Forecasting the Future: Electricity Load and Price Prediction Based on Weather

Naga Vara Pradeep Yendluri
NagaVaraPradeepYendluri@my.unt.edu

Tarun Teja Nallan Chakravertula
TarunTejaNallanChakravertula@my.unt.edu

Sai Bhavani Kartheek Sure
KartheekSure@my.unt.edu

Prathyusha Gangisetty
PrathyushaGangisetty@my.unt.edu

Jayabhi Sankar Reddy Illuri
JayabhiSankarReddyIlluri@my.unt.edu

Sathwik Karthikeya Mannava
SathwikKarthikeyaMannava@my.unt.edu

Abstract

Electricity price and load prediction are crucial for effective energy management and planning. In this project, we used two machine learning algorithms, Long Short-Term Memory (LSTM) and Random Forest Regression, to independently predict electricity prices and loads based on weather data. Using a dataset consisting of electricity prices, loads, and weather information such as temperature, humidity, and wind speed, we preprocess the data and conduct feature engineering to extract relevant patterns. After cleaning dataset, we train the LSTM and Random Forest Regression models separately on the preprocessed data. The performance of each model is evaluated using standard metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R2) score.

1. Introduction

In today's world, accurately predicting electricity demand and pricing is essential for efficient energy management. Machine learning (ML) models offer a promising approach to predicting these factors by analyzing historical energy consumption records and weather data. In this project, we explore the application of ML algorithms, like Random Forest Regression and Long Short-Term Memory (LSTM) networks, for predicting electricity load and price patterns.

1.1. Objective

The primary objective of our project is to develop and compare ML models for predicting electricity load and price.

1.2. Dataset

Our analysis is based on two main datasets: one containing weather features such as temperature, humidity, and wind speed, and another containing electricity load and price data. We perform data cleaning, pre-processing, and feature engineering to create a consolidated dataset for model training and evaluation.

1.3. Methodology

1. **Data Pre-processing:** We handle missing values, convert data types, and extract relevant time components to prepare the dataset for model training. We used date from two datasets to combine data into single dataframe and then extract day, time of day, from date and added them to features list.
2. **Model Development:**
   - **Random Forest Regressor:** We trained separate models for load and price prediction, considering features like temperature, humidity, and time components.
   - **LSTM Network:** We developed LSTM models for load and price prediction, using the sequential nature of the data.
3. **Model Evaluation:** We assess the performance of each model using metrics such as Root Mean Squared Error (RMSE) to determine their accuracy in predicting electricity load and price.

2. Related Work

1. **Short-Term Electricity Price and Load Forecasting using Enhanced Support Vector Machine and K-Nearest Neighbor [1]**: This paper presents an enhanced approach for short-term electricity load and price forecasting in Smart Grids. The approach utilizes

data from the New York Independent System Operator (NYISO) and employs Decision Tree for feature selection and Recursive Feature Elimination for extraction. Two classifiers, Support Vector Machine (SVM) and K-Nearest Neighbor (KNN), are used, achieving high accuracies. Specifically, the modified SVM achieves approximately 89.6% accuracy for load forecasting and around 88.3% for price forecasting, while the modified KNN achieves 89.9% and 85.6% accuracy, respectively.

2. **Day ahead hourly load and price forecast in ISO New England market using ANN[10]**: This study introduces an AI-based method for forecasting day-ahead hourly load and price in the ISO New England market. The approach relies on artificial neural networks (ANN) and historical data on temperature, electricity load, and natural gas prices. The ANN model produces highly accurate forecasts, which can assist power producers and consumers in devising optimal bidding strategies to enhance profitability.

3. **Electricity load forecasting using fuzzy logic[2]**: This study introduces a fuzzy logic-based method for short-term electricity load forecasting, with a focus on the influence of weather parameters. By incorporating weather and temperature data, the model improves forecast accuracy, aiding in effective generation planning and reserve management for system operators. Notably, the model excludes season-dependent factors like agricultural load, concentrating solely on short-term load prediction.

3. Proposed Method

The proposed method involves several steps for forecasting electricity load and price using machine learning techniques:

1. **Data Cleaning and Pre-processing:** The code starts by reading and cleaning two main datasets containing weather features and energy consumption records. Missing values are handled, and data types are converted as needed.

2. **Feature Engineering:** Feature engineering is performed to extract relevant time components such as hour, day of the week, and month, which are crucial for time series forecasting.

3. **Model Development:**

    - **Random Forest Regressor:** Separate models are trained for load and price prediction using features like temperature, humidity, and time components. Random Forests are known for their ability to handle non-linear relationships in data and are effective for regression tasks.
    - **LSTM Network:** LSTM models are developed for load and price prediction. LSTM networks are well-suited for sequential data and can capture long-term dependencies, making them suitable for time series forecasting.

4. **Model Evaluation:** The performance of each model is evaluated using metrics such as Root Mean Squared Error (RMSE) to assess their accuracy in predicting electricity load and price.

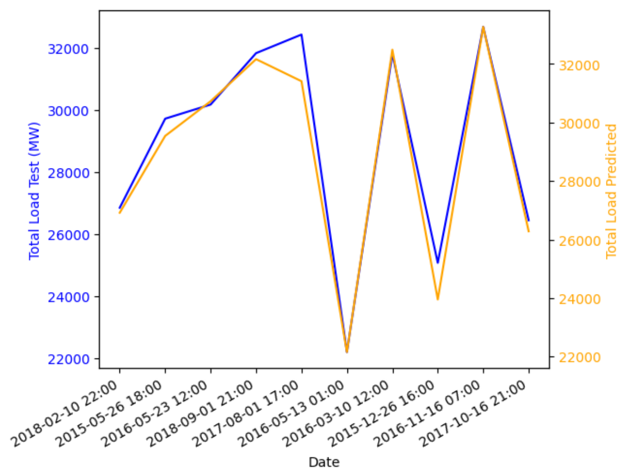

Figure 1. Random Forest - Load Predicted vs Load Actual

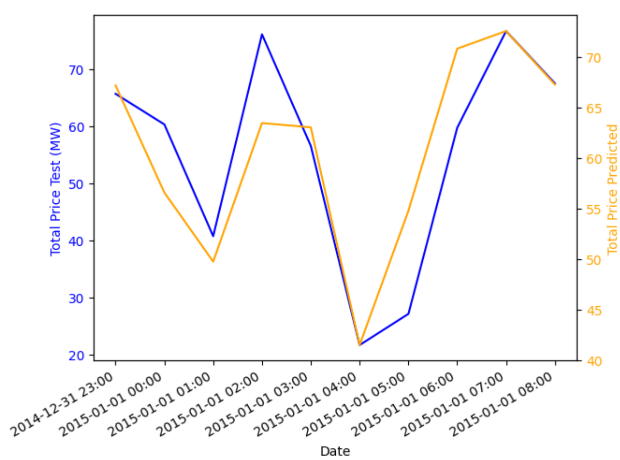

Figure 2. Random Forest - Price Predicted vs Price Actual

5. **Visualization:** The project includes several visualization techniques to help understand the data and model predictions better. For example, line plots are used to visualize electricity load and temperature over time.

    Overall, the proposed method combines the strengths of Random Forest Regressors and LSTM networks

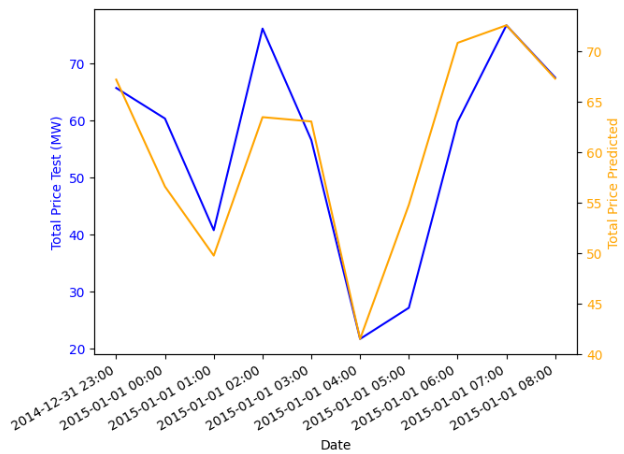

Figure 3. LSTM - Price Predicted vs Price Actual

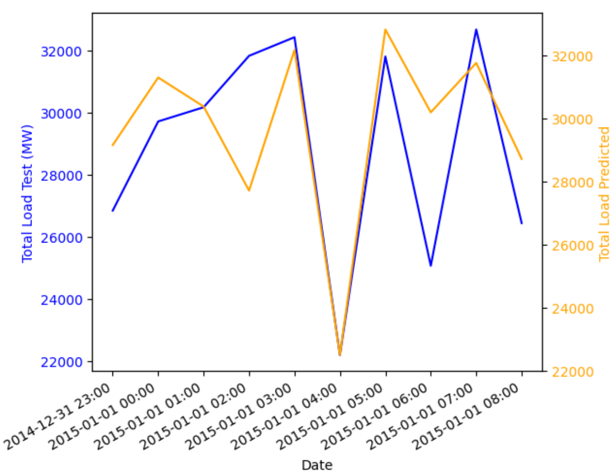

Figure 4. LSTM - Load Predicted vs Load Actual

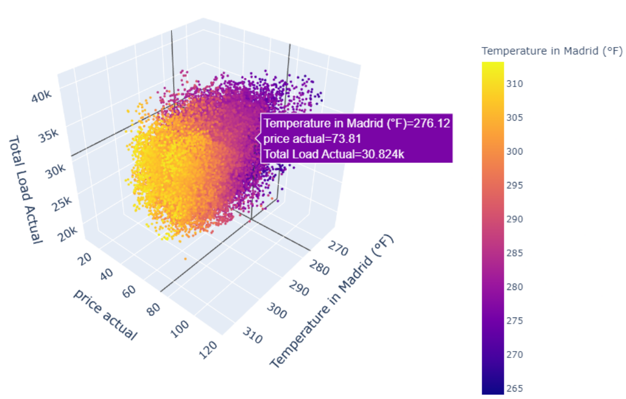

Figure 5. 3D Scatter Plot: Weather Conditions in Madrid vs. Electricity Load

for electricity load and price forecasting, leveraging weather data and historical energy consumption records for improved accuracy.

4. Experiments

The experiments conducted in this study focused on evaluating the performance of machine learning models, specifically Random Forest Regressors and Long Short-Term Memory (LSTM) networks, for the task of electricity load and price forecasting. The experiments were divided into several key components:

4.1. Dataset

The dataset used in this study contains weather features and electricity-related data. It includes weather data for five cities, and the columns used in the analysis along with their importance as features are as follows:

- **temp_Madrid**: Represents the temperature in Madrid. Temperature can significantly affect electricity demand and supply, as it influences heating and cooling loads.
- **humidity_Madrid**: Indicates the humidity level in Madrid. Humidity affects the comfort levels indoors and can alter energy consumption patterns, particularly for air conditioning.
- **wind_speed_Madrid**: Wind speed in Madrid. This is particularly relevant for regions relying on wind energy, as it directly impacts the generation capabilities of wind turbines.
- **price actual**: The actual price of electricity at a given time. This is the target variable for price forecasting models and reflects the immediate market conditions influenced by demand and supply.
- **hour**: Hour of the day, recorded as an integer from 0 to 23. Electricity usage patterns vary significantly throughout the day, and this feature helps model these variations.
- **day_of_week**: Day of the week encoded as an integer (0=Monday, 6=Sunday). Energy usage trends can differ substantially on weekends compared to weekdays.
- **month**: Month of the year, recorded numerically. This feature captures seasonal variations in energy usage, which are influenced by factors such as weather conditions and holiday periods.
- **total load actual**: The actual total electrical load or demand observed. This is a crucial measure for energy supply management and is influenced by various factors including the mentioned meteorological conditions and economic activities.

These features were selected based on their known impact on electricity demand and price, and they play a crucial role in the forecasting models developed in the study.

4.2. Software

The software used in this project includes:

1. Python: Python programming language was used for data preprocessing, model development, and analysis.[11]
2. Pandas: Pandas library was used for data manipulation and analysis, especially for handling the datasets.[8]
3. NumPy: NumPy library was used for numerical computing, especially for array operations and mathematical functions.[9]
4. Matplotlib and Seaborn: Matplotlib and Seaborn libraries were used for data visualization, especially for creating plots and charts.[4]
5. Scikit-learn: Scikit-learn library was used for machine learning tasks, such as model training, evaluation, and prediction.[3]
6. TensorFlow and Keras: TensorFlow and Keras libraries were used for developing and training the LSTM models.[6]
7. Plotly Express: Plotly Express library was used for creating interactive 3D scatter plots.[7]
8. Google Colab: Google Colab was used as the development environment for the project, providing access to computational resources and collaborative features.[5]

These software tools were instrumental in conducting the experiments and analysis described in the project.

4.3. Hardware

The hardware resources used in this project include:

- Google Colab: Google Colab provided access to a virtual machine with a GPU for running the code.
- Personal Machine: If running the code on a personal machine without Google Colab, the following hardware specifications would be needed:
  - A CPU with sufficient processing power to handle data processing and model training tasks. A modern multi-core processor would be ideal.
  - A GPU (optional but recommended) for faster training of deep learning models like the LSTM network. NVIDIA GPUs are commonly used for this purpose.
  - Sufficient RAM (at least 8GB recommended) to handle the dataset and model data structures. More RAM may be needed depending on the size of the dataset and complexity of the models.
  - Adequate storage space for storing datasets, code, and any intermediate or output files generated during the analysis.

Running the code on a personal machine may require installing necessary libraries manually and managing dependencies. Training times for models, especially the LSTM network, may be longer compared to using a GPU-accelerated environment like Google Colab.

4.4. Experiment: Effect of Incorporating Forecast Data on Model Accuracy

4.4.1 Objective

The goal of this experiment was to assess the impact of incorporating "day ahead price" and "load forecast" data into predictive models on the accuracy of load and price forecasts. This was done by comparing the performance of models with and without these forecast features.

4.4.2 Feature Selection

Initial features included temperature, humidity, and wind speed from Barcelona, along with time-based features such as hour, day of the week, and month. To test the impact of forecast data, "total load forecast" and "price day ahead" were added to the feature set in subsequent trials.

4.4.3 Evaluation Metrics

The models were evaluated using the following metrics:

- Root Mean Squared Error (RMSE)
- Mean Absolute Error (MAE)
- Coefficient of Determination (R^2)

Root Mean Squared Error (RMSE) Mean Absolute Error (MAE) Coefficient of Determination (R^2)

4.4.4 Experiment Considerations

Data Leakage: Ensure that using these forecast columns does not introduce data leakage—where the model has access to information about the future that it wouldn't have in a real-world scenario. If "total load forecast" and "price day ahead" are values that would be known at the time of making predictions in a live setting, they are safe to use. **Correlation and Redundancy:** Check the correlation between these forecast columns and the actuals. If they are highly correlated, they can significantly enhance model accuracy. However, if they are too closely aligned with the target variable, they might reduce the model's ability to generalize. **Model Complexity:** Adding more features can increase the complexity of the model. It's essential to balance complexity with performance, ensuring the model remains interpretable and manageable.

4.4.5 Experiment Results

Model	RMSE	MAE	R^2
Random Forest	412.75	288.041	0.99
LSTM	440.27	322.67	0.99

Table 1. Performance metrics of the Random Forest model & LSTM in price prediction

Model	RMSE	MAE	R^2
Random Forest	5.534	3.3579	0.8488
LSTM	7.3950	5.1390	0.730

Table 2. Performance metrics of the Random Forest model & LSTM in price prediction

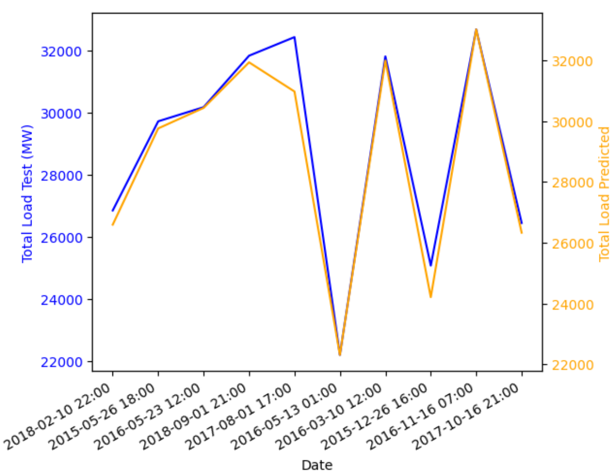

Figure 6. Experiment Model - Random Forest - Load Actual vs Load Predicted

5. Results and Analysis

- The results of the experiments showed that both Random Forest Regressor and LSTM networks were effective in predicting electricity load and price.
- The LSTM network demonstrated superior performance in capturing complex temporal patterns, particularly for price forecasting.
- The models' performance was further improved through feature engineering and data pre-processing techniques.

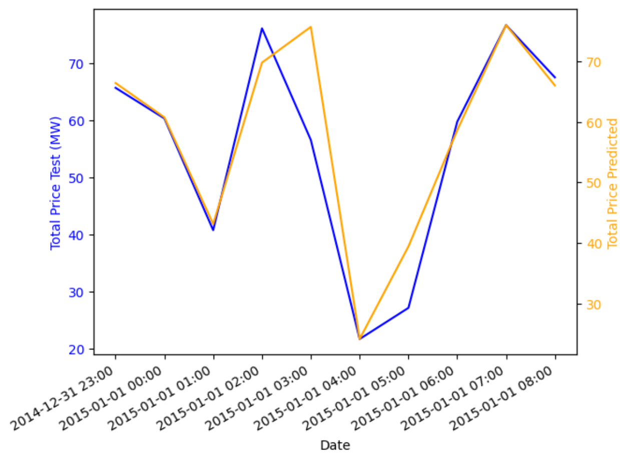

Figure 7. Experiment Model - Random Forest - Price Actual vs Price Predicted

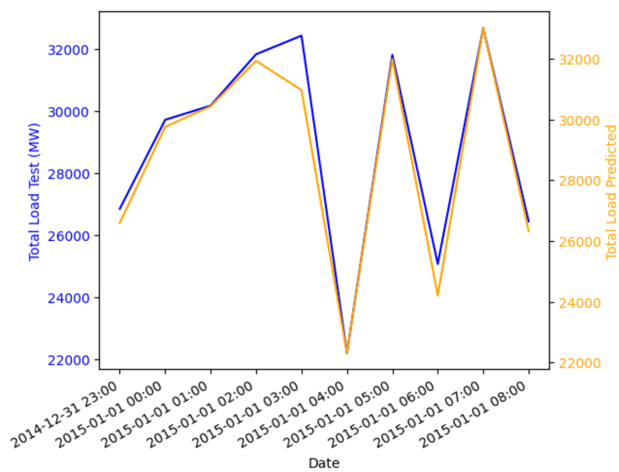

Figure 8. Experiment Model - LSTM - Load Predicted vs Load Actual

5.1. Final Results

In this project we compared two models: Long Short-Term Memory (LSTM) and Random Forest Regressor. Here is the comparision between load prediction and price prediction in both models.

Model	RMSE	MAE	R^2
Random Forest	2365.998	1616.823	0.7384
LSTM	2798.170	2091.270	0.6341

Table 3. Comparison of Load Prediction Performance Between Random Forest and LSTM Models

5.2. Discussion

Both in load and price predictions, Random Forest outperformed the LSTM model. This indicates that the feature set used, which consisted of temperature, humidity, wind

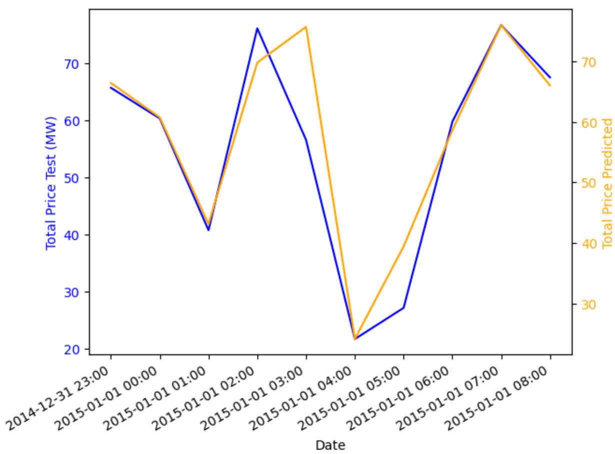

Figure 9. Experiment Model - LSTM - Price Predicted vs Price Actual

Model	**RMSE**	**MAE**	$\mathbf{R^2}$
Random Forest	8.583	6.216	0.6365
LSTM	10.491	8.259	0.4568

Table 4. Comparison of Price Prediction Performance Between Random Forest and LSTM Models

speed from many cities, was well-captured by the Random Forest ensemble method for features, which takes care of the non-linearity of variables properly. The poorer results of the LSTM could be, among other reasons, due to the complexity of the model and the nature of the data. Though LSTM can actually handle temporal sequences, it needs careful tuning and an abundance of training data in order to well generalize, both aspects that perhaps were not ideal in this case.

6. Project Web Interface

To increase the accessibility and practical application of our Random Forest model, we have developed a web interface that is publicly accessible at `https://ml.pradeep.win`. This platform serves as a frontend showcase of our predictive model, allowing users to interact with the model in real-time.

6.1. Web Interface Features

The website is designed to be user-friendly and provides a seamless interaction with the Random Forest model. Users can input specific parameters related to the weather and time, which are significant predictors in our model. The web interface then displays the predicted electricity load and price based on the input values.

6.2. Technical Implementation

The web interface is built using Flask, ensuring robust performance and security. The backend, which hosts the Random Forest model, interacts with the frontend through a well-defined API, facilitating a smooth data exchange. This setup not only makes the model more accessible to non-experts but also provides a practical demonstration of its capabilities in real-world scenarios.

7. Conclusions

This study conducted a detailed comparison between two sophisticated modeling approaches, Random Forest and LSTM, for forecasting electricity load and price. The results indicate that while the Random Forest model generally demonstrated superior performance in our specific tests, the LSTM model also showed considerable potential under certain conditions.

For load prediction, the Random Forest model yielded a lower RMSE of 2365.998 and a higher R^2 of 0.7384, suggesting stronger predictive accuracy compared to the LSTM's RMSE of 2798.170 and R^2 of 0.6341. In price prediction, similar trends were observed, with the Random Forest achieving a RMSE of 8.583 and R^2 of 0.6365, compared to the LSTM's RMSE of 10.491 and R^2 of 0.4568. These findings highlight the effectiveness of the Random Forest model in capturing complex interactions between variables, which is crucial in the dynamic energy market.

However, the LSTM model, known for its ability to handle sequential data, demonstrated a noteworthy capacity to model temporal dynamics, which is particularly valuable in scenarios where patterns over time are critical. Although the LSTM did not outperform the Random Forest in this instance, its architecture is well-suited for applications requiring the analysis of time-series data, suggesting that with further tuning and adaptation, it could yield significant improvements.

7.1. Future Directions

Some of the following points are discussed in future work to improve our prediction:

- **Improved Feature Engineering:** More weather-related features included and polynomial features added in hopes of capturing more of the interaction detail.

- **Model Tuning and Experimentation:** Even more tuning of the LSTM model parameters and model architectures are experimented with to observe better results.

- **Hybrid Models:** Exploration into possibility of a hybrid model to fuse strength from tree-based models and neural networks, allowing synergy between both models in an attempt to increase prediction accuracy.

The implications of accurate load and price forecasting are significant across several domains. Improved predictions can lead to better grid management, optimized energy production, and more informed policy-making. Additionally, consumers can benefit from more stable prices and availability, potentially leading to lower costs and reduced energy waste.

8. Limitations

This project, while comprehensive, has certain limitations. The models tested were confined to specific dataset and might not perform similarly across different regions or under varying economic conditions. Moreover, the predictive accuracy of the models might degrade over time without retraining and adapting to new data patterns.

9. Further Research

Future research could explore the integration of more diverse data sources, such as economic indicators or more granular consumer usage patterns. Additionally, the development of real-time adaptive models that can dynamically adjust to new data could significantly enhance forecasting accuracy and reliability. Testing the models across different geographic locations would also provide insights into their robustness and adaptability.

10. Potential Applications

Improved forecasting models have the potential to revolutionize energy management systems globally. Utilities can leverage these forecasts to automate and optimize the operation of power plants, integrate renewable energy sources more efficiently, and reduce operational costs. On a larger scale, accurate forecasts can contribute to national energy strategies that promote sustainability and energy independence.

11. Acknowledgements

We would like to thank dataset provider, nicholasjhana (https://www.kaggle.com/nicholasjhana). This was very helpful data in conducting the research and getting the insights provided in this report.

12. Contributions

1. **Tarun Teja Nallan Chakravertula:** Engaged in initial data gathering, pre-processing, and setting up the data cleaning pipelines.

2. **Kartheek Sure:** Focused on the development and fine-tuning of the Random Forest regression models, including hyperparameter tuning.

3. **Naga Vara Pradeep Yendluri:** Led the design and implementation of LSTM model, handled the model training and evaluation phases.

4. **Prathyusha Gangisetty:** Was responsible for data visualization, including creating plots and interpreting the results for the team.

5. **Jayabhi Sankar Reddy Illuri:** Worked on merging datasets and ensuring the integrity of the data throughout the project.

6. **Sathwik Karthikeya Mannava:** Took charge of the final analysis, report writing, and preparation of presentation materials.

References

[1] M. Ali, Z. A. Khan, S. Mujeeb, S. Abbas, and N. Javaid. Short-term electricity price and load forecasting using enhanced support vector machine and k-nearest neighbor. In *2019 Sixth HCT Information Technology Trends (ITT)*, pages 79–83, 2019.

[2] J. Blancas and J. Noel. Short-term load forecasting using fuzzy logic. In *2018 IEEE PES Transmission & Distribution Conference and Exhibition - Latin America (T&D-LA)*, pages 1–5, 2018.

[3] D. Cournapeau. *scikit-learn*. https://scikit-learn.org/, 2007.

[4] M. Droettboom. *Matplot Lib*. https://matplotlib.org/, 2003.

[5] Google. *Google Colab*. https://colab.research.google.com/.

[6] G. B. Google. *Tensorflow*. https://www.tensorflow.org/, 2015.

[7] P. T. Inc. *Plotly*. https://www.plotly.com/, 2013.

[8] W. McKinney. *Pandas*. https://pandas.pydata.org/, 2008.

[9] T. Oliphant. *NumPy*. https://numpy.org/, 2006.

[10] K. B. Sahay and M. M. Tripathi. Day ahead hourly load and price forecast in iso new england market using ann. In *2013 Annual IEEE India Conference (INDICON)*, pages 1–6, 2013.

[11] G. van Rossum. *https://python.org*. python.org, 1991.