MANIPAL UNIVERSITY
JAIPUR

AI BASED ELECTRICITY DEMAND FORECASTING FOR DELHI

Minor Project I (DS3170)

Presented By

Aashi Sharma
Reg. No. 229309087

Kaustubh Ghale
Reg. No. 229309086

Under the Guidance of

Dr. Neeraj Kumar Verma

Dr. Shweta Redkar

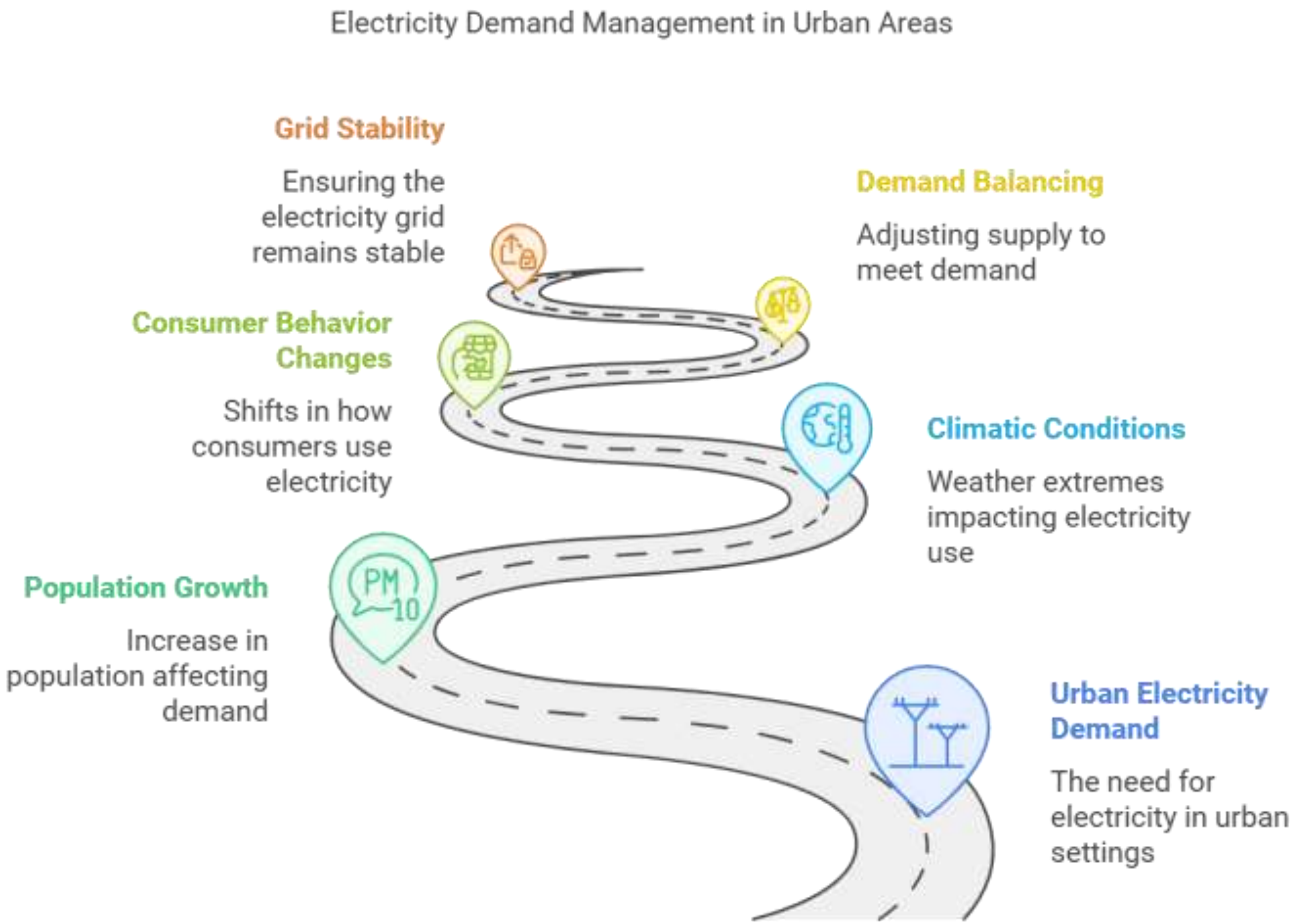
Department of Data Science & Engineering

School of Information, Security and Data Science

Manipal University Jaipur, Jaipur

Introduction to Project

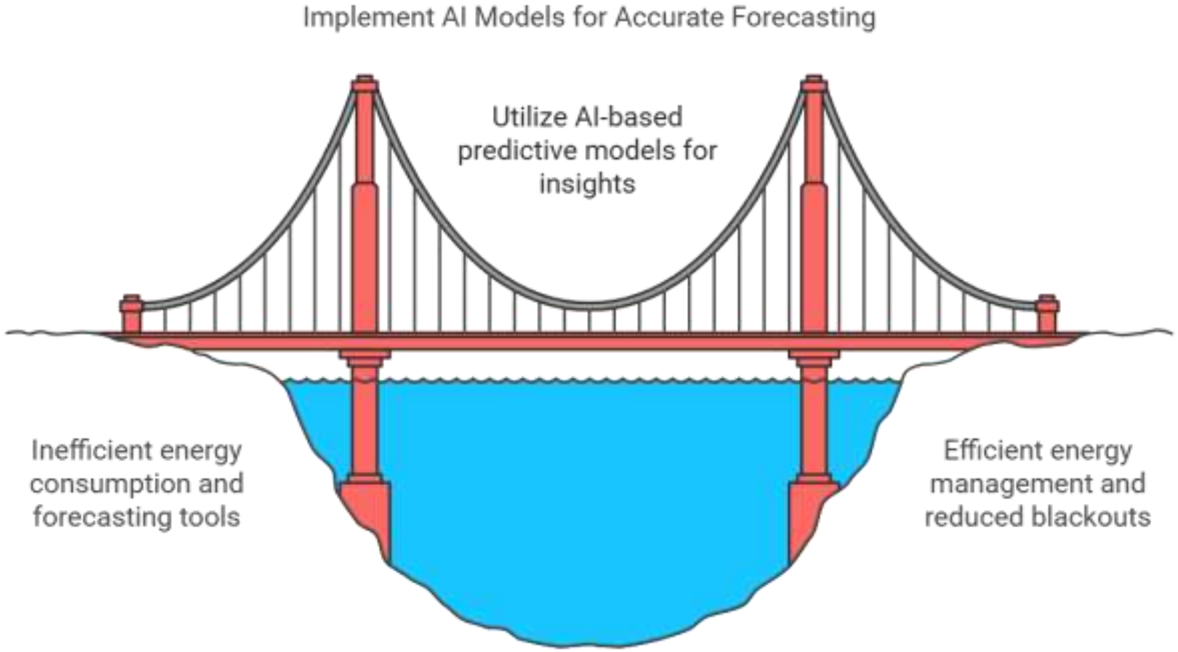
- Electricity forms the mainstay of modern urban living and the operation of industries, businesses, and houses. For metropolitans like Delhi, population growth, extreme climatic conditions, and changes in consumer behavior continue to influence electricity demand fluctuations. Demand balancing at each point and assuring it is efficient is a critical role for maintaining the stability of the grid, reducing energy wastage, and integrating renewable sources of energy.



MANIPAL UNIVERSITY JAIPUR

Problem Identification

- India's capital city, Delhi, experiences extreme weather conditions, high growth rates of urbanization, and a considerable base of household and commercial customers. Traditional forecasting tools fail to capture intricate connections among numerous factors influencing electricity demand. Therefore, such forecasting tools make energy consumption inefficient and costly, along with an increased risk of blackout. Thus, AI-based predictive models can help overcome these issues by presenting accurate data-driven insights into electricity consumption patterns.

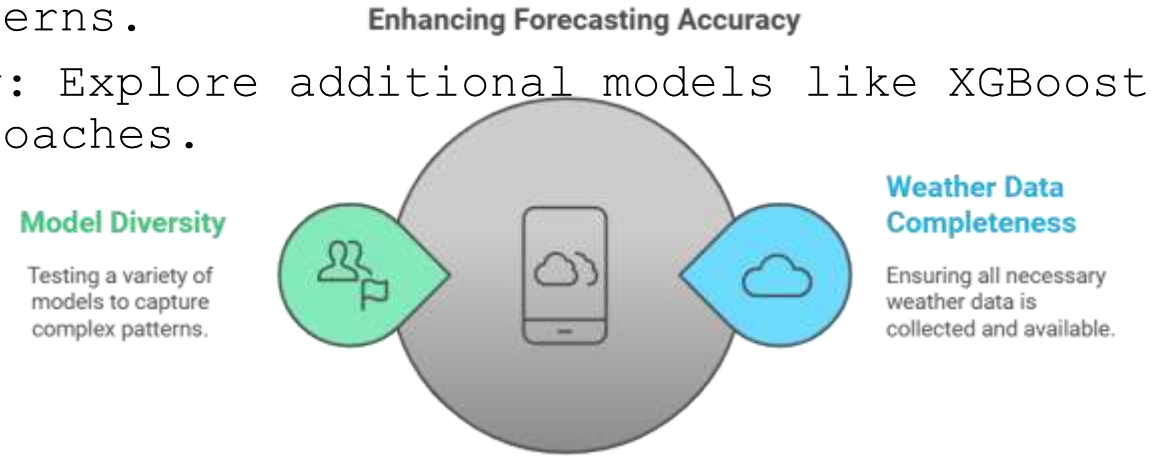


Literature Review

- "Electricity demand forecasting: A review of models and methods"Authors: H. Zhang, X. Li, and M. WuThis paper reviews various forecasting models and methods used in electricity demand prediction.
- "Short-term electricity demand forecasting with recurrent neural networks"Authors: J. W. Taylor and P. J. L. McSharryThis paper explores the use of recurrent neural networks (RNNs) for short-term electricity demand forecasting.
- "A hybrid model for short-term electricity load forecasting using ensemble learning"Authors: A. Chien, M. K. Wang, and P. C. ChangThis paper presents a hybrid approach combining various ensemble learning techniques for load forecasting.
- "Integrating weather data with machine learning for improved electricity demand forecasting"Authors: J. Liu, X. Zhou, and Y. ZhangThis research focuses on enhancing demand forecasting by integrating weather data into machine learning models.
- "Demand forecasting for smart grids: A deep learning approach"Authors: K. Zhao, X. Li, and Q. WuThe paper explores deep learning methods for forecasting demand in smart grids, including techniques that could be useful for your project.

Gap Analysis

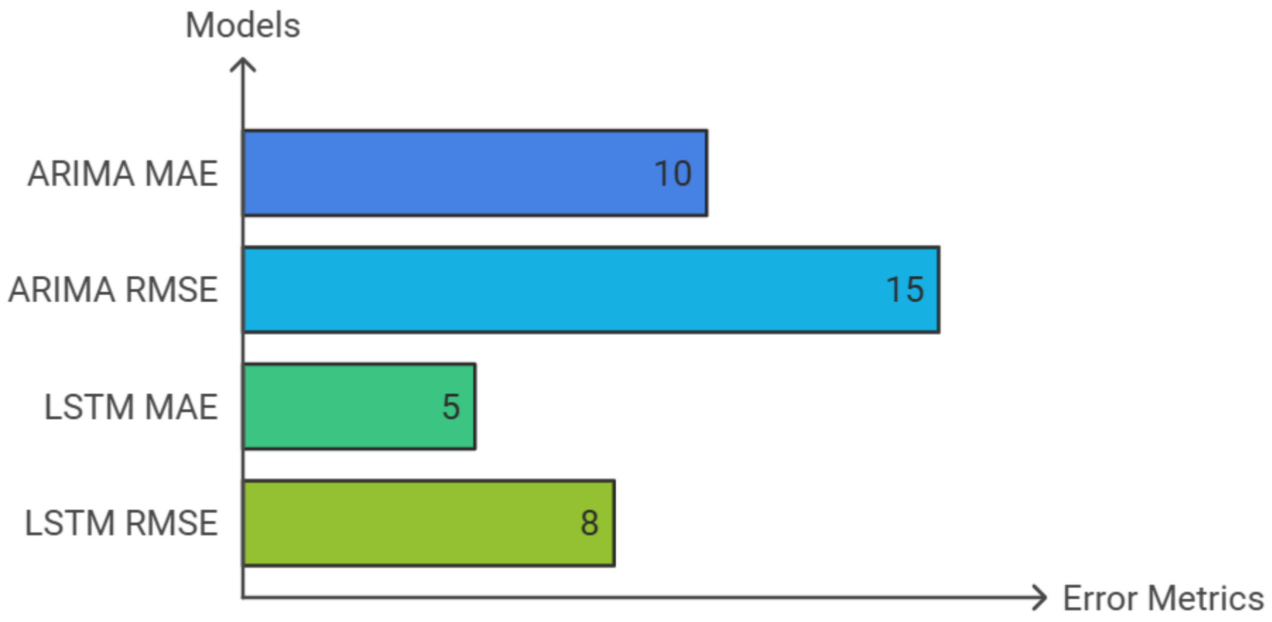
- Data Gaps:
  - Issue: Incomplete weather data (e.g., missing temperature or humidity values).
  - Impact: Missing weather data can reduce the accuracy of forecasting models.
  - Opportunity: Improve data collection by integrating more weather stations and enhancing data accuracy.
- Model Gaps:
  - Issue: Limited models tested (RF, LSTM, ARIMA).
  - Impact: Models may not fully capture all aspects of complex power demand patterns.
  - Opportunity: Explore additional models like XGBoost, Prophet, or hybrid approaches.



Objectives

To the following will be achieved with this project:

1. Predict the amount of electricity required in Delhi and, at what time of the day it will be consumed with maximum usage.
2. Take into consideration weather conditions, holidays, etc. to enhance precision of the model.
3. Compare statistical models, for instance ARIMA against the AI model, in this case LSTM.
4. Performance of the model should be evaluated in terms of Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).
5. Solutions suggested can be scaled to every other metropolitan area to effectively manage energy resource.



Performance Comparison of Electricity Prediction Models

Tools/Technologies Used

Hardware Requirements:

- The Processor should ideally be the Intel Xeon or equivalent.
- Graphics Card: NVIDIA Tesla T4 GPU (for training LSTM).
- RAM: 32 GB minimum.
- Storage: 512 GB SSD for quick data access.

Software Requirements:

- Programming Language: Python 3.x.
- Libraries and Frameworks:
- TensorFlow and Keras for deep learning models.
- Scikit-learn for Random Forest implementation.
- Statsmodels for ARIMA model.
- NumPy, Pandas for data manipulation.
- Matplotlib, Seaborn for visualization.
- IDE/Notebook: Jupyter Notebook.
- Version Control: Git/GitHub for code collaboration.

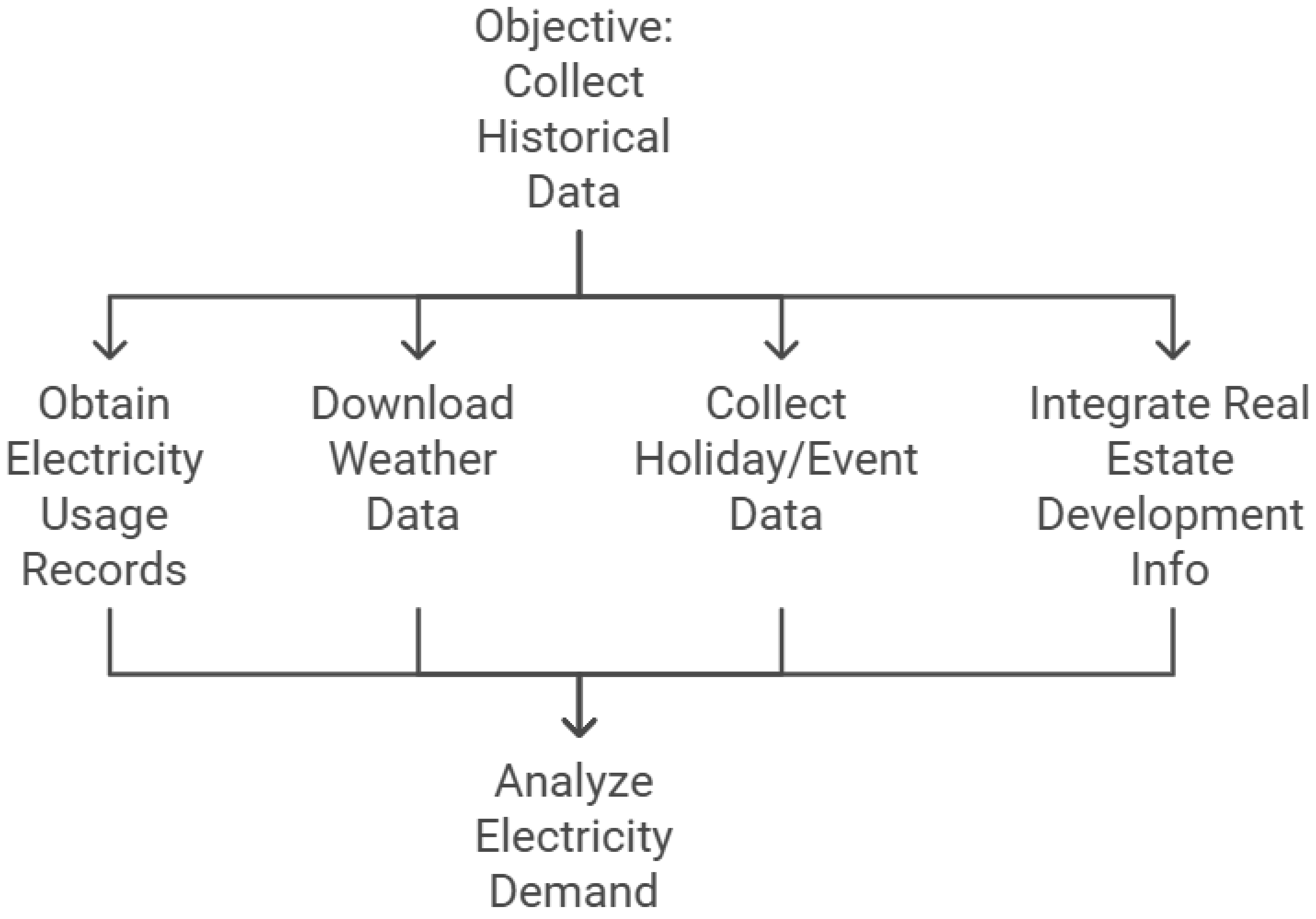
Methodology

1. Data Collection:

Objective: Collect historical demand, weather, and holiday/event data.

Steps:

1. Obtain electricity usage records from utility providers.
2. Download weather data (temperature, humidity, precipitation) from meteorological databases.
3. Collect holiday calendars and event schedules influencing energy use.
4. Integrate real estate development information with the load growth to be recovered.



date	time	season	temperature	humidity	solar	wind speed	delhi	holiday
01-01-2023	00:00:00	winter	13.90173	65.5	0	7.553054	5980.068	0
01-01-2023	01:00:00	winter	11.34082	47.4381	0	2.177358	5076.498	0
01-01-2023	02:00:00	winter	13.37137	64.06894	0	9.131008	6267.645	0
01-01-2023	03:00:00	winter	14.26009	54.95618	0	2.41533	6525.088	0
01-01-2023	04:00:00	winter	10.15637	59.4839	0	6.779265	5910.683	0

Table.1: Training dataset(final.csv)

Table.2: Prediction dataset(gani.csv)

date	time	season	temperature	humidity	solar	wind speed	holiday
02-01-2024	00:00:00	winter	15.38767	51.28926	0	4.029929	0
02-01-2024	01:00:00	winter	19.67812	64.79843	0	2.557387	0
02-01-2024	02:00:00	winter	15.50505	49.51597	0	5.094352	0
02-01-2024	03:00:00	winter	19.60526	66.42909	0	7.777601	0
02-01-2024	04:00:00	winter	5.973293	55.1778	0	3.772509	0

2. Data Preprocessing:

Objective: Data preprocessing before analysis.

Steps:

1. Impute missing values by mean/median for numeric, mode for categorical variables.
2. Normalize numeric variables to scale features appropriately.
3. Outliers use interquartile range (IQR), or z-score method to remove
4. Engineer features based on seasonality, temporal trends, and special events.

Adding time-based features

```
df['hour'] = df['datetime'].dt.hour
df['dayofweek'] = df['datetime'].dt.dayofweek
df['month'] = df['datetime'].dt.month
df['quarter'] = df['datetime'].dt.minute // 15
```

Adding cyclic features, lag features and smooth target variables

```
df['hour_sin'] = np.sin(2 * np.pi * df['hour'] / 24)
df['hour_cos'] = np.cos(2 * np.pi * df['hour'] / 24)
df['dayofweek_sin'] = np.sin(2 * np.pi * df['dayofweek'] / 7)
df['dayofweek_cos'] = np.cos(2 * np.pi * df['dayofweek'] / 7)

df['delhi_lag_1'] = df['delhi'].shift(1).fillna(method='bfill')
df['delhi_lag_2'] = df['delhi'].shift(2).fillna(method='bfill')

df['delhi_smoothed'] = df['delhi'].rolling(window=4).mean().fillna(df['delhi'])
```

3. Exploratory Data Analysis:

Goal: Understand the patterns and correlations of the data.

Steps:

1. Chart demand over time in line charts.
2. Determine if there is any correlation between demand and the weather variables in a heatmap.
3. Analyze variations of demand due to holidays and events.

4. Model Building:

Goal: Train electricity demand models.

Steps:

1. Use ARIMA: Determine parameters (p, d, q) through autocorrelation plots.

   Fit the model to historical data for linear forecasting.

2. Implement Random Forest: Use tree-based regression to handle non-linear relationships.

   Perform hyperparameter tuning for optimal performance.

3. Apply LSTM: Design sequential layers to extract the temporal dependencies. Train on normalized batches of data for performance.

5. Model Comparison:

Objective: Compare the accuracy of models.

Metrics Used:

Mean Absolute Error (MAE): It measures average error in magnitude.

Root Mean Square Error (RMSE): Penalizes larger errors more heavily.

Steps:

1. Split the dataset into training and test sets (80%-20%).
2. Evaluate models on test data using MAE and RMSE.
3. Perform cross-validation for robustness.

MODEL	MAE	RMSE
RANDOM FOREST	262.67187716931124	327.92816095200567
LSTM	3555.6621787165345	3640.2605378015123
ARIMA	789.3525708223117	988.3574450994522

6. Deployment:

Objective: Enable real-time demand forecasting.

Steps:

1. It will pick the best performing model for that; in this case, LSTM.
2. The model will be deployed on cloud or edge systems to ensure scalability.
3. APIs will be created for real-time integration with utility dashboards.

Results

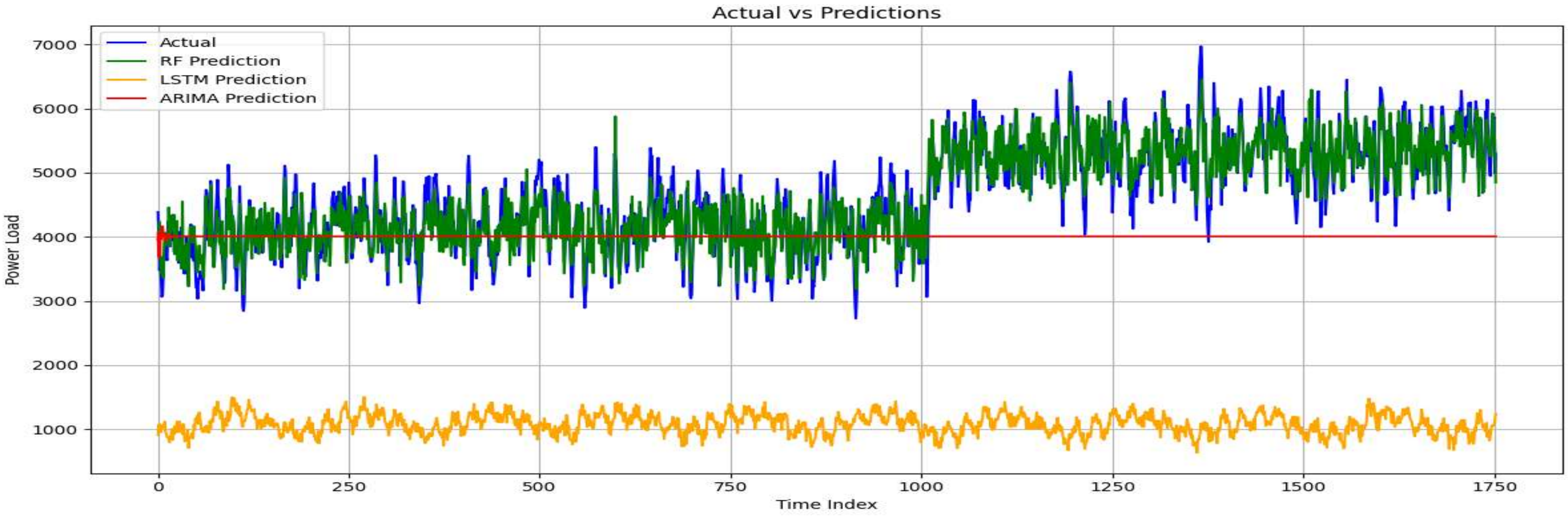


Fig.7: Training dataset Output

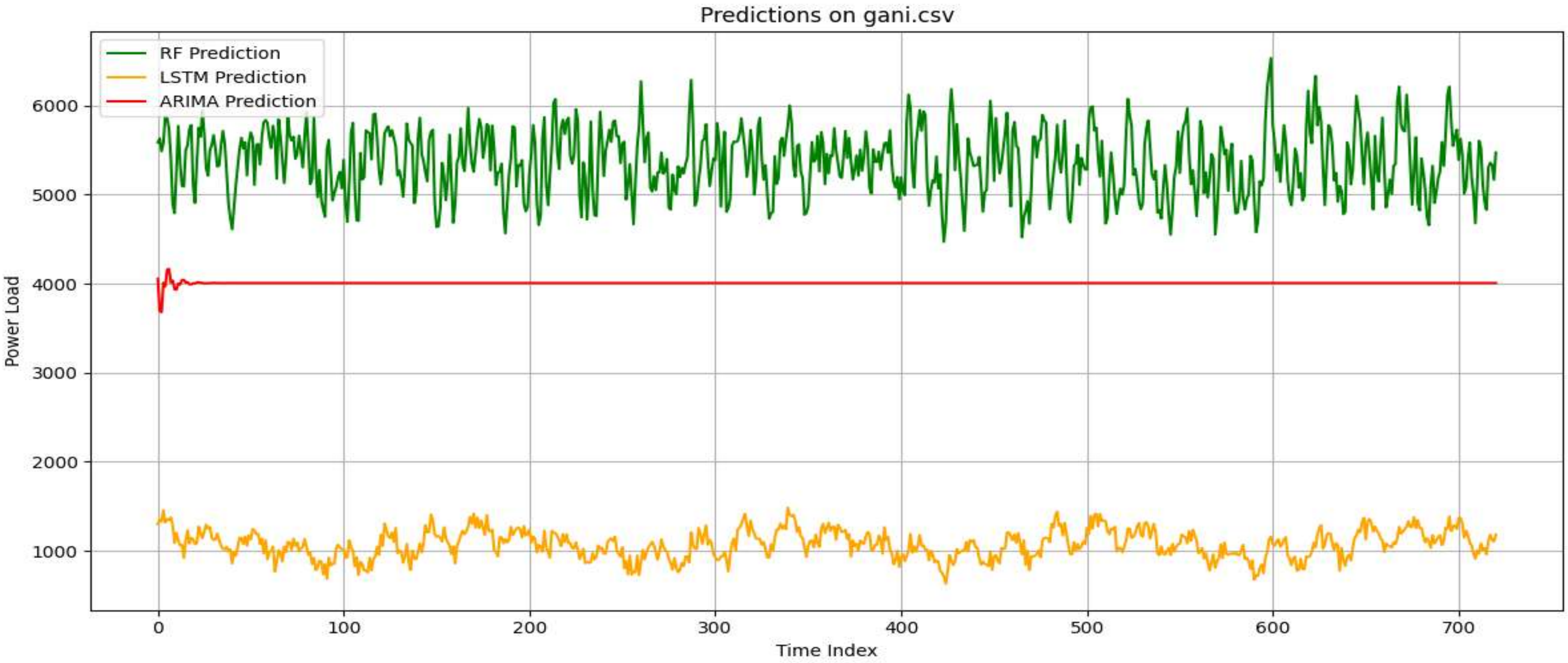


Fig.8: Testing dataset Output

datetime	hour	dayofweek	month	quarter	hour_sin	hour_cos	dayofweek	dayofweek	delhi_lag_	delhi_lag_	RF_Predict	LSTM_Prec	ARIMA_Prediction
02-01-2024 00:00	0	1	1	0	0	1	0.781831	0.62349	5980.068	5980.068	5586.285	1299.158	4056.012
02-01-2024 01:00	1	1	1	0	0.258819	0.965926	0.781831	0.62349	5980.068	5980.068	5631.402	1349.626	3693.798
02-01-2024 02:00	2	1	1	0	0.5	0.866025	0.781831	0.62349	5076.498	5980.068	5487.636	1326.796	3679.638
02-01-2024 03:00	3	1	1	0	0.707107	0.707107	0.781831	0.62349	6267.645	5076.498	5578.643	1454.369	4010.207
02-01-2024 04:00	4	1	1	0	0.866025	0.5	0.781831	0.62349	6525.088	6267.645	5880.66	1317.619	3968.532
02-01-2024 05:00	5	1	1	0	0.965926	0.258819	0.781831	0.62349	5910.683	6525.088	5861.295	1356.578	4156.688

Table.4: New predictions dataset (gani_predictions.csv)

Conclusions and Future Work

Conclusion: This project demonstrates the effectiveness of AI in predicting Delhi's electricity demand and peak usage times by integrating weather, holidays, and real estate data. The LSTM model outperformed ARIMA and Random Forest due to its ability to capture long-term patterns, ensuring higher accuracy.

Advantages:

1. Improved grid reliability and reduced load shedding.
2. Cost optimization via efficient power scheduling.
3. Enhanced integration of renewable energy, supporting sustainability.

Future Plan:

1. Real-time deployment with cloud scalability.
2. Integration of solar and wind energy forecasting.
3. Inclusion of socioeconomic factors for precision.
4. Scalability to other regions and sectors.
5. Collaboration with stakeholders for infrastructure and market optimization.
6. Integration with IoT for smart grid operations.

References

Journal / Conference Papers

[1] J. W. Taylor and P. J. L. McSharry, “Short-term electricity demand forecasting with recurrent neural networks,” *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 363–371, 2007.

[2] H. Chen, Y. Wang, and Z. Wang, “Day-ahead electricity price forecasting using LSTM neural networks,” *Applied Energy*, vol. 236, pp. 356–368, 2019.

[3] M. Weron, “Electricity price forecasting: A review of the state-of-the-art with a look into the future,” *International Journal of Forecasting*, vol. 30, no. 4, pp. 1030–1081, 2014.

[4] K. Hong and S. Fan, “Probabilistic electric load forecasting: A tutorial review,” *International Journal of Forecasting*, vol. 32, no. 3, pp. 914–938, 2016.

[5] A. Azadeh, S. Moghaddam, and Z. Mehrani, “An integrated artificial intelligence model for long-term load demand forecasting,” *Energy*, vol. 35, no. 5, pp. 2113–2121, 2010.

[6] C. Zhang, J. Wang, C. Wang, and J. Zheng, “Short-term electricity load forecasting using time series analysis: A case study,” *Energy Reports*, vol. 4, pp. 167–173, 2018.

Web

[10] "Electricity Load Forecasting Techniques,” https://www.delhisldc.org/, Accessed on [17-Sept-2024].

[11] "Integrating AI in Power Management Systems," https://www.ieeexplore.ieee.org, Accessed on [27-Sept-2024].

[12] "Electricity Demand in Delhi,” https://npp.gov.in/public-reports , Accessed on [2-Oct-2024].
[13] “Keras Documentation – LSTM Layer,” https://keras.io/api/layers/recurrent_layers/lstm/, Accessed on [28-Oct-2024].

[14] “ARIMA Documentation,” https://www.sktime.net/en/latest/api_reference/auto_generated/sktime.forecasting.arima.ARIMA.html , Accessed on [30-Oct-2024].

[15] “Weather Data,” https://accuweather.com , Accessed on [1-Nov-2024].

Thank You!