Maximum Demand Forecasting in the Delhi Region using Machine Learning

SujayKumar Reddy M
Department of Computer Science & Engineering
Vellore Institute of Technology
Vellore, India
sujaykumarreddy.m2020@vitstudent.ac.in

Gopakumar G
Department of Computer Science & Engineering
National Institute of Technology
Calicut, India
gopakumarg@nitc.ac.in

***Abstract*—The National Capital Territory (NCT) of India, Delhi, is home to New Delhi, the country's capital. Being a significant political and economic hub, Delhi experiences unique considerations in terms of power consumption. As of May 31st, 2023, the highest recorded power demand in Delhi occurred on June 29th, 2022, reaching a peak of 7770 MW. The dataset used in this paper is the daily Power-Supply-Position (PSP) reports generated by the NLDC, GRID Energy Sector, India which is a division of the Ministry of Power. This paper forecasts the maximum demand produced by Delhi taking in consideration of the Producers side by using various Machine Learning Algorithms with different pre-processing techniques such as data Imputation and Transformation Techniques and MAPE (mean-absolute-percentage-error) is used as a metric to evaluate models. The results suggest that Gradient Boosting Regressor with 7 days shift feature extraction gives the highest accuracy through Linear-Interpolation-imputation for univariate data.**

***Index Terms*—Machine Learning(ML), Deep Learning (DL), Demand Forecasting, Indian GRID Energy Sector, Time Series Forecasting, Delhi.**

I. Introduction

The two most important sectors of the Indian Energy GRID are maintained by POWERGRID [2] which has the objectives of running the GRID efficiently and installing transmission lines etc... and the other being the National Load Dispatch Center (NLDC) [1] which is focussed on the Supervision over the Regional Load Dispatch Centres, Scheduling and dispatch of electricity over inter-regional links in accordance with grid standards specified by the Authority and grid code specified by Central Commission in coordination with Regional Load Dispatch Centres, Monitoring of operations and grid security of the National Grid, etc...

India's Statewise Per-capita of Power raised to 1974.4 Kilo-Watt Hours in Delhi in 2018-2019 and all India's 1115.3 Kilo-Watt Hours in 2021-2022 is the highest per-capita of power which is recorded to date according to Reserve Bank of India (RBI) [5]. Statistically, the Fluctuations in Power/Electricity may affect many factors in the Economy and vice-versa.

To address the challenge of Demand Forecasting, our approach begins with time series forecasting, specifically exploring the potential of Auto-Regression Integrated Moving Average (ARIMA). Following a thorough examination of Time Series Forecasting methods, including ARIMA and Seasonal ARIMA (SARIMA), and guided by the findings in our previous paper [11], where we achieved the lowest recorded error score of 15.717, we transition to the realm of Machine Learning. Here, we employ manual feature extraction strategies to further reduce the error rate and enhance forecasting accuracy.

This paper contains five sections. Section-2 reviews different forecasting techniques that different authors have adopted. Section-3 describes the methodology that has been adopted in this paper. Section-4 focuses on Data Preparation and Data Pre-processing, encompassing imputation and transformation techniques. In Section-5, Machine Learning techniques will be explored, incorporating various types of Feature Extraction techniques to identify interactions and relationships between variables.

II. Related Work

For Demand Forecasting, the study adopts a combination of three distinct methodologies: (i) statistical methods, which leverage historical data and time-series analysis, and (ii) machine learning, which utilizes advanced algorithms to identify intricate patterns and relationships within the data. (iii) deep learning, which utilizes Long Short Term Memory (LSTM) and Artificial Neural Network (ANN).

For the Classical methods like Time-series Forecasting methods, Carlos et al [7] analyzed a time series dataset for Brazilian Electricity Demand Forecasting and divides Brazil into two regions and forecasts the Electricity demand according to using ARIMA models. Kakoli et al. [8] used a Seasonal ARIMA model to forecast electricity demand for Assam in the Northeast Region, achieving MAPE of 10.7%. Srinivasa et al [9] provided a forecasting method that is formulated monthly for the whole of India without considering the states and regions. It has been found that the MSARIMA model outperforms CEA forecasts in both in-sample static and out-of-sample dynamic forecast horizons in all five regional grids in India.

For Machine Learning methods, Christos et al. [14] focused on forecasting Peak Demand from the Producer's side using a dataset from three regions in the Netherlands. They compared methods like ARIMA, Ridge Regression, and Lasso Regression, and found that the Bi-directional LSTM model performed the best. Saravanan et al. [13] devised a model based on 64

fuzzy logic neural networks using per-capita Gross Domestic Product (GDP), population, and Import/Export as variables, achieving a low MAPE of 2.3. Mannish et al. [12] proposed an Ensemble Approach for the Distribution Companies (DISCOMs) in Delhi post-Covid, combining XGBoost, LightGBM, and CatBoost algorithms, with an average MAPE of 5.0. Banga et al. [10] compared Machine Learning Algorithms for electricity demand forecasting using a dataset with 29 attributes, highlighting the superiority of the Facebook Prophet model with MAPE scores of 0.4 for daily and 0.2 for hourly datasets.

Anil et al [25] uses Levenberg-marquardt back propagation algorithm ANN on day ahead Short term load Forecasting on the state of Uttar Pradesh trained on hourly data with the MAPE score of average MAPE 3.05, This work suggests to use the ANN model to check with our dataset too. Navneet et al [26] uses the New Delhi Adani Enterprises Ltd (ADEL) data to forecast the load by using different Neural Network Architectures in which ELMANN Neural Network Architecture has given the good accuracy. Dharmoju et al [27] provided a sector of Residental buildings by the United States Dataset by using LSTM model for monthly forecasting. Shaswat et al [28] uses a Temporal Fusion Architecture to capture the interactions which are scaled between 0 and 1 for daily data which acheives 4.15% more than the existing models and this is for the whole India which is not region specific. Saravanan et al [29] uses the economic factors like GDP, national income, consumer price index etc.. with that they used Principle component Analysis following with ANN which gives the highest accuracy of MAPE score 0.43. Vishnu et al [30] concentrates on the work on Renewable Energy Resources devised two major LSTM models.

III. Proposed Methodology

Figure 1 illustrates the structured process flow of this paper. The initial step involves data extraction from the daily reports supplied by the National Load Dispatch Center (NLDC) [3], which are predominantly in the Portable Format Document (.pdf) file format. Given our specific focus on the Union Territory of India, Delhi, we employ the keyword "Delhi" to facilitate the conversion of data within the PDF files into a text format for each individual file. Subsequently, we proceed with data filtration, isolating and extracting the pertinent information related to the keyword "Delhi." This refinement culminates in the creation of the final dataset, formatted as a Comma Separated Values (.csv) file.

After the final dataset has been generated, we employ imputation techniques to address the missing data on days when the report was not generated by the NLDC. We utilize imputation methods, including mean, median, mode, and linear interpolation, to fill in the null values within the dataset. Additionally, we conduct a comparative analysis of these imputation techniques with the original dataset, in which all instances of missing data are removed (referred to as the 'dropna dataset'). This comparative methodology provides valuable insights into which dataset configuration offers the highest accuracy when forecasting daily demand.

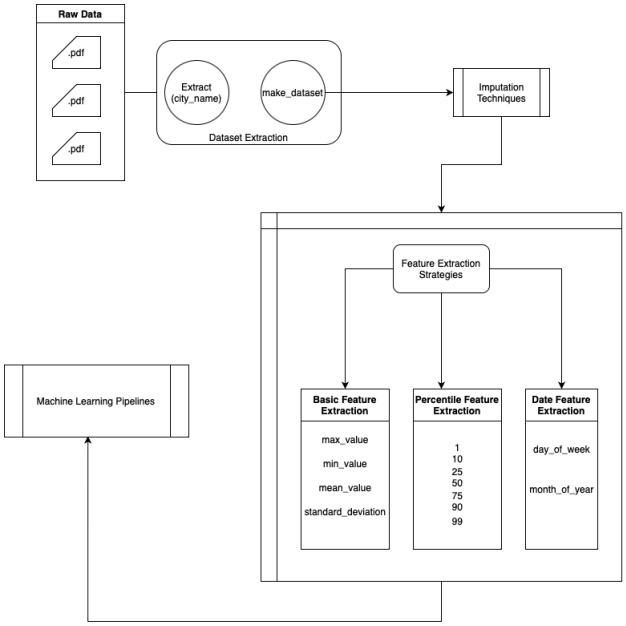

Fig. 1: Flowchart of forecasting process based on Feature Extraction Strategies

Deep Learning algorithms like LSTM and gated recurrent units (GRU) work well for large datasets and for multivariant analysis [25], However, considering that the final dataset encompasses daily data from 2013 to 2023, which may be relatively limited in data points, it necessitates manual feature extraction strategies. For this reason, we opted to employ Machine Learning techniques, utilizing four distinct Regression algorithms: Lasso Regression (LR), Ridge Regression (RR), Gradient Boosting Regressor (GBR), and Support Vector Regressor (SVR). We selected these four algorithms due to their simplicity and the fact that they are grounded in four different working principles. We evaluated their performance using the Mean Absolute Percentage Error (MAPE) as a metric.

As elucidated in Figure 1, we have organized our feature extraction into three primary categories. The first category, Basic Feature Extraction, encompasses fundamental statistical features like mean, max, min, and standard deviation (std). The second category, Percentile Feature Extraction, offers a wider perspective by including features across the 1st to the 99th percentiles, providing insights into data distribution. The third category, Date Features, introduces attributes such as the month of the year and the day of the week, extracted from the available data. This paper provides an overview of the various feature extraction strategies and their impact on forecasting accuracy. However, it's important to note that there is room for further exploration and refinement, especially in the realm of combined feature extraction techniques.

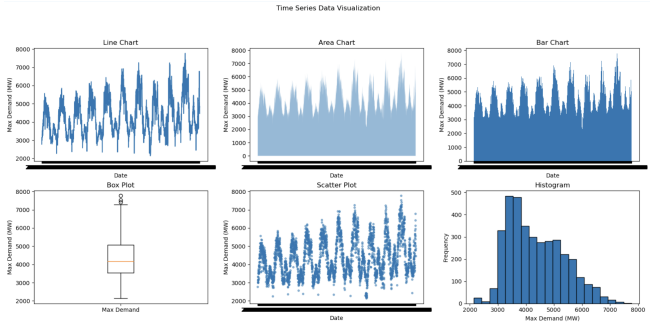

Fig. 2: Different types of Vizualizations for the data extracted from the output .csv file

IV. Data Preparation and Statistics

In this study, we utilized daily generation reports spanning from April 1, 2013, to May 31, 2023, generating a total of 3713 data points. To ensure a robust evaluation of our forecasting algorithms, we divided the dataset into three major segments as shown in Table 1. This division allows for the training of all the algorithms on the training data and subsequently testing their performance on the testing data, from which MAPE scores were obtained. These MAPE scores were pivotal in ranking the performance of our algorithms. Given the relatively small dataset size, we adopted a monthly coherence approach to segment the data effectively. Specifically, we used the most recent month for testing purposes and the subsequent month for validation purposes. This methodology optimizes the utility of the available data and ensures that our algorithms are rigorously evaluated for their forecasting capabilities.

TABLE I: Data Division Methodology

April 2013 to April 2023	Train
May 2023	Test
June 2023	Validation

The dataset provided by the Dataset Generation Algorithm has 8 features Date (YYYY-MM-DD), Max. Demand met during the day (MW), Shortage during maximum Demand (MW), Energy Met (MU), Drawal Schedule (MU), OD(+)/UD(-) (MU), Max OD (MW), Energy Shortage (MU). From these columns, we select the Date (YYYY-MM-DD), Max.Demand met during the day (MW). We restrict ourselves for univariant analysis of the data so, we use the column of Date and Max.Demand met during the day (MW). The total number of data points which are available from the daily reports is 3640 with 73 missing data points. This dataset is subjected to preprocessing which generates multiple datasets from Imputation and Transformation Techniques.

Figure 2 offers a comprehensive dataset analysis by presenting six distinct visualization plots: the Line chart, Area chart, Bar chart, Box plot, Scatter plot, and Histogram. Figure 3 elucidates the dataset's seasonality aspect, providing valuable insights into its temporal patterns. Table 2 depicts the statistics

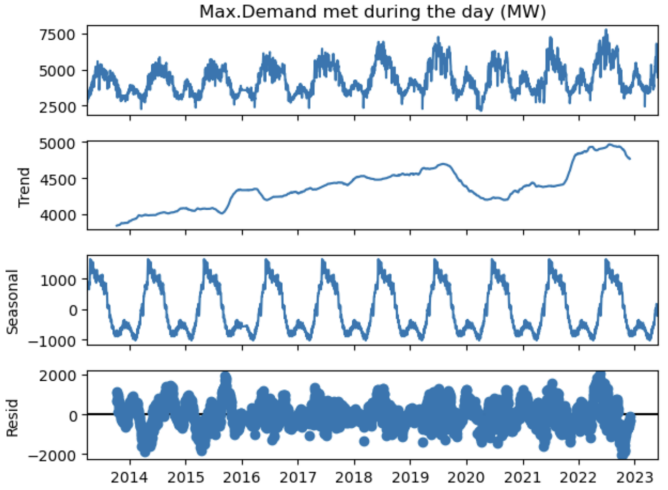

Fig. 3: Seasonal decomposition of the Maximum Demand column with trend and Residuals

TABLE II: Data Statistics

Number of Columns	8
Number of Datapoints	3640
Mean	4356.58
Standard Deviation	994.5
Minimum Value	2139
Maximum Value	7770

of the final dataset where we can see that many of the datapoints are missing. We use four imputation techniques: mean, median, mode, and interpolation. By using these methods, we generate five datasets: dropna-dataset (by dropping all null values), mean-dataset, median-dataset, mode-dataset, and interpolation-linear-dataset. Transformation techniques are also applied, such as sliding window mean. Statistical models like time-series forecasting and machine learning models are applied to each of these datasets, as discussed in the further sections of this paper.

V. Machine Learning Algorithms

This section explains the Machine Learning Algorithms which has been used in this paper. The models considered are Ridge Regression, Lasso Regression, Gradient Boosting Regression, and Support Vector Regression. For the latter subsections, we will be explaining each one and how can it be applied to Time series data.

A. Ridge Regression

Ridge Regression, often used in applications like time series data analysis [16], serves as a valuable tool for improving linear regression. It helps tackle issues such as multicollinearity, which arises when predictor variables are highly correlated, and it prevents overfitting. Ridge Regression introduces a parameter λ that's tailored to the dataset, influencing the linear regression model. The Usual Linear Regression model which is represented according to Equation 1 where y is

the dependent variable, $\beta_0, \beta_1, \beta_2, \ldots, \beta_n$ are the coefficients of the predictor variables (features), $x_1, x_2, \ldots, x_n$ are the predictor variables and ε is the error term.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \varepsilon \tag{1}$$

$$\text{Cost Function} = \sum (y_i - \hat{y}_i)^2 + \lambda \sum \beta_i^2 \tag{2}$$

The goal of Ridge Regression is to minimize the following cost function as shown in the Equation 2 where $\sum (y_i - \hat{y}_i)^2$ is the sum of squared errors (similar to the Ordinary Least Squares method), $\sum \beta_i^2$ is the sum of squared coefficients and λ (lambda) is the regularization parameter. It controls the strength of the regularization. Higher values of λ lead to stronger regularization, which shrinks the coefficients closer to zero. with L2 regularization. The Ridge Regression model aims to find the coefficients $\beta_0, \beta_1, \beta_2, \ldots, \beta_n$ that minimize this cost function. The regularization term $\lambda \sum \beta_i^2$ encourages the model to keep the coefficients small, which helps in reducing the impact of multicollinearity and preventing overfitting.

B. Lasso Regression

Lasso Regression, or Least Absolute Shrinkage and Selection Operator Regression, is a variant of linear regression that incorporates L1 regularization into the cost function. Its appeal lies in its ability to enhance predictive accuracy and foster model interpretability. These LASSO-based methods are not only adept at addressing model uncertainty but also excel in improving forecasting accuracy by accounting for non-pervasive shocks. Particularly in the realm of Dynamic Factor models, especially when applied to time series data, Lasso Regression has established its efficacy in out-of-sample forecast evaluations [17]. The process of solving Lasso Regression involves employing diverse optimization techniques, such as coordinate descent or subgradient descent. Through this methodology, the final coefficients achieve a delicate balance between effective data fitting and feature sparsity, with certain coefficients being driven to zero.

$$CostFunction = \sum (y_i - \hat{y}_i)^2 + \lambda \sum |\beta_i| \tag{3}$$

The Lasso Regression model aims to find the coefficients $\beta_0, \beta_1, \beta_2, \ldots, \beta_n$ that minimize this cost function which is shown in Equation 3, where $\sum (y_i - \hat{y}_i)^2$ is the sum of squared errors (similar to the Ordinary Least Squares method), $\sum |\beta_i|$ is the sum of the absolute values of the coefficients, and λ (lambda) is the regularization parameter. It controls the strength of the regularization. Higher values of λ lead to stronger regularization. The unique aspect of Lasso is that the regularization term $\lambda \sum |\beta_i|$ encourages some coefficients to become exactly zero. This means that Lasso not only fits the data but also performs feature selection, effectively eliminating irrelevant predictors.

C. Gradient Boosting Regressor

Gradient Boosting (GB) learns an additive expansion of simple basis-models. This is accomplished by iteratively fitting an elementary model to the negative gradient of a loss function with respect to the expansion's values at each training datapoint evaluated at each iteration. Notably, Alexandros et al. introduced a GB methodology, as documented in their work [18], which found application in the domain of financial time-series modeling. The goal of Gradient Boosting Regression is to find a predictive model, denoted as $F(x)$, that maps the features X to the target y. The objective in Gradient Boosting is to find the optimal values of β_i and $h_i(x)$ that minimize a loss function, typically a mean absolute percentage error (MAPE) loss function. This is achieved through an iterative process. At each iteration, a new weak learner is trained to capture the errors or residuals made by the ensemble model up to that point.

$$F(x) = \sum_{i=1}^{M} \beta_i h_i(x) \tag{4}$$

This model is constructed as a weighted sum of M individual decision trees, each referred to as a "weak learner." The ensemble model can be represented as shown in Equation 4, where X represents the feature matrix with n samples and m features. It can be written as $X = \{x_1, x_2, \ldots, x_n\}$, y represents the target vector containing the actual values for the n samples, $F(x)$ is the overall prediction for a given input x, M is the total number of weak learners (decision trees) used in the ensemble, β_i represents the weight or contribution of the i-th decision tree, and $h_i(x)$ is the prediction made by the i-th decision tree for input x.

D. Support Vector Regression

Support Vector Regression (SVR) is a variant of Support Vector Machines used for regression tasks. It aims to find a function $f(x)$ that predicts continuous output values, given input features x. The objective is to minimize the regularized loss while ensuring that the errors. The SVR model can be expressed for the timeseries data [19] as shown in the equation 5 where y_t is the target value to be predicted at time t, X_t represents the input features at time t, $f(X_t)$ is the predicted value at time t, w is the weight vector, $\phi(X_t)$ represents the feature mapping, often involving kernel functions to handle non-linearity and b is the bias term.

$$y_t = f(X_t) = \langle w, \phi(X_t) \rangle + b \tag{5}$$

In time series forecasting, input features may include lagged values of the target variable and other relevant time series or exogenous variables. The choice of kernel function and the hyperparameters, such as C and ε, are critical considerations in building an effective SVR model for time series forecasting.

TABLE III: ML model Comparison Results for Null dropped (dropna)

Feature Extraction	Window Size	Prev Days Data	Ridge MAPE	Lasso MAPE	GBR MAPE	SVR MAPE
shift_features	10	30	0.187	0.178	0.131	0.186
shift+date_features	10	30	0.187	0.178	0.131	0.186
shift_features	0	30	0.187	0.178	0.131	0.186
shift+date_features	0	30	0.187	0.178	0.131	0.186
shift_features	10	7	0.172	0.150	0.152	0.204
shift+date_features	10	7	0.172	0.152	0.144	0.213
shift_features	0	7	0.061	0.061	0.059	0.120
shift+date_features	0	7	0.060	0.060	0.059	0.134
percentile_features	10	30	0.127	0.159	0.182	0.226
percentile_features	0	30	0.136	0.136	0.141	0.147
percentile_features	10	7	0.161	0.108	0.159	0.208
percentile_features	0	7	0.090	0.090	0.081	0.097
basic_features	10	30	0.185	0.127	0.181	0.226
basic_features	0	30	0.185	0.127	0.180	0.226
basic_features	10	7	0.175	0.102	0.152	0.212
basic_features	0	7	0.175	0.102	0.152	0.212

TABLE IV: ML model Comparison Results for Mean Imputation

Feature Extraction	Window Size	Prev Days Data	Ridge MAPE	Lasso MAPE	GBR MAPE	SVR MAPE
shift_features	10	30	0.182	0.173	0.132	0.184
shift+date_features	10	30	0.182	0.173	0.132	0.184
shift_features	0	30	0.182	0.173	0.132	0.184
shift+date_features	0	30	0.182	0.173	0.132	0.184
shift_features	10	7	0.172	0.152	0.145	0.201
shift+date_features	10	7	0.173	0.154	0.131	0.209
shift_features	0	7	0.065	0.065	0.063	0.120
shift+date_features	0	7	0.063	0.063	0.058	0.134
percentile_features	10	30	0.131	0.186	0.078	0.226
percentile_features	0	30	0.134	0.134	0.129	0.147
percentile_features	10	7	0.164	0.122	0.160	0.205
percentile_features	0	7	0.089	0.089	0.079	0.098
basic_features	10	30	0.185	0.139	0.156	0.220
basic_features	0	30	0.185	0.139	0.154	0.220
basic_features	10	7	0.174	0.103	0.175	0.205
basic_features	0	7	0.174	0.103	0.175	0.205

E. Evaluation Metric

The Mean Absolute Percentage Error (MAPE) is a commonly used metric for time series and demand forecasting [20], [21], [22], [23], [24]. MAPE gives more weight to large errors, as it takes the absolute percentage difference for each observation as shown in Equation 6 where A_i is the actual value at time i, F_i is the forecasted value at time i, and n is the total number of observations. This means that substantial errors have a more significant impact on the overall MAPE, which can be crucial for forecasting applications where large errors are particularly costly.

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{A_i - F_i}{A_i}\right| \times 100\% \tag{6}$$

F. Results and Discussion

In this paper, we offer a comprehensive comparative analysis of diverse feature extraction strategies within the realm of electricity demand forecasting. Our study aims to make a distinct contribution to the field of demand forecasting without merely replicating minor adjustments made in previous work by other authors.

Statistical feature extraction has been a fundamental and widely adopted strategy in time series data analysis. In our research, we utilize three distinct feature extraction methods, as detailed in Section III of the paper. These methods include "shift_features," which involves creating new features based on past days' data, and "shift+date_features," which combines features derived from shifts with date-related features. These feature extraction techniques play a pivotal role in our ap-

TABLE V: ML model Comparison Results for Median Imputation

Feature Extraction	Window Size	Prev Days Data	Ridge MAPE	Lasso MAPE	GBR MAPE	SVR MAPE
shift_features	10	30	0.182	0.173	0.134	0.181
shift+date_features	10	30	0.182	0.173	0.134	0.181
shift_features	0	30	0.182	0.173	0.134	0.181
shift+date_features	0	30	0.182	0.173	0.134	0.181
shift_features	10	7	0.173	0.153	0.140	0.206
shift+date_features	10	7	0.174	0.154	0.137	0.214
shift_features	0	7	0.065	0.065	0.064	0.119
shift+date_features	0	7	0.063	0.063	0.061	0.134
percentile_features	10	30	0.132	0.184	0.084	0.229
percentile_features	0	30	0.136	0.136	0.136	0.147
percentile_features	10	7	0.164	0.121	0.158	0.211
percentile_features	0	7	0.089	0.089	0.078	0.098
basic_features	10	30	0.187	0.137	0.124	0.223
basic_features	0	30	0.187	0.137	0.124	0.223
basic_features	10	7	0.176	0.105	0.124	0.212
basic_features	0	7	0.176	0.105	0.122	0.212

TABLE VI: ML model Comparison Results for Mode Imputation

Feature Extraction	Window Size	Prev Days Data	Ridge MAPE	Lasso MAPE	GBR MAPE	SVR MAPE
shift_features	10	30	0.186	0.176	0.139	0.183
shift+date_features	10	30	0.186	0.176	0.139	0.183
shift_features	0	30	0.186	0.176	0.139	0.183
shift+date_features	0	30	0.186	0.176	0.139	0.183
shift_features	10	7	0.177	0.157	0.144	0.209
shift+date_features	10	7	0.178	0.160	0.130	0.217
shift_features	0	7	0.067	0.067	0.064	0.120
shift+date_features	0	7	0.065	0.065	0.062	0.136
percentile_features	10	30	0.136	0.155	0.099	0.227
percentile_features	0	30	0.136	0.136	0.131	0.145
percentile_features	10	7	0.164	0.112	0.146	0.211
percentile_features	0	7	0.091	0.091	0.079	0.097
basic_features	10	30	0.187	0.130	0.159	0.230
basic_features	0	30	0.187	0.130	0.159	0.230
basic_features	10	7	0.177	0.103	0.133	0.213
basic_features	0	7	0.177	0.103	0.136	0.213

proach to time series forecasting, enhancing our ability to capture meaningful patterns and relationships within the data.

Our methodology proves effective by providing insights into the pivotal features and algorithms contributing significantly to the accurate forecasting of Maximum Demand. We incorporate a transformation technique known as a rolling window, a widely-used approach in time series analysis, signal processing, and data analysis. The rolling window's primary purpose is to analyze data within a moving interval of a fixed size. In our implementation, we set a "window size" variable to 10, signifying a 10-day rolling window. This choice aligns with our focus on capturing short- to medium-term trends and patterns in the data, facilitating our analysis and forecasting efforts.

Our approach capitalizes on historical data from preceding days to anticipate the demand for the next day. Our results indicate that this methodology provides a more extensive set of features compared to forecasting the next day based solely on the current day's information. Instead of relying solely on today's data for prediction, we incorporate data from the preceding week (7 days) and the previous month (averaging 30 days). This approach generates a dense matrix of information that enhances the performance of all the algorithms we've employed, leading to significantly more accurate forecasts.

Tables III to VII present the Mean Absolute Percentage Error (MAPE) scores for all the machine learning models in our analysis. These scores are provided for different feature extraction processes. "shift_features" represent shift feature extraction without including date-related features like month and week. "shift+date_features" encompass shift feature extraction with the inclusion of date features. "percentile_features" denote percentile feature extraction, and "basic_features" encom-

TABLE VII: ML model Comparison Results for Linear Interpolation Imputation

Feature Extraction	Window Size	Prev Days Data	Ridge MAPE	Lasso MAPE	GBR MAPE	SVR MAPE
shift_features	10	30	0.185	0.175	0.133	0.183
shift+date_features	10	30	0.185	0.175	0.133	0.183
shift_features	0	30	0.185	0.175	0.133	0.183
shift+date_features	0	30	0.185	0.175	0.133	0.183
shift_features	10	7	0.173	0.152	0.163	0.207
shift+date_features	10	7	0.175	0.154	0.143	0.216
shift_features	0	7	0.061	0.061	0.060	0.119
shift+date_features	0	7	0.060	0.060	0.058	0.134
percentile_features	10	30	0.128	0.155	0.118	0.228
percentile_features	0	30	0.136	0.136	0.137	0.147
percentile_features	10	7	0.162	0.109	0.171	0.211
percentile_features	0	7	0.090	0.090	0.079	0.097
basic_features	10	30	0.185	0.126	0.171	0.228
basic_features	0	30	0.185	0.126	0.171	0.228
basic_features	10	7	0.176	0.103	0.163	0.214
basic_features	0	7	0.176	0.103	0.165	0.214

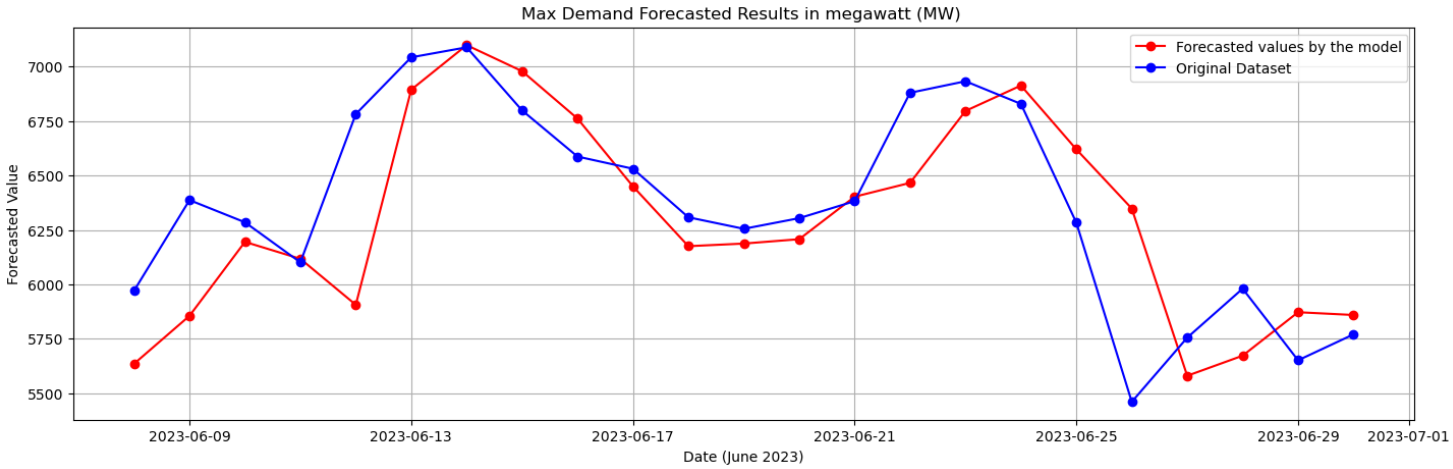

Fig. 4: The Forecasted Values from the Gradient Boosted Regressor with Seven-Day Shift Features, Including Date Features, Trained Using Linear Interpolation Imputation

pass basic feature extraction methods. These tables collectively offer a comprehensive perspective on the performance of our models across various feature extraction strategies. Additionally, Figure 4 provides a visual representation of forecasted values for the upcoming month, June 2023.

VI. Conclusion and Future Work

Based on the above results, we have identified two models with good MAPE scores. For further evaluation, we utilized the Validation dataset from June 2023. The Mean Imputation with 7 days shift, including date features, yielded a MAPE score of 0.040, while the Linear Interpolation Imputation with 7 days shift and date features resulted in a MAPE score of 0.038. Therefore, we selected the Linear Interpolation Imputation technique with 7 days of shift and date features as the Baseline model for our future work. To enhance our forecasting approach, we intend to explore incorporating additional features beyond the univariate analysis. Previous studies in the field have showcased the significance of including a more extensive set of features. Particularly, we are interested in exploring the application of Reinforcement Learning for Model Selection in Time-series Forecasting. We aim to integrate data from various ministries under the PM-Gati-Shakti Scheme in conjunction with the Reinforcement Learning model selection methods to improve our forecasting model further.

Acknowledgment

Authors express their gratitude to the PM-Gati-Shakti Scheme Innovators and all the centers, including the National Institute Of Industrial Engineering, Mumbai, for initiating this program for addressing numerous challenges that India is currently facing.

REFERENCES

[1] GRID CONTROLLER OF INDIA LIMITED, Formerly known as Power System Operation Corporation Limited. https://posoco.in. Accessed: July 2023.

[2] Ministry of Power, Government of India, Power Grid Corporation of India Ltd (POWERGRID). https://www.powergrid.in. Accessed: July 2023.

[3] National Load Despatch Centre, Power System Operation Corporation Limited, 2013. https://report.grid-india.in/index.php?p=Daily+Report, New Delhi, India. Accessed: July 2023.

[4] Government of India. (2021). PM Gati Shakti: National Master Plan for Multi-Modal Connectivity.Retrieved from https://www.india.gov.in/spotlight/pm-gati-shakti-national-master-plan-multi-modal-connectivity. Accessed: July 2023.

[5] Reserve Bank of India (2022). Retrieved from https://www.rbi.org.in/Scripts/publications.aspx. Accessed: July 2023.

[6] Dickey, D. A., & Fuller, W. A. (1979). Distribution of the Estimators for Autoregressive Time Series With a Unit Root. Journal of the American Statistical Association, 74(366), 427–431. https://doi.org/10.2307/2286348

[7] C. E. Velasquez, M. Zocatelli, F. B. G. L. Estanislau, and V. F. Castro, "Analysis of time series models for Brazilian electricity demand forecasting," Energy, vol. 247, pp. 123483, 2022, ISSN: 0360-5442, doi: 10.1016/j.energy.2022.123483, url: https://www.sciencedirect.com/science/article/pii/S0360544222003863.

[8] K. Goswami and A. B. Kandali, "Electricity Demand Prediction using Data Driven Forecasting Scheme: ARIMA and SARIMA for Real-Time Load Data of Assam," in 2020 International Conference on Computational Performance Evaluation (ComPE), 2020, pp. 570-574, doi: 10.1109/ComPE49325.2020.9200031.

[9] S. R. Rallapalli and S. Ghosh, "Forecasting monthly peak demand of electricity in India—A critique," in Energy Policy, vol. 45, pp. 516-520, 2012, ISSN: 0301-4215, doi: 10.1016/j.enpol.2012.02.064,

[10] A. Banga and S. Sharma, "Electricity Demand Forecasting Models at Hourly and Daily Level: A Comparative Study," 2022 International Conference on Advanced Computing Technologies and Applications (ICACTA), Coimbatore, India, 2022, pp. 1-5, doi: 10.1109/ICACTA54488.2022.9753055.

[11] Reddy M, S. and Gopakumar G. 2023 "PM-Gati Shakti: A Case Study of Demand Forecasting" Preprints. https://doi.org/10.20944/preprints202308.0535.v1

[12] M. Uppal, R. Kumari and S. Shrivastava, "An Ensemble Approach for Short-Term Load Forecasting for DISCOMS of Delhi Across the COVID-19 Scenario," 2021 9th IEEE International Conference on Power Systems (ICPS), Kharagpur, India, 2021, pp. 1-6, doi: 10.1109/ICPS52420.2021.9670013.

[13] S. Saravanan, S. Kannan, R. Nithya and C. Thangaraj, "Modeling and prediction of India's electricity demand using fuzzy logic," 2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014], Nagercoil, India, 2014, pp. 93-96, doi: 10.1109/ICCPCT.2014.7054813.

[14] C. L. Athanasiadis, G. Tsoumplekas, A. Chrysopoulos and D. I. Doukas, "Peak demand forecasting: A comparative analysis of state-of-the-art machine learning techniques," 2022 2nd International Conference on Energy Transition in the Mediterranean Area (SyNERGY MED), Thessaloniki, Greece, 2022, pp. 1-6, doi: 10.1109/SyNERGYMED55767.2022.9941434.

[15] Konstantinos Benidis, Syama Sundar Rangapuram, Valentin Flunkert, Yuyang Wang, Danielle Maddix, Caner Turkmen, Jan Gasthaus, Michael Bohlke-Schneider, David Salinas, Lorenzo Stella, François-Xavier Aubet, Laurent Callot, and Tim Januschowski. 2022. Deep Learning for Time Series Forecasting: Tutorial and Literature Survey. ACM Comput. Surv. 55, 6, Article 121 (June 2023), 36 pages. https://doi.org/10.1145/3533382

[16] A. Wibowo, I. Yasmina, A. Wibowo "Food Price Prediction Using Time Series Linear Ridge Regression with The Best Damping Factor", Advances in Science, Technology and Engineering Systems Journal, vol. 6, no. 2, pp. 694-698 (2021).

[17] Jiahan Li, Weiye Chen, Forecasting macroeconomic time series: LASSO-based approaches and their forecast combinations with dynamic factor models, International Journal of Forecasting,Volume 30, Issue 4, 2014, Pages 996-1015, ISSN 0169-2070, https://doi.org/10.1016/j.ijforecast.2014.03.016.

[18] Agapitos, A., Brabazon, A. & O'Neill, M. Regularised gradient boosting for financial time-series modelling. Comput Manag Sci 14, 367–391 (2017). https://doi.org/10.1007/s10287-017-0280-y

[19] Sahoo, B.B., Jha, R., Singh, A. et al. Application of Support Vector Regression for Modeling Low Flow Time Series. KSCE J Civ Eng 23, 923–934 (2019). https://doi.org/10.1007/s12205-018-0128-1

[20] Spiliotis, E., Makridakis, S., Semenoglou, AA. et al. Comparison of statistical and machine learning methods for daily SKU demand forecasting. Oper Res Int J 22, 3037–3061 (2022). https://doi.org/10.1007/s12351-020-00605-2

[21] Sungil Kim, Heeyoung Kim, A new metric of absolute percentage error for intermittent demand forecasts, International Journal of Forecasting, Volume 32, Issue 3, 2016, Pages 669-679, ISSN 0169-2070, https://doi.org/10.1016/j.ijforecast.2015.12.003.

[22] Spyros Makridakis, Evangelos Spiliotis, Vassilios Assimakopoulos, The M4 Competition: 100,000 time series and 61 forecasting methods, International Journal of Forecasting, Volume 36, Issue 1, 2020, Pages 54-74, ISSN 0169-2070, https://doi.org/10.1016/j.ijforecast.2019.04.014.

[23] Spyros Makridakis, Rob J. Hyndman, Fotios Petropoulos, Forecasting in social settings: The state of the art, International Journal of Forecasting, Volume 36, Issue 1, 2020, Pages 15-28, ISSN 0169-2070, https://doi.org/10.1016/j.ijforecast.2019.05.011.

[24] Xenochristou, M., Hutton, C., Hofman, J., & Kapelan, Z. (2020). Water demand forecasting accuracy and influencing factors at different spatial scales using a Gradient Boosting Machine. Water Resources Research, 56, e2019WR026304. https://doi.org/10.1029/2019WR026304

[25] Anil K Pandey, Kishan Bhushan Sahay, M. M Tripathi, and D Chandra. Short-term load forecasting of uppcl using ann. In 2014 6th IEEE Power India International Conference (PIICON), pages 1–6, 2014.

[26] Navneet Kumar Singh, Asheesh Kumar Singh, and Manoj Tripathy. A comparative study of bpnn, rbfnn and elman neural network for short-term electric load forecasting: A case study of delhi region. In 2014 9th International Conference on Industrial and Information Systems (ICIIS), pages 1–6, 2014.

[27] Pavan Kumar Dharmoju, Karthik Yeluripati, Jahnavi Guduri, and Kowstubha Palle. Forecasting electrical demand for the residential sector at the national level using deep learning. In 2021 International Conference on Artificial Intelligence and Machine Vision (AIMV), pages 1–6, 2021.

[28] Shashwat Jha and Vishvaditya Luhach. Indian peak power demand forecasting: Trans- former based implementation of temporal architecture. In 2022 IEEE Global Conference on Computing, Power and Communication Technologies (GlobConPT), pages 1–5, 2022.

[29] Saravanan S, Kannan Subramanian, and C. Thangaraj. India's electricity demand forecast using regression analysis and artificial neural networks based on principal com- ponents. ICTACT Journal on Soft Computing, 2:365–370, 07 2012.

[30] Vishnu Vardhan Sai Lanka, Millend Roy, Shikhar Suman, and Shivam Prajapati. Re- newable energy and demand forecasting in an integrated smart grid. In 2021 Innovations in Energy Management and Renewable Resources(52042), pages 1–6, 2021.