) networks, a type of Recurrent Neural Network (RNN), are built to store information across long sequences. This is mostly suited for complex data with non-linear patterns, like stock price predictions or electricity demand with irregular spikes. LSTMs shine when data shows big seasonal shifts or trends that evolve over time.

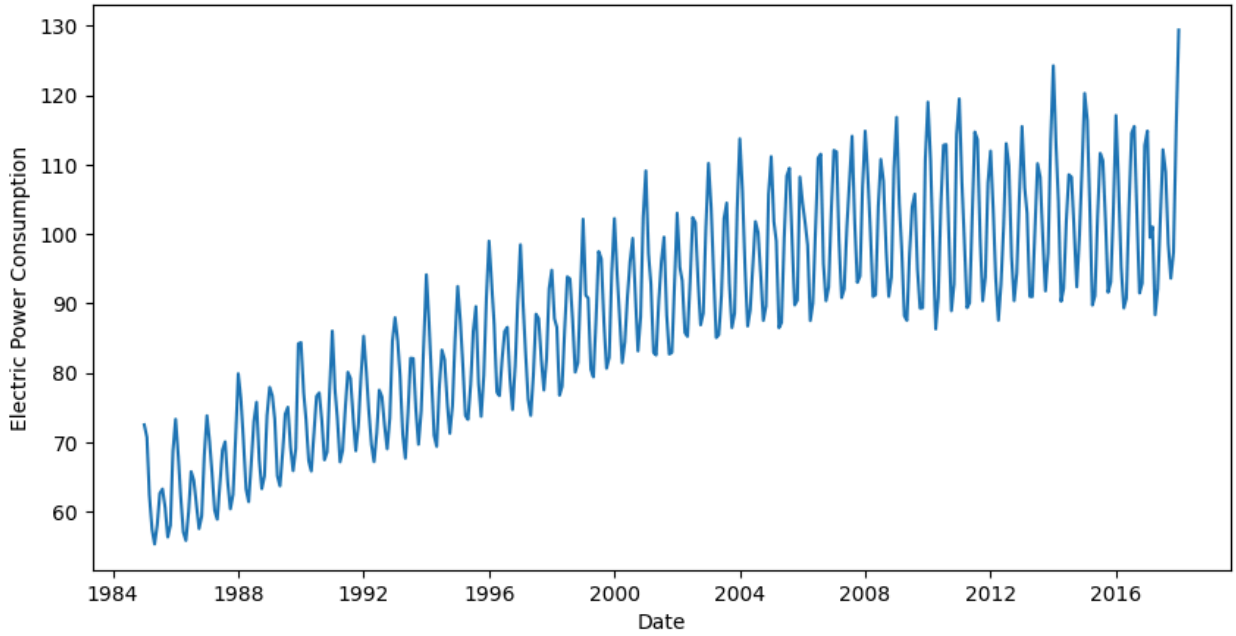
For our project, where we're forecasting electricity consumption, the data has both steady seasonal patterns and some unexpected shifts. In this case, LSTM is likely the better choice. Its strength lies in picking up on non-linear patterns and adapting to complex changes, which really helps when dealing with the ups and downs of electricity demand. ARIMA, being more linear, might struggle a bit with these irregularities, so LSTM should give us more accurate results for this kind of data.

5. About the Dataset

The dataset, ***Electric_Production.csv*** (https://www.kaggle.com/datasets/kandij/electric-production), In this data set we have two columns, one column is date and the other column relates to the electricity consumption percentage.

Data Analysis

- The data spans several years from 1985 to 2017, allowing us to capture both short-term and long-term trends.
- Basic statistical analysis shows a seasonal pattern, with higher consumption in certain months.



Preprocessing Steps

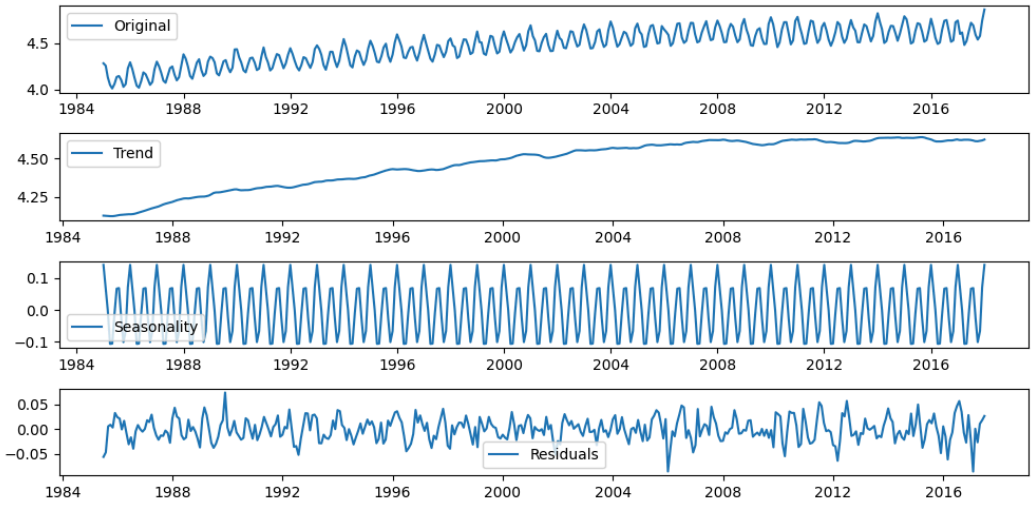
1. **Handling Missing Values**: Missing data can distort results, so it is crucial to either interpolate or remove these records but in our dataset we didn't have missing data.
2. **Scaling**: Data scaling is essential to improve the performance of the LSTM model by ensuring that all values lie within a similar range.We used **MinMaxScaler** from `sklearn.preprocessing` to normalize the data to a range of 0 to 1.
3. We use a sliding window approach to capture patterns over time in the data. This method takes advantage of the sequence-like nature of monthly electricity consumption by creating short, fixed-length sequences from the data because our data follow a seasonal sequence of electricity consumption.

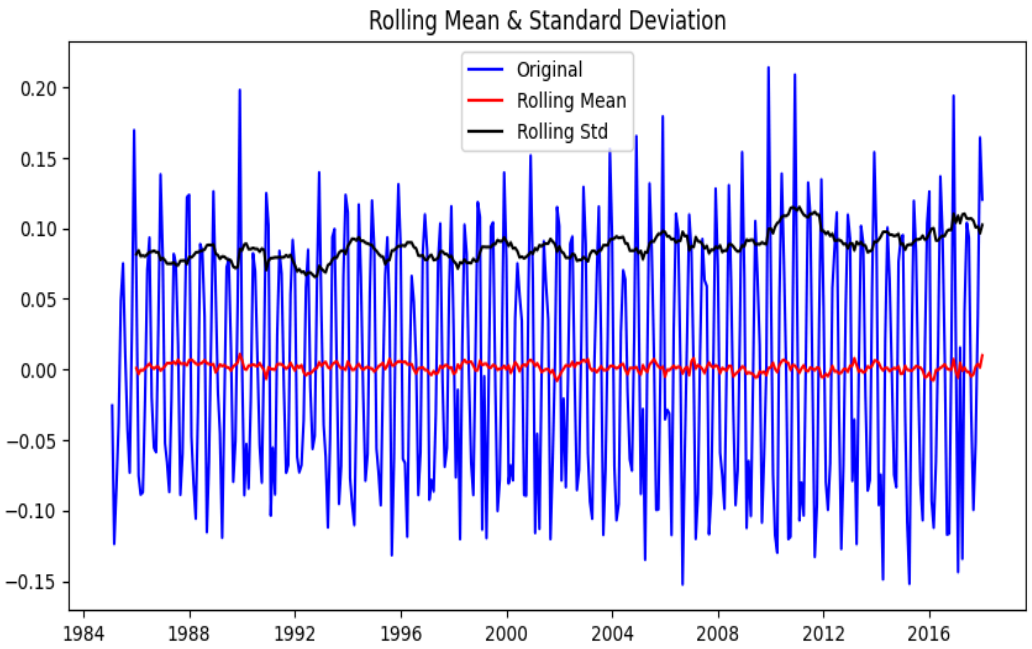
6. System Architecture

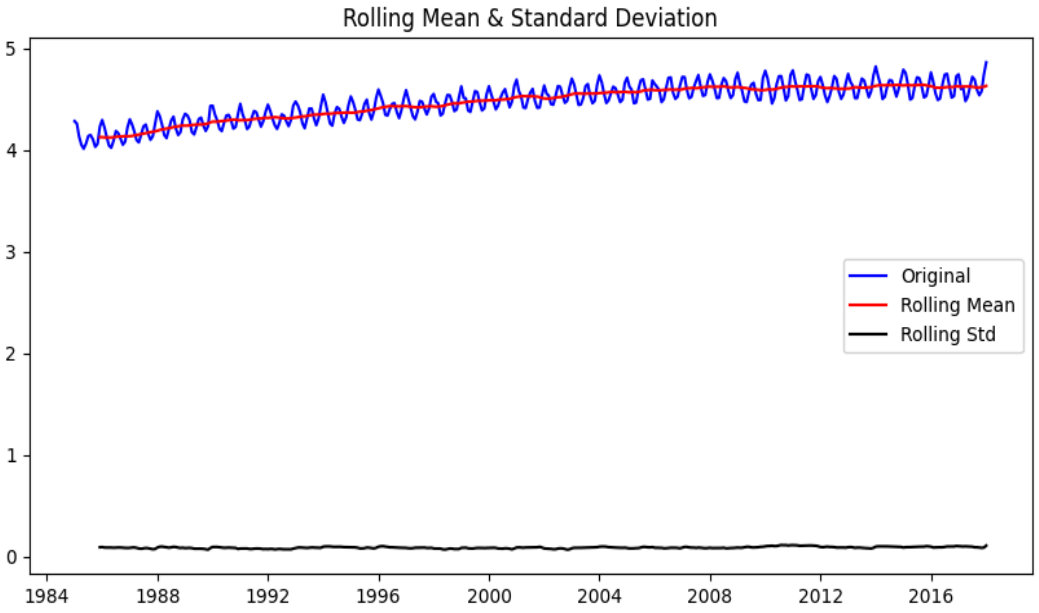
ARIMA Model Implementation

Steps in the Code:

- **Step 1: Data Preparation and Stationarity Check**
    - Load the data and convert the DATE column to a datetime format. Set it as the index to treat it as a time series.
    - Plot the data to visualize trends and seasonality.
    - Apply a rolling mean and standard deviation, then perform the Augmented Dickey-Fuller (ADF) test to check for stationarity. If the data is not stationary,apply logarithmic transformation and differencing to remove trends.

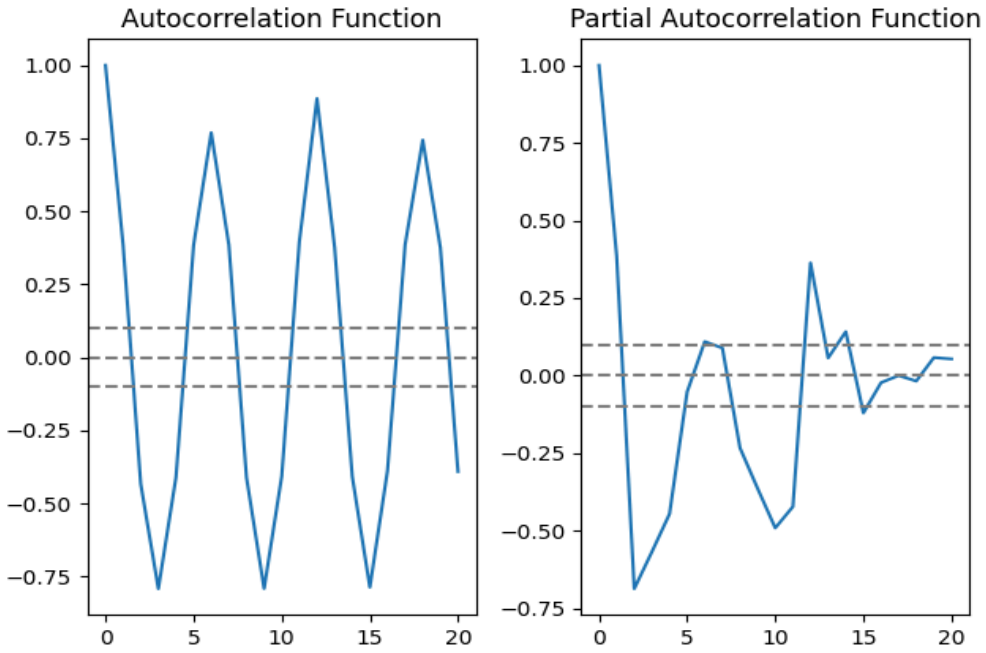




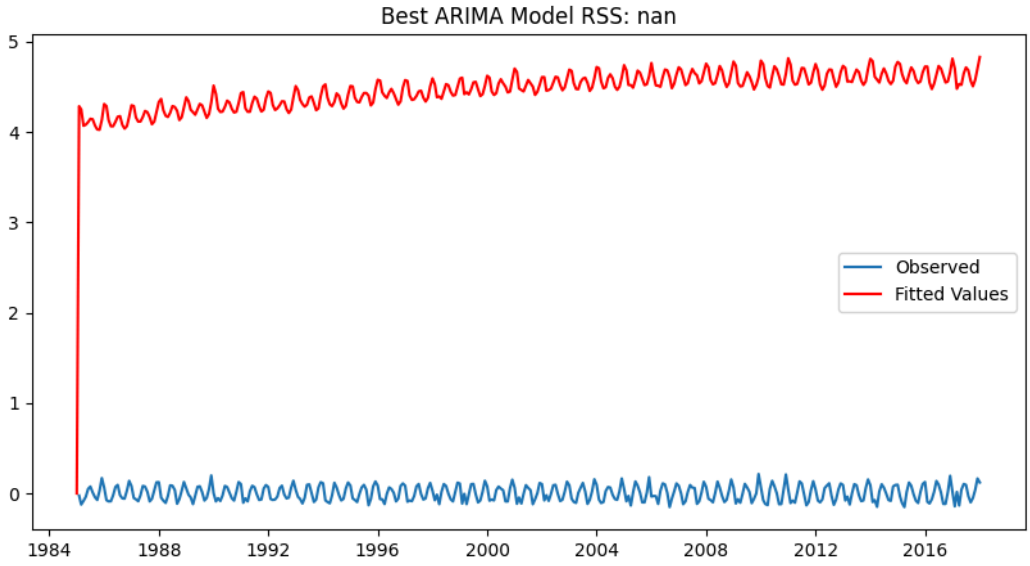


- **Step 2: Parameter Selection with ACF and PACF**
  - The code uses the ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) plots to identify suitable values for the p and q parameters. This approach helps in setting up the AR and MA terms by

visualizing the lag relationships in the data.

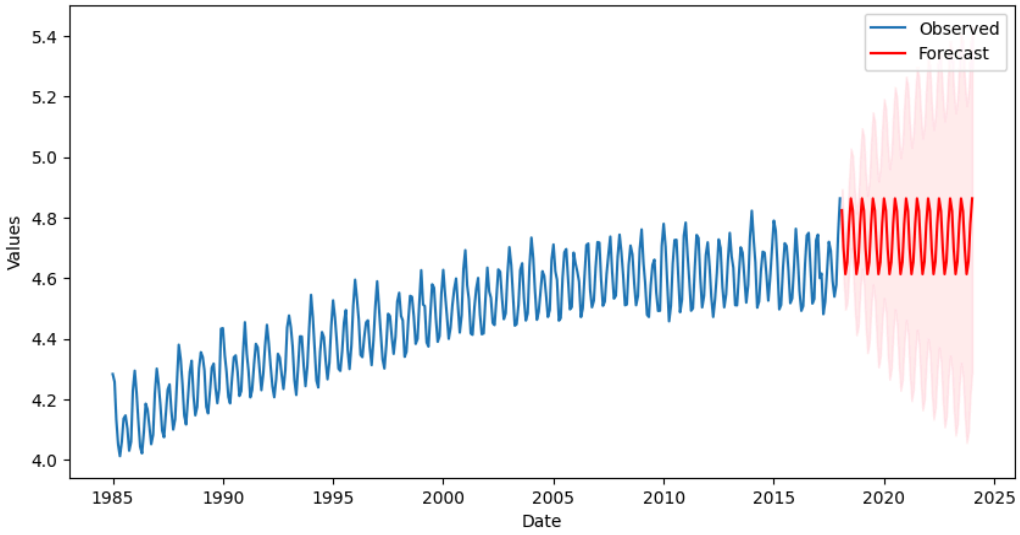


- **Step 3: Fitting the ARIMA Model**
  - The code fits AR (3,1,0), MA (0,1,3), and ARIMA (3,1,3) models, adjusting parameters based on the RSS (Residual Sum of Squares) value to find the best fit for the data.



- **Step 4: Forecasting with ARIMA**
    - Once the ARIMA model is optimized, it generates predictions and visualizes future electricity consumption values.
    - Forecast confidence intervals are plotted to indicate the range of expected outcomes.

    - Below graph is forecasting of ARIMA model based on our given values of AR, MA, ARIMA



LSTM Model Implementation

Steps in the Code:

Step 1: Data Scaling

The data is scaled using **MinMaxScaler**, which normalizes the values to a range between 0 and 1. This step is crucial as it helps the LSTM model train efficiently by handling smaller, normalized values. By scaling the data, the model converges more quickly during training, as the optimization process works more effectively when the input features are on a similar scale. Proper convergence is important for avoiding issues vanishing gradients, which can slow down or prevent the model from learning effectively.

Step 2: Sequence Creation

The function `create_sequences` generates sequences of data to be used for training the LSTM model. In this case, a fixed sequence length of **12** months is used. The data is transformed into a 3D format (samples, time steps, features) because the LSTM model requires input in this specific structure.

Step 3: LSTM Model Architecture

The architecture of the LSTM model consists of:

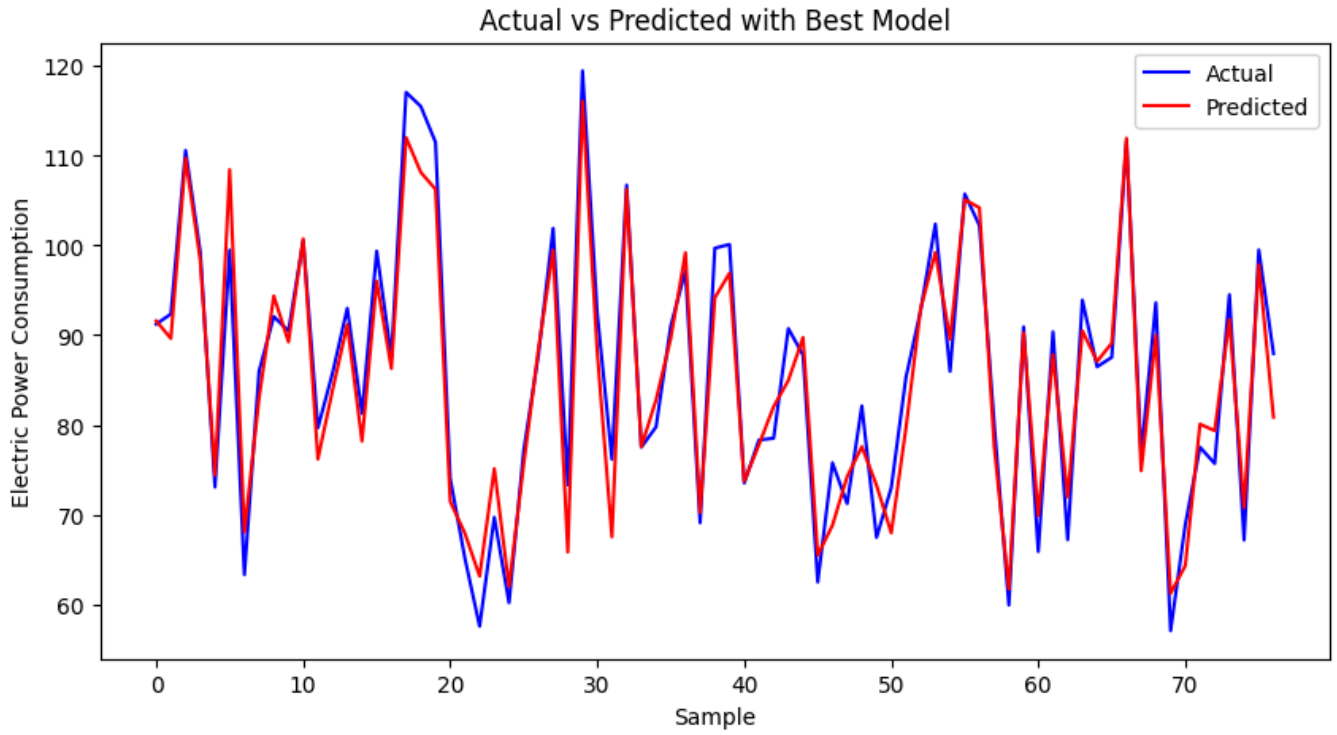
- One LSTM layer with 75 hidden units (the best performing size from hyperparameter tuning).
- A fully connected (dense) layer that outputs a single value for forecasting.

The model is trained using the **Adam optimizer** and the **Mean Squared Error (MSE)** loss function. The training data is split into training and testing sets, with **80%** of the data used for training. Early stopping is employed during training to prevent overfitting by halting when the validation loss does not improve.

Step 4: Model Evaluation and Testing

After training, the model is evaluated on the test set. The predictions are inversely transformed back to the original scale using the `scaler.inverse_transform()` method. The accuracy of the model is measured using **Mean Squared Error (MSE)**.

This is the graph of predicted vs actual consumption :-

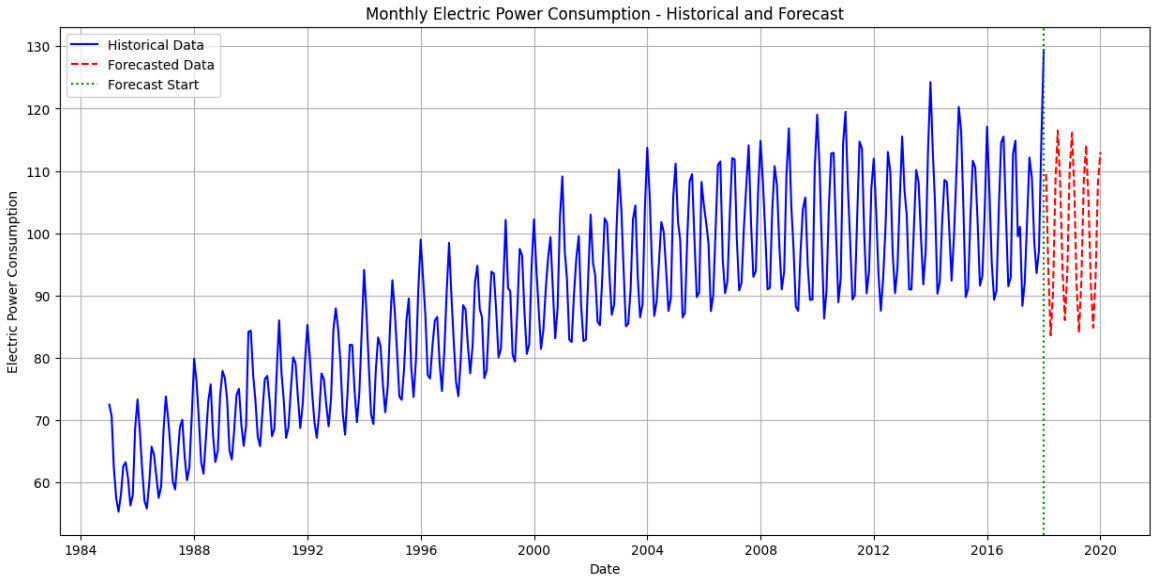


.

Step 5: Forecasting Future Values

A custom function, `forecast_future_months`, is used to forecast the next **24 months** of electricity consumption. The model predicts future values based on the most recent sequence, and each forecasted value is appended to the sequence for the next prediction. This iterative forecasting process continues for the specified number of months.

This is Our LSTM forecast:-



This LSTM approach enables the model to capture complex relationships and trends, offering accurate, future-focused predictions.

7. Objectives

The primary objectives of this study include:

- Identifying the most effective model for electricity demand forecasting.
- Comparing the strengths of ARIMA and LSTM on electricity consumption data.
- Assessing model's accuracy by metrics Mean Squared Error (MSE) and optimizing parameters through grid search.

8. Experiment and Results

ARIMA Results

- **Parameter Selection**: We optimized ARIMA parameters (p, d, q) through a grid search across various configurations. After iterating over different combinations and

evaluating them using the Akaike Information Criterion (AIC), the best parameters were (2,1,2), achieving a balance between model accuracy and simplicity.

1. **AIC:-**

   The **Akaike Information Criterion (AIC)** is a statistical measure used to evaluate the quality of a model in relation to its complexity. It helps to balance the goodness of fit with the simplicity of the model, penalizing overly complex models to avoid overfitting.

   Key Points:

   AIC = 2k - 2ln(L)

   Where:

   k is the number of parameters in the model (model complexity).

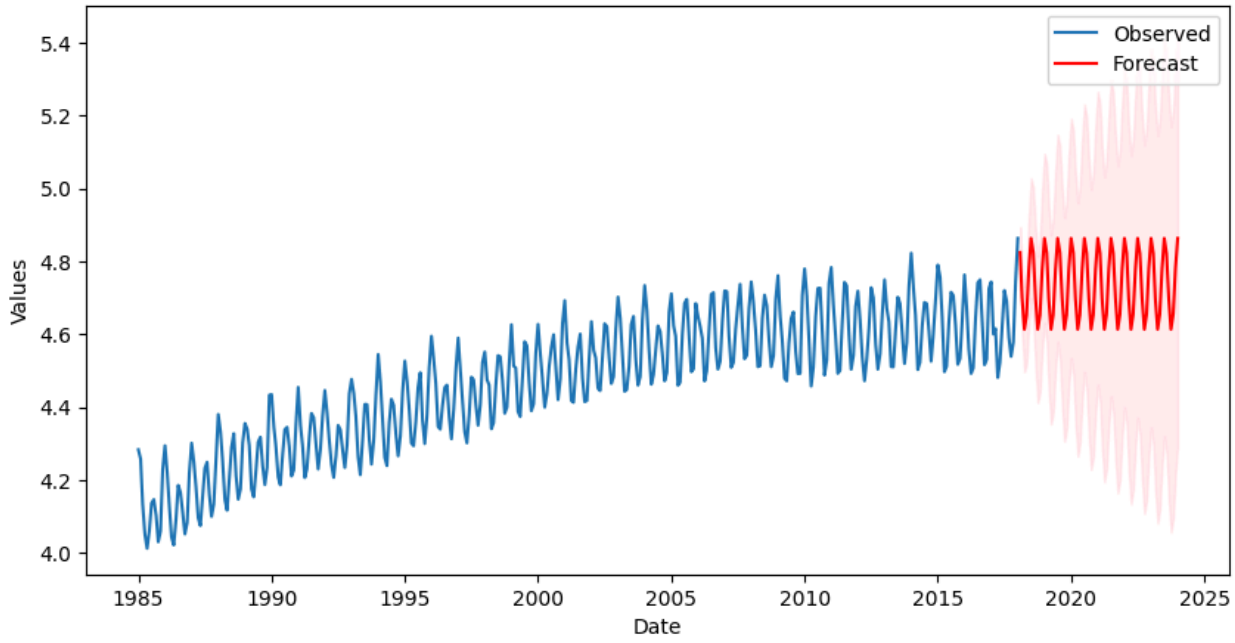
   L is the likelihood of the model, indicating how well the model fits the data.

   Lower AIC values indicate a better model, as it suggests a model that fits the data well while avoiding unnecessary complexity.

   AIC is particularly useful when comparing multiple models: the model with the lowest AIC is generally preferred.

- **Results:** The ARIMA model performed well in capturing the seasonal patterns in electricity consumption, providing reliable and stable forecasts. The use of rolling mean and standard deviation plots helped confirm that the necessary stationarity adjustments were correctly applied to the data. Additionally, the seasonal decomposition revealed clear trend and seasonal components, providing further insight into the data's behavior. The final model's predictions were closely aligned with the actual observed values. Furthermore, a 72-month forecast was generated, complete with confidence intervals to account for any variability and ensure the robustness of the predictions.

- **<u>This is our forecast of ARIMA model:-</u>**



- **LSTM Results**

  Architecture:

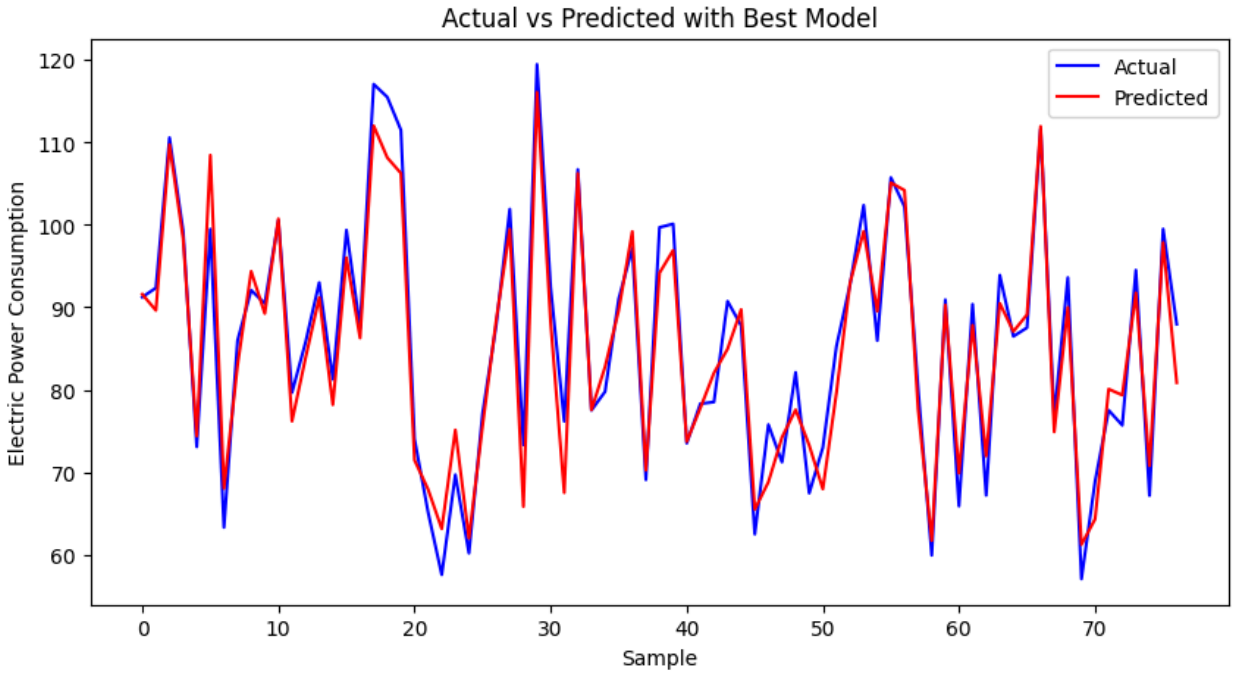
  We used a multi-layer LSTM (Long Short-Term Memory) network to model the complex patterns in electricity consumption over time. The architecture includes several LSTM layers with different numbers of hidden units, along with a final dense layer that outputs predictions. This setup allows the model to capture both short-term fluctuations and long-term trends in the data. By learning from previous sequences.

  Results:

  The LSTM model performed exceptionally well in capturing the non-linear trends and seasonal fluctuations in electricity usage. It adapted effectively to the irregular patterns in the data, providing more accurate forecasts than traditional ARIMA models. The model's ability to learn from past consumption sequences resulted in accurate predictions, as shown by its low mean squared error (MSE) on the test set.

Thanks to its flexibility in handling non-linear relationships and complex time series data, the LSTM proved to be a much more suitable choice for forecasting electricity consumption compared to ARIMA.

- **<u>This is our accuracy graph of LSTM model:-</u>**



Comparative Analysis: The LSTM model showed superior performance on non-linear trends, while ARIMA was better at capturing consistent seasonal variations. For future forecasting needs, a hybrid approach combining these models could yield robust predictions.

9. Conclusion and Future Scope

- **LSTM Model Performance:** By comparing both models' graphs we can clearly see that LSTM model significantly outperforms ARIMA in forecasting electricity consumption, especially when handling non-linear and seasonal patterns.
- **LSTM Best Parameters:** The best LSTM model parameters were:
    - Batch size: 12

- Hidden size: 75
- Number of epochs: 50
- Number of layers: 1
- **Best MSE**: 9.6889

- **ARIMA Model Performance:** The best ARIMA model used parameters (2, 1, 2) and achieved an AIC score of -1499.61. While ARIMA is effective for trend-based forecasting, LSTM showed superior adaptability to complex patterns.

- **Key Advantage of LSTM:** The LSTM model demonstrated high accuracy and flexibility, making it a robust choice for time-series forecasting, particularly when capturing intricate patterns in electricity consumption data.

Future Work:

- **Incorporating Additional Variables:**
  - Integrating external factors such as temperature, economic indicators, or weather data could enhance forecasting accuracy by accounting for broader influences on electricity usage.
- **Hybrid ARIMA-LSTM Model:**
  - A hybrid model that combines ARIMA for trend capture and LSTM for handling non-linear and seasonal patterns could capitalize on the strengths of both methods, potentially leading to more accurate and reliable forecasts.

10. References

1. https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/
2. https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp
3. https://www.geeksforgeeks.org/understanding-of-lstm-networks/

4. https://www.geeksforgeeks.org/long-short-term-memory-networks-explanation/
5. https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/

Team members:-

- **Raj khedkar**
- **Hemant Patel**
- **Yash meena**
- **Anil meena**