Load Forecasting Using Classical Machine Learning Algorithms and Time Series Models

Navpallav Borthakur, *Student,* Vedash Pandey, *Student,* Vaibhav Joshi, *Student*, under Dr. Avirup Maulik, Department of Electrical Engineering, Indian Institute of Technology (BHU), Varanasi.

***Abstract*—In the realm of electricity, accurately predicting demand plays a vital role in planning regular operations and developing infrastructure. The deregulation of energy markets has led to an increase in the complexity of demand patterns. Consequently, it has become challenging to find a suitable forecasting model that can be universally applied to all electricity networks. Although several forecasting techniques have been developed, no single method can cover all demand patterns. This review paper seeks to examine the complexities of various cutting-edge existing solutions, evaluate their strengths and limitations, and identify the opportunities and challenges associated with forecasting tools. By utilizing a geolocalized dataset from SLDC Maharashtra, our objective is to establish a benchmark for these solutions, particularly in the context of Indian metropolitan scenarios and recent surges in load demand. This benchmarking exercise aims to provide novel insights into the task of load forecasting.**

***Index Terms*— Load forecasting using classical Machine Learning and Time Series Analysis**

# I. INTRODUCTION

Electricity is a distinctive commodity that cannot be stored and must be generated upon demand. The primary goal of commercial electric power companies is to ensure the safe and stable provision of electricity to end-users. Electric Load Forecasting plays a critical role in planning and operating the electricity industry and power systems. Accurate forecasts of electricity demand result in cost savings for operations and maintenance, improved reliability in power supply and delivery, and informed decision-making for future development. Forecasts are categorized based on the duration of the planning horizon. Short-term forecasts cover periods up to one day or one week, medium-term forecasts span from one day or one week to one year, while long-term forecasts encompass durations exceeding one year. Short-term forecasts are employed to schedule electricity generation and transmission, medium-term forecasts assist in planning fuel purchases, and long-term forecasts facilitate the development of the power supply and delivery system. The electricity demand pattern is influenced by various factors, including time, social, economic, and environmental factors, which contribute to complex variations. Social and environmental factors introduce randomness or noise into the load pattern. As a result, Electric Load Forecasting methods have evolved to become increasingly intricate in order to accommodate the diversity and complexity of demand patterns. The following report offers a range of Load Forecasting methods,

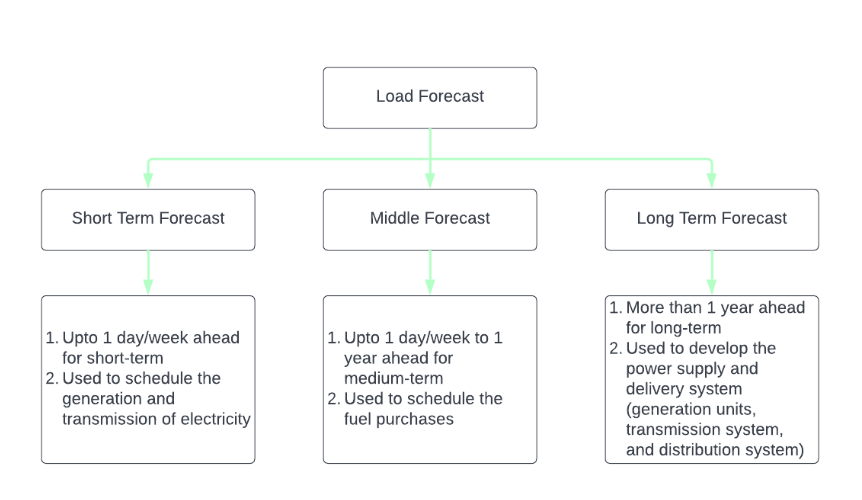
Linear Regression, Random Forest, ARIMA, XGBoost, and LSTM.

Fig.1. Flow chart of types of Load Forecasting.

The nature of electricity as a non-storable product necessitates its generation on-demand, making accurate Electric Load Forecasting vital for the planning and operation of the electricity industry. Load forecasting is conducted over different time periods, including hourly, daily, weekly, monthly, and yearly, categorized as short-term, medium-term, and long-term forecasts based on the planning horizon.

The complexity of demand patterns arises from diverse factors, including temporal, social, economic, and environmental aspects, leading to the development of intricate forecasting methodologies. While ARIMA models have shown success in Electric Load Forecasting, there is no universally applicable method that can effectively handle all cases, especially when multiple factors are considered. Consequently, each electric power plant must adopt a customized forecasting approach by modifying general methods to suit their specific circumstances. This paper proposes a practical forecasting methodology that integrates diverse forecasting models for short periods. By utilizing real-time data from SLDC Maharashtra and incorporating weather details, our aim is to establish a benchmarking platform for analyzing and advancing algorithms in the field of load forecasting.

# II. Dataset Used for Predictions

# A. Sldc Maharashtra Dataset

The primary objective of this research project is to create a benchmark for real-time prediction by utilizing the time series data of theState Load Dispatch Centre Maharashtra. To conduct this study, we obtained a dataset comprising the state's load values (measured in MW) collected at 1 hour-intervals between 3 January 2015 and 27 June 2020. The data was gathered using a custom data scraping script, resulting in a total of approximately 48,200 data points, with 24 values recorded per day. The model incorporates various features, including date, temperature, precipitation and humidity at the corresponding time, with the aim of providing precise predictions. The ultimate goal is to leverage these predictions to effectively manage electricity demand, leading to cost reduction and improved reliability.

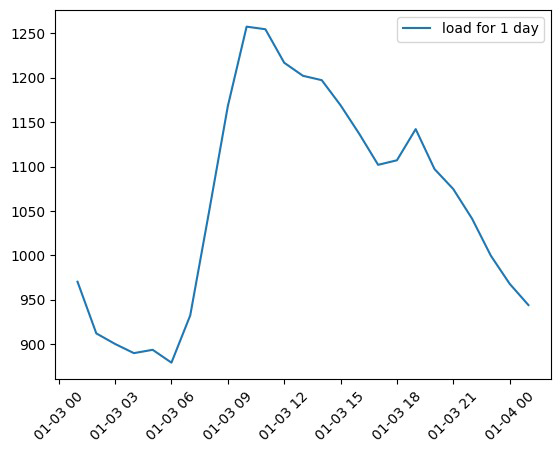


Fig.2. Load curve for Delhi on 3rd January 2018.

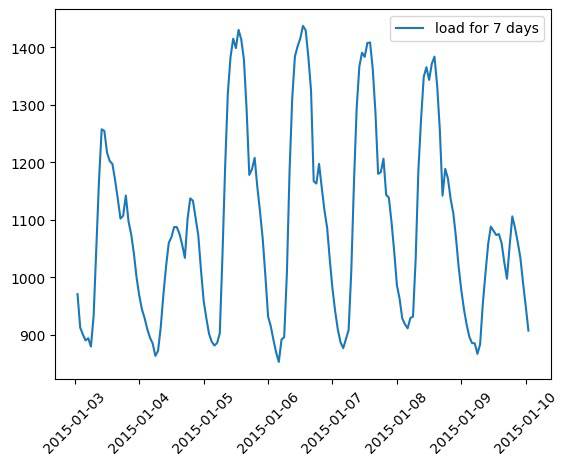


Fig.3. Weekly Load Curve for Maharashtra

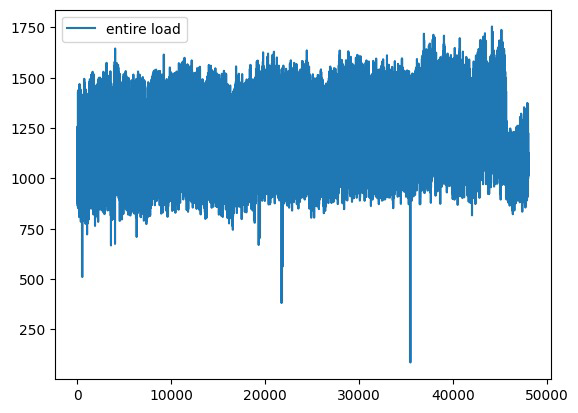


Fig.3. Load curve for entire Data

# B. Data Preprocessing

Maintaining the comprehensiveness and consistency of data holds utmost importance when it comes to precise analysis and prediction. Nevertheless, when data is gathered from online sources, the possibility of encountering missing values arises due to factors like website downtime or maintenance. In the specific context of our study, it is plausible that the data extracted from the SLDC website of Maharashtra may contain certain missing values resulting from such circumstances. To address this concern, conventional techniques such as forward fill and backfill were employed to impute the missing values within the dataset. By applying these techniques, the data was rendered more consistent and complete, thereby ensuring the reliability of subsequent analysis and prediction tasks.

# C. Data Analysis (Trends and Seasonality)

Both additive and multiplicative decomposition are widely utilized techniques for analyzing time series data and uncovering trends, seasonality, and other underlying patterns. Additive decomposition involves separating the time series data into distinct components, namely trend, seasonality, and residual. The trend component represents the overall direction of the data over time, while the seasonality component captures the recurring patterns. The residual component comprises the remaining unexplained data after considering the trend and seasonality components.

In contrast, multiplicative decomposition employs a similar breakdown of the time series data into trend, seasonality, and residual components, but with a multiplication-based approach. This method is particularly useful when the magnitude of the seasonality component varies with the trend component. Both additive and multiplicative decomposition methods have found extensive application in various domains, including finance, economics, weather forecasting, and energy demand analysis.

By employing these techniques on time series data, analysts can gain valuable insights into trends, seasonality, and other patterns that can support decision-making processes, such as predicting future demand or identifying areas for potential improvement.

# III. Methods

This paper explores a range of methods, including moving averages and non-traditional algorithms like ARIMA and LSTM, which have been applied to the dataset discussed in the previous section.

*A.* *Multiple Linear Regression*

Multiple linear regression is a statistical technique used to model the relationship between a dependent variable and two or more independent variables.The goal of multiple linear regression is to find the best fitting line (or hyperplane in higher dimensions) that describes the relationship between the dependent variable and the independent variables.

The line is determined by estimating the coefficients (or weights) of each independent variable that minimize the sum of squared errors between the predicted values and the actual values of the dependent variable.The multiple linear regression model assumes that the relationship between the dependent variable and the independent variables is linear and additive, meaning that the effect of each independent variable on the dependent variable is independent of the other variables.

The performance of the multiple linear regression model can be evaluated using various metrics such as R-squared, adjusted R-squared, and root mean squared error (RMSE).

Easiest algorithm to implement, Takes less time to process

*B.* *Random Forest*

Random forest regression is an ensemble learning method that combines multiple decision trees to make a prediction.

Each decision tree is trained on a subset of the training data and with a subset of the features. Random forest regression is a powerful algorithm that can handle large datasets with a large number of features. It can also handle missing data and outliers well.

Random forest regression is robust to overfitting, which is a common problem in decision tree algorithms. The algorithm uses a technique called bagging, which involves training multiple trees on different subsets of the data and features.

*We have not used Decision Tree Regression as Random Forest Regression is an Improvement upon Decision Tree,Thus Random Forest Regression is used.*

Random forest regression is relatively easy to implement and requires minimal tuning of hyperparameters. Random forest regression can be used for both continuous and categorical variables, making it a versatile algorithm for a wide range of applications.Random forest regression is relatively easy to implement and requires minimal tuning of hyperparameters.

One potential disadvantage of random forest regression is that it can be slower to train than other regression algorithms, especially on large datasets.

Random forest regression can also be difficult to interpret compared to some other regression algorithms, such as linear regression.

*C.* *XGBoost Gradient Boosting Algorithm*

Gradient boosting is an ensemble learning method that combines the predictions of multiple weak learners (typically decision trees) to create a more accurate and robust model. The key idea behind gradient boosting is to sequentially fit new base learners to the residuals (or errors) of the previous learners, to reduce the residuals at each step.

One of the main features of XGBoost is its optimized algorithm for finding the best splits in the decision trees. Instead of exhaustively searching all possible splits, XGBoost uses "approximate tree learning" to efficiently find the optimal splits, greatly speeding up the training process.

Benefits of the XGBoost Algorithm are:

* **Accuracy:** Known for its improved accuracy compared to traditional methods.
* **Efficiency:** Optimized for speed and efficiency, making it well-suited for large-scale load forecasting tasks. It can handle large datasets and nonlinear relationships.
* **Flexibility:** XGBoost can handle various data types, including numerical, categorical, and ordinal data. This makes it well-suited for load forecasting tasks involving different features.
* **Regularization:** Handles missing data values effectively by assigning a default direction for them when splitting a tree node. This means the missing values are treated as either left or right children, depending on which direction gives the best split.

*D. AutoRegressive Integrated Moving Average (ARIMA)*

ARIMA models serve as an alternate approach to time series forecasting, providing a complementary perspective to exponential smoothing models. While exponential smoothing models primarily concentrate on capturing the trend and seasonality within the data, ARIMA models aim to capture the autocorrelations present in the data. This modeling technique finds extensive application across diverse domains such as finance, economics, and engineering.

The components of an ARIMA model can be elucidated as follows:

* **Autoregression (AR)** pertains to a model in which a variable's current value is dependent on its own past values or lagged values.
* **Integration (I)** involves the process of differencing the raw observations, which aids in making the time series stationary. This entails computing the difference between the current data values and their respective previous values.
* **Moving Average (MA)** incorporates the relationship between an observation and the residual error from a moving average model applied to past observations. It considers the influence of lagged observations on the current observation.

The standard notation for ARIMA models includes three parameters: p, d, and q. These parameters help define the specific type of ARIMA model utilized. Here's a breakdown of their definitions:

* **p:** Represents the number of lag observations included in the AR (autoregressive) component of the model. It indicates the lag order and determines the influence of past observations on the current one.
* **d:** Denotes the number of times the raw observations are differences in the I (integrated) component. This process is carried out to achieve stationarity, removing any trends or seasonality present in the data.
* **q:** Signifies the size of the moving average window in the MA (moving average) component. It determines the order of the moving average and captures the relationship between the current observation and the residual errors from past observations.

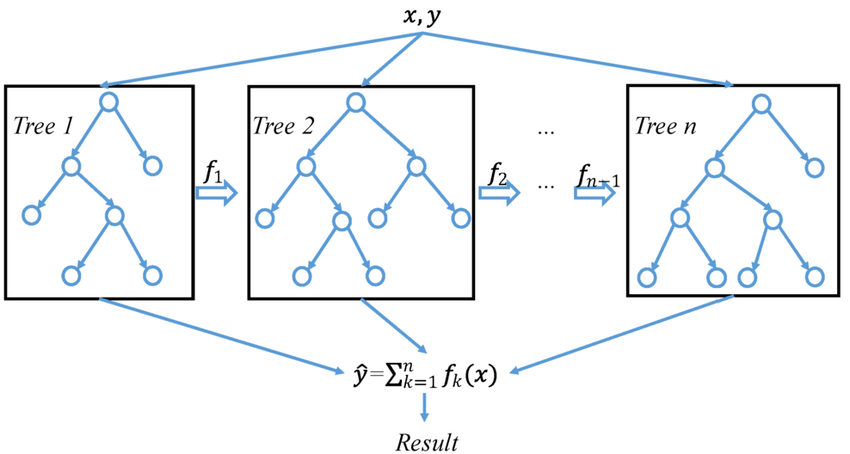


Fig.5. Flow of XGBoost Algorithm

*E. Long Short Term Memory (LSTM)*

LSTM is a type of recurrent neural network that is designed to handle long-term dependencies in sequential data.

It uses memory cells and gates to selectively forget or remember information from the past, allowing it to capture long-term patterns.

The architecture includes an input gate, a forget gate, an output gate, and a memory cell.

LSTM is widely used in tasks such as speech recognition, natural language processing, and time series prediction.

While powerful, LSTM can be computationally expensive and requires a large amount of training data to perform well.

There are many variations of LSTM, including variants with peephole connections, stacked LSTM layers, and bidirectional LSTM

# IV. Results

The following section presents the outcomes of the aforementioned methodologies applied to the load data for different dates. In all the plots below, the actual load data obtained from SLDC Delhi is displayed alongside the load curve forecasted by the aforementioned methods. The real load curve predominantly exhibits the characteristic day-to-day variations.

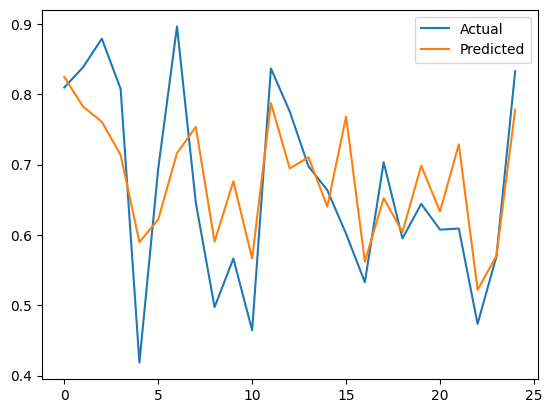


Fig.7. Linear Regression

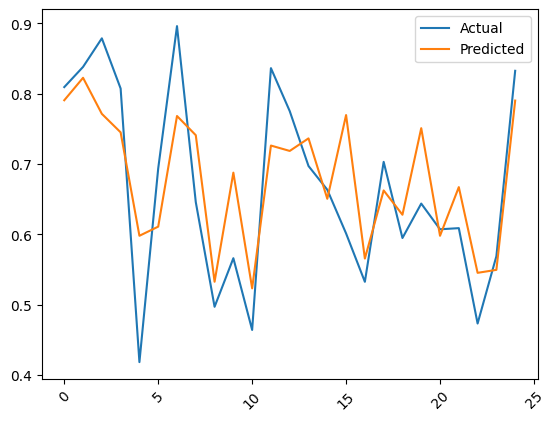


Fig.8. Random Forest Regression

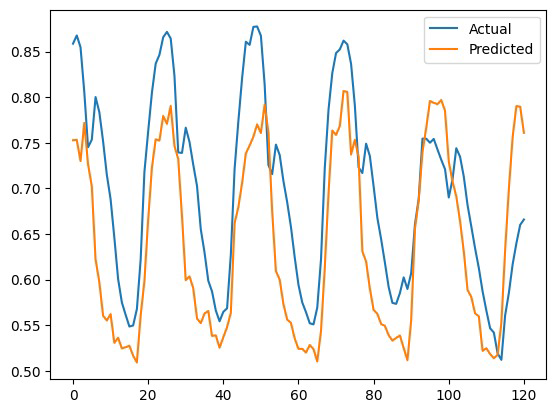


Fig.9. XGBoost

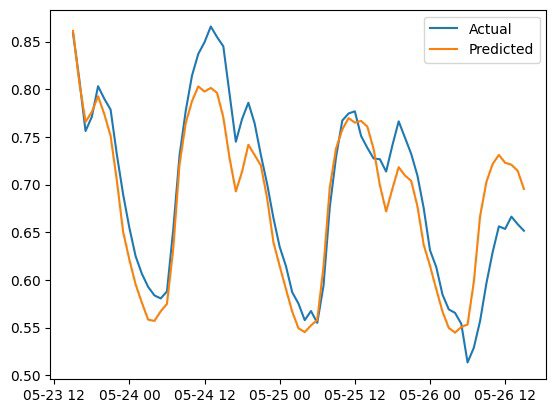


Fig.10. ARIMA

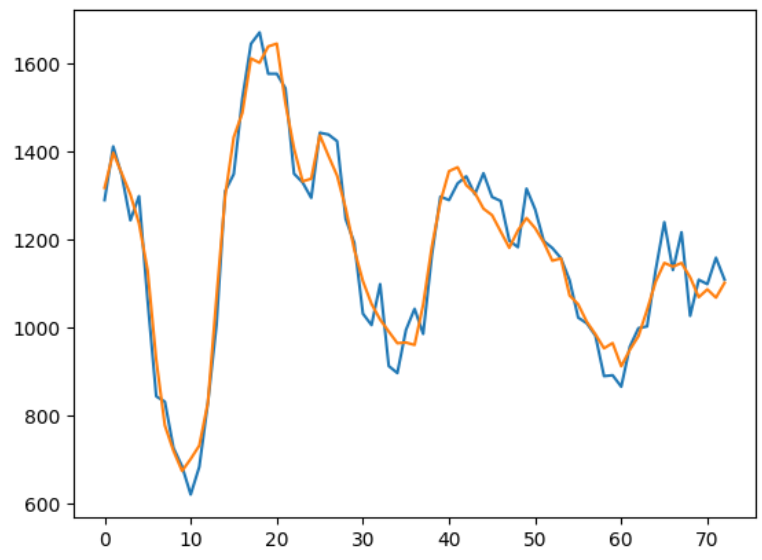


Fig.11. LSTM

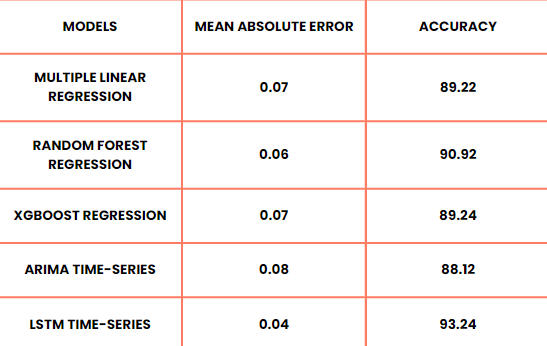


Fig.12. Comparison between different Models

# IV. Conclusion and future prospects

In conclusion, Load Forecasting plays a crucial role in electrical engineering, considering the complexity of transmission and load networks. In this research, we conducted a thorough literature survey and implemented various state-of-the-art techniques for load forecasting. While statistical techniques like Linear Regression, Random Forest Regression, and ARIMA performed well in short-term load forecasting, they struggled to capture the intricate temporal patterns in load time series analysis. Deep learning techniques such as LSTM showed promise in handling these complex patterns while considering additional factors like weather and humidity in load forecasting. However, advanced time series forecasting techniques like CNN, RNN, or auto-regressive transformers have not yet been applied to load forecasting, and exploring their potential on large-scale data could yield significant results. In the future, our aim is to prototype these algorithms using real-time data from the Maharashtra Load Dispatch Centre, benchmark their performance, and gain insights into their practical relevance in load forecasting by incorporating factors such as weather conditions.

Furthermore, it is advisable to conduct further research to investigate the potential of hybrid models that leverage the strengths of multiple approaches, aiming to enhance the accuracy of load forecasting. Additionally, the exploration and advancement of novel techniques like deep learning and machine learning could present valuable opportunities for improving the precision of load forecasting.

# References

1. Short-Term Load Forecasts Using LSTM Networks *Author: Shahzad Muzaffar, Afshin Afshari*
2. Hou Shengren , Edgar Mauricio Salazar , Pedro P. Vergara1, Peter Palensky, *Performance Comparison of Deep RL Algorithms for Energy Systems Optimal Scheduling,* Delft University of Technology
3. Slawek Smyl, Grzegorz Dudek, Pawel Pelka, *ES-dRNN: A Hybrid Exponential Smoothing and Dilated Recurrent Neural Network Model for Short-Term Load Forecasting*
4. Yufan Zhang, Honglin Wen, Qiuwei Wu, Qian Ai, *Optimal Adaptive Prediction Intervals for Electricity Load Forecasting in Distribution Systems via Reinforcement Learning*
5. Load forecasting for energy communities: a novel LSTM-XGBoost hybrid model based on smart meter data

*Authors:Leo Semmelmann, Sarah Henni & Christof Weinhardt*

1. Short term load forecasting based on ARIMA and ANN approaches Author links open overlay panel

*Authors: Chafak Tarmanini , Nur Sarma , Cenk Gezegin , Okan Ozgonenel*