# Energy Demand Forecasting for Electric Vehicles

# Project Report

## I. Introduction

The transition to electric vehicles (EVs) is accelerating globally, necessitating accurate forecasting of energy demand to ensure the reliability of power supply systems. As the adoption of EVs increases, understanding and predicting their energy consumption patterns becomes critical for utility companies and policymakers.

This project aims to develop effective forecasting models that predict energy consumption based on historical electricity data delivered for EV charging. Accurate forecasting is essential for optimizing grid operations, reducing costs, and planning infrastructure development.

This report outlines the objectives, methodology, results, and implications of implementing various forecasting models, including Prophet, traditional machine learning algorithms such as Random Forest, Linear Regression, KNN, and ARIMA.

## II. Objectives

The specific objectives of this project are as follows:

1. Implementing the Prophet forecasting model to generate accurate forecasts based on historical energy consumption data.
2. Applying traditional machine learning and statistical models to evaluate their effectiveness in capturing patterns in time series data.
3. Comparing the performance of different machine learning models (Random Forest, Linear Regression, KNN) with ARIMA and Prophet results for the same dataset.

## III. Dataset Description

The dataset used in this work was downloaded from the ACN-Data portal, which provides real data on electric vehicle charging sessions. The dataset includes detailed information about each charging session, such as connection times and energy delivered. A glimpse of the dataset is shown in Table 1.

The characteristics of the dataset are as follows:

* Source: [ACN-Data](https://ev.caltech.edu/dataset)
* Format: CSV file containing records of EV charging sessions.
* Key Columns:
* Date: The date of the charging sessions.
* number\_of\_devices: The count of devices charged on that date.
* total\_kWh\_delivered: The total kilowatt-hours delivered during the sessions.

*Table 1 – Examples from the Dataset*

|  |  |  |
| --- | --- | --- |
| date | number\_of\_devices | total\_kWh\_delivered |
| 2020-01-01 | 4 | 70.029 |
| 2020-01-02 | 16 | 156.19 |
| 2020-01-03 | 14 | 64.08 |
| 2020-01-04 | 10 | 83.119 |
| 2020-01-05 | 11 | 40.056 |
| 2020-01-06 | 25 | 168.634 |
| 2020-01-07 | 28 | 197.765 |
| 2020-01-08 | 27 | 195.688 |
| 2020-01-09 | 35 | 249.751 |

## IV. Methodology

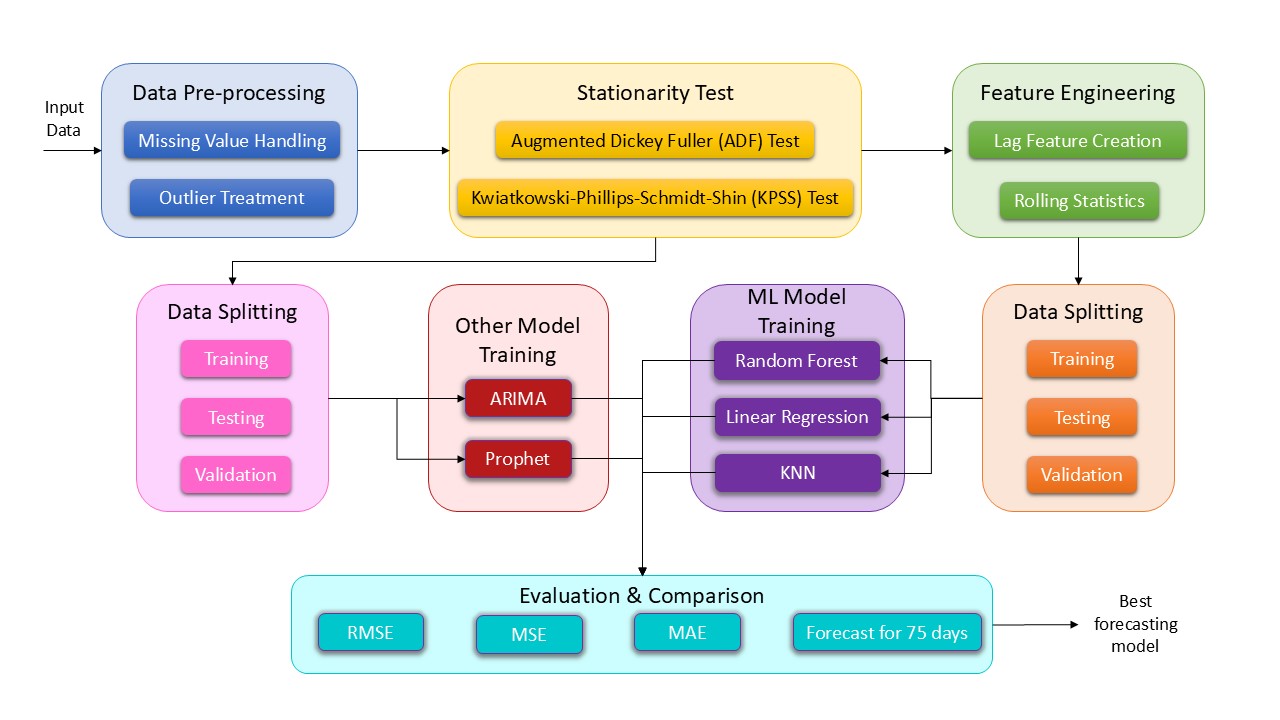
The proposed method deals with the development of several key machine learning models, which can predict the energy demand for electric vehicles. Three machine learning models of Random Forest, K Nearest Neighbors, and Linear Regression will be trained and evaluated. Alongside, the established ARIMA model and state of the art Prophet model will also be constructed. By comparing the performance of the three machine learning models against the advanced models, the most superior model will be identified. The flow of the proposed method is shown in Figure 1.

Figure 1 – Overall Flow of the Proposed Method

### Pre-processing

Data preprocessing are the steps to enhance the quality of input data. It included handling missing values through interpolation and detecting outliers using the z-score method.

* **Handling Missing Values:** Missing values in the total\_kWh\_delivered column are addressed using interpolation methods. This approach ensures that the time series remains continuous without introducing bias from imputation techniques that could distort trends.
* **Outlier Detection and Treatment:** Outliers are identified using the z-score method. Observations with a z-score greater than 3 or less than -3 are considered outliers and are removed from the dataset to prevent them from skewing model predictions.
* **Dropping Unwanted Columns:** The number of devices column was dropped before proceeding with further analysis and modelling.

### Stationarity Test

Before applying time series forecasting models, it is essential to check for stationarity in the dataset. The Augmented Dickey-Fuller (ADF) Test and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test are conducted to assess whether the time series has a constant mean and variance over time.

If the time series is found to be non-stationary (i.e., ADF p-value > 0.05 or KPSS p-value < 0.05), differencing is applied to stabilize the mean of the time series by removing changes in the level of a time series.

### Feature Engineering

This step deals with the creation of new features which help the model make better predictions.

* **Lag Feature Creation**: Lagged features are created for up to 5 days to help machine learning models capture temporal dependencies effectively. This allows models to learn from previous values when making predictions.
* **Rolling Statistics**: Rolling mean and rolling standard deviation are calculated over a window of 7 days to provide additional insights into trends and volatility in energy consumption patterns.

### Data Splitting

In this step, the dataset is divided into distinct subsets for training, testing, and evaluating machine learning models.

The dataset is divided into training, testing, and validation sets based on specified date ranges as shown in Table 2. This division allows for model training on historical data while evaluating performance on unseen data.

Table 2 – Division of dataset in Train, Test, Validation

|  |  |  |
| --- | --- | --- |
| **Set** | **Duration** | **No of Values** |
| Training | 2020-01-01 to 2021-03-31 | 296 values |
| Testing | 2021-04-01 to 2021-06-30 | 92 values |
| Validation | 2021-07-01 to 2021-09-14 | 75 values |

### Modelling

As stated above, a total of 5 models were implemented:

1. **Random Forest:** A robust machine learning model that operates by constructing multiple decision trees during training and outputting the mode of their predictions.
2. **Linear Regression:** A simple yet effective statistical model used for predicting the value of a variable based on its relationship with one or more other variables.
3. **KNN (K-Nearest Neighbors):** A non-parametric method used for classification and regression that predicts the value of a variable based on its nearest neighbors.
4. **ARIMA (AutoRegressive Integrated Moving Average):** A traditional statistical method that captures temporal structures in time series data.
5. **Prophet**: A robust forecasting tool developed by Facebook, designed to handle seasonal effects and holidays effectively.

The implementation of these models was done using Python, with libraries such as Scikit-learn for Random Forest, Linear Regression, and KNN, Statsmodels for ARIMA, and Facebook's Prophet library for Prophet. These libraries provided efficient and effective implementation of the models, allowing for easy experimentation and comparison of results.

The parameters used for each model were as follows:

1. Random Forest used 1000 estimators, a maximum depth of 11, and a random state of 42.
2. Linear Regression required no hyperparameters.
3. KNN used 5 nearest neighbors.
4. ARIMA used an order of (3, 1, 3).
5. Prophet used its default parameters.

## V. Results

The five models of Prophet, ARIMA, Random Forest Regression, Linear Regression, and KNN, were evaluated on the testing set for MAE, RMSE, and MSE values. The results achieved are reported in Table 3. This comparison highlights how each model performs in terms of accuracy when predicting the total kWh delivered.

Table 3 – Performance of ML Models on Testing Data

|  |  |  |  |
| --- | --- | --- | --- |
|  | **RMSE** | **MSE** | **MAE** |
| **Random Forest** | 53.873463 | 2902.349992 | 45.277685 |
| **Linear Regression** | 47.933881 | 2297.656918 | 40.958995 |
| **KNN** | 63.727242 | 4061.161316 | 54.158386 |
| **ARIMA** | 72.116235 | 5200.751418 | 59.980263 |
| **Prophet** | 73.793625 | 5445.499126 | 60.407375 |

From the above comparison in Table 3, it is evident that the Linear Regression model output formed all other models including the well established ARIMA and Prophet models. It achieve the lowest RMSE of 47.93, MSE of 2297.65 and MAE of 40.95. The ARIMA and Prophet models had RMSE of 72.1 and 73.79 respectively, which is significantly higher than that achieved by the linear regression model.

The Linear Regression model is followed by the random forest model, which achieved an RMSE of 53.87, MSE of 2902.34, and MAE of 45.27. At the third place stands the KNN model with RMSE value of 63.72, MSE value of 4061.16 and MAE value of 54.15. All these models have achieved lower error rates compared to the state-of-the-art models and outperformed by a large margin.

Visualizing predictions from different models against actual values allows us to assess model performance visually. It is shown in Figure 2, and clearly indicates that the ML models are giving better predictions compared to ARIMA and Prophet. The most superior among all is the Linear Regression model.

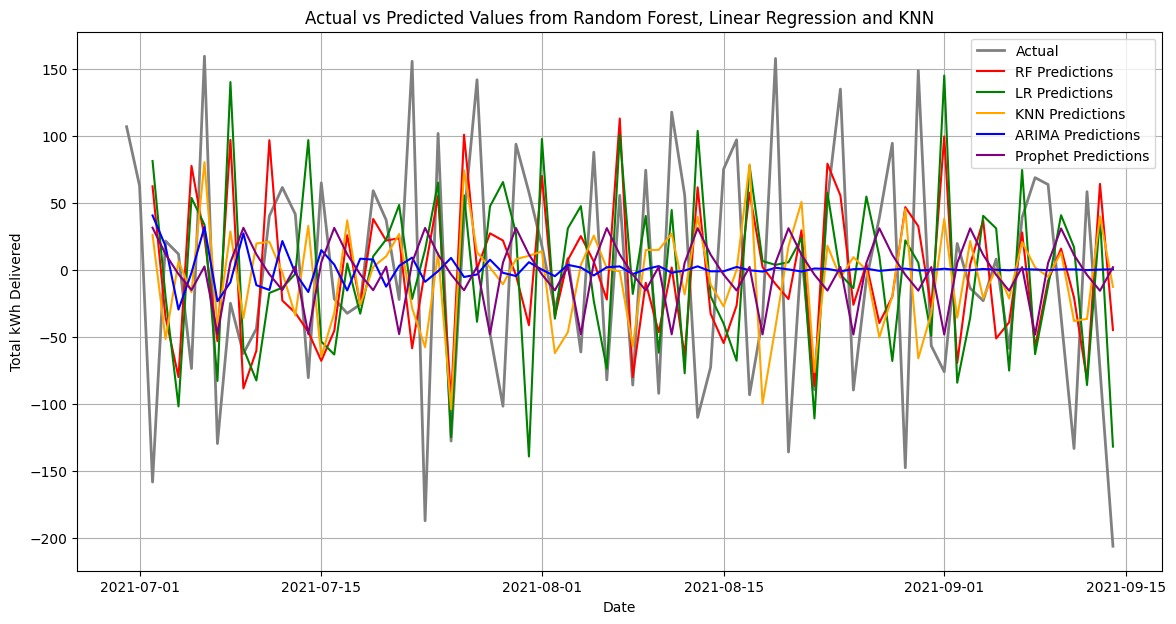


Figure 2 – Forecast Comparison of Different Models

## VI. Conclusion

In conclusion, the work has successfully demonstrated the effectiveness of various forecasting models in predicting energy demand for electric vehicles (EVs) based on historical charging data.

The implementation of the Prophet model, alongside traditional machine learning algorithms such as Random Forest, Linear Regression, KNN, and ARIMA, provided a comprehensive analysis of their capabilities in capturing temporal patterns in energy consumption.

The results revealed that machine learning models, particularly Linear Regression, outperformed both the Prophet and ARIMA models in terms of accuracy metrics such as RMSE, MSE, and MAE.

This finding suggests that machine learning approaches may be better suited for modeling complex relationships in time series data associated with EV energy consumption. The ability of these models to incorporate lagged features and capture non-linear relationships contributed significantly to their predictive performance.

Moreover, the work highlighted the importance of rigorous data preprocessing steps, including handling missing values and outlier detection, which are crucial for ensuring the integrity of the analysis. Stationarity testing further reinforced the need to transform the data appropriately before applying time series forecasting techniques.

The implications of this study extend beyond mere academic interest; they provide valuable insights for utility companies and policymakers. Accurate forecasting models can lead to more efficient energy management strategies, enabling better resource allocation and infrastructure planning for EV charging stations. As the adoption of electric vehicles continues to rise, the demand for reliable energy forecasting will become increasingly critical.

Future work could explore advanced techniques such as hyperparameter tuning, ensemble methods, or deep learning approaches to further enhance forecasting accuracy. Additionally, incorporating external factors such as weather conditions or socio-economic indicators may improve model robustness and predictive capabilities.

Overall, this project contributes to the growing body of knowledge in energy demand forecasting and emphasizes the need for continuous improvement in modeling techniques to meet the challenges posed by the evolving landscape of electric vehicle adoption.