Comparative Analysis of Short-Term Load Forecasting Using Machine Learning Techniques

Hagos L.Shifare[1][0009−0003−1448−9425], Ronak Doshi[2][0009−0002−3937−204X], and Amit Ved[3][0000−0002−5513−4888]

[1] Department of Computer Engineering-DS, Marwadi University, Rajkot, India
hagos.shifare122619@marwadiuniversity.ac.in
[2] Department of Electrical Engineering, Marwadi University, Rajkot, India
ronak.doshi@marwadieducation.edu.in
[3] Department of Electrical Engineering, Marwadi University, Rajkot, India
amit.ved@marwadieducation.edu.in

Abstract. Short-Term Load Forecasting is essential in estimating future energy demand in power systems, energy utilities, and industrial settings. For energy suppliers and other players in the markets for electric energy generation, transmission, and distribution, load forecasting is a crucial instrument. Additionally, the prediction of load is essential for effectively planning and overseeing power system operations. Load forecasting has significant effects on a variety of power system applications, such as energy production, load shedding, contract analysis, and infrastructure construction. In this study, the authors compare and contrast three forecasting methods for short-term load forecasting namely, Gradient Boosting (referred to as GB), Random Forest (referred to as RF), and K-Nearest Neighbors (abbreviated as KNN). The historical load (annual) and New York calendar data are input parameters into the forecasting models for each of these strategies. The research assesses how well these methods perform in terms of Root Mean Square Error (RMSE), Mean Absolute Error, R2-score, and computing time. The comparative analysis finds that KNN is the most effective option, with an amazing R2-score of 98.43%, followed by RF at 97.12% and GB at 95.52%. Furthermore, KNN offers tremendous computing efficiency, emphasizing its suitability for real-time applications. This work lays the door for improved load forecasting in dynamic energy systems by using the capabilities of these models.

Keywords: Short-term load forecasting · Machine learning algorithms · K-Nearest Neighbors (KNN) · Electricity demand estimation

1 Introduction

The electricity sector faces increasing challenges due to the variability of load demand, especially with the growing adoption of renewable energy sources. Load forecasting employs various techniques and methods to predict future electricity demand accurately. Some of the different techniques commonly used for load forecasting [13] include time series analysis, machine learning techniques (like neural networks, decision trees, and regression models), statistical methods, and artificial intelligence-based approaches [9,22].

Time series analysis [1] is a fundamental technique that involves looking at historical load data to spot patterns and trends. To produce predictions, machine learning algorithms use past data and a variety of features, providing flexibility and adaptability. Seasonality and trends are well-captured by statistical techniques like ARIMA (AutoRegressive Integrated Moving Average), which has a long history [28]. The ability of artificial intelligence techniques to manage intricate relationships inside data is helping them gain prominence. Additionally, the integration of deep learning models [26] can enhance the capacity to capture complex temporal dependencies in load forecasting.

Depending on the time frame and level of detail, there are different types of load forecasting. The most common type is short-term load forecasting (STLF), which foresees demand within hours or days and is useful for grid management and day-ahead market activities. The prediction horizon is extended to weeks or months with the help of medium-term load forecasting (MTLF), which is useful for scheduling maintenance and allocating resources. In order to help with decisions about infrastructure construction and capacity growth, long-term load forecasting (LTLF) entails predicting demand years in the future [32,14].

Accurate short-term load forecasting is crucial for ensuring efficient energy distribution, resource allocation, and grid stability. Traditional methods [20,31] and machine learning techniques have been employed to address this challenge. In this research work, we aim to compare the effectiveness of various forecasting methods to identify the most accurate and reliable approach for short-term energy consumption prediction. Load forecasting is vital for optimizing energy use in the electrical power system [6]. It enables efficient management of spinning reserve capacity, improves device repair scheduling, and informs unit commitment decisions. Aligning electrical generation with load demand is crucial for optimal power system performance. The utilization of this study is anticipated to rectify the imbalance between energy supply and demand.

2 Related Work

Numerous studies have addressed short-term load forecasting using both traditional and machine learning methods. Researchers have explored techniques [22,9] like recurrent neural networks (RNNs), K-Nearest Neighbors (KNN) [41], support vector machines (SVM) [29], Random Forest (RF) [34], ensemble methods, artificial neural networks (ANN) [3], convolutional neural networks (CNN) [40], long short-term memory (LSTM) [7], and others. Energy forecasting in this area has long been done using time series models that capture seasonality and trends in past consumption data, for instance, the autoregressive integrated moving average model (ARIMA) [28] or the seasonal autoregressive integrated moving average (SARIMA) [33,24]. ML and DL algorithms are gaining popularity in time series data analysis.SVM has been successfully used to forecast energy consumption in low-energy buildings [29] as well as cooling load in HVAC systems [15]. Strictly choosing pertinent data allowed for increased prediction accuracy [27]. Many researchers have used KNN for forecasting tasks, and the outcomes have shown its capacity to provide precise forecasts. In the context of power system load forecasting, the K-Nearest Neighbors (KNN) model [4] has demonstrated superior performance, surpassing polynomial and sinusoidal regressions. Additional research findings are showcased in [5,41,19]. These studies have demonstrated the importance of accurate load forecasting for efficient energy management. Nonetheless, as load forecasting continues to evolve, challenges persist in effectively addressing Real-time data and handling complex relationships.

3 Dataset description and Pre-Processing

The dataset employed in this study serves as the foundation for training and testing the predictive models, and the analysis was conducted using Python version 3.11.0. Electrical energy demand information from 1st Jan 2020 to 31st Dec 2020, one-year real-time load data is gathered from NYISO. It encompasses a comprehensive collection of 107,000 entries, representing a diverse array of load demand instances. The recorded data set is in a five-minute time interval. The dataset is segregated into two principal components: a time series feature and the corresponding target variable, which encapsulates the actual load demand. In line with standard data partitioning practices, the majority of the dataset, comprising 80%, was used for model training, with the remainder allotted for subsequent testing.

Visualization of Load Fluctuations: Insights from the Dataset The dataset visualization provided in Figures 1 and 2 offers insightful information about the variations in annual load. Figure 1 shows the load variations over the course of a year, with each color representing a specific month. The y-axis denotes the load in megawatts (MW), while the x-axis represents time in months. Figure 2, a Heatmap, provides a more detailed view of the monthly load dispersion in megawatts (MW) for each month and day of the entire year. The horizontal axis depicts the months of the year, while the vertical axis represents the days of the month. The color scale is provided on the heat map's right-hand side. The color of each cell represents the magnitude of the load demand for a particular month. The deeper the hue, the greater the load demand, while the paler the color, the lower the load demand.

During July, the deep blue coloring indicates a significant load dispersion. This marks the height of the summer season, characterized by heightened energy consumption due to increased reliance on cooling systems and other appliances. In contrast, the pale yellow shades in April and May suggest a more consistent and stable load demand. During these months, energy consumption tends to exhibit less fluctuation, aligning with the milder conditions of the spring season. January to April and July to September display gradient shifts, with deepening blues signifying increasing load demand during seasonal transitions. These shifts involve heating and cooling adjustments.

Finally, from November to December, as winter approaches, the deepening blue shades reveal an uptick in load demand. This trend corresponds to the increasing need for heating systems to maintain comfortable indoor temperatures.

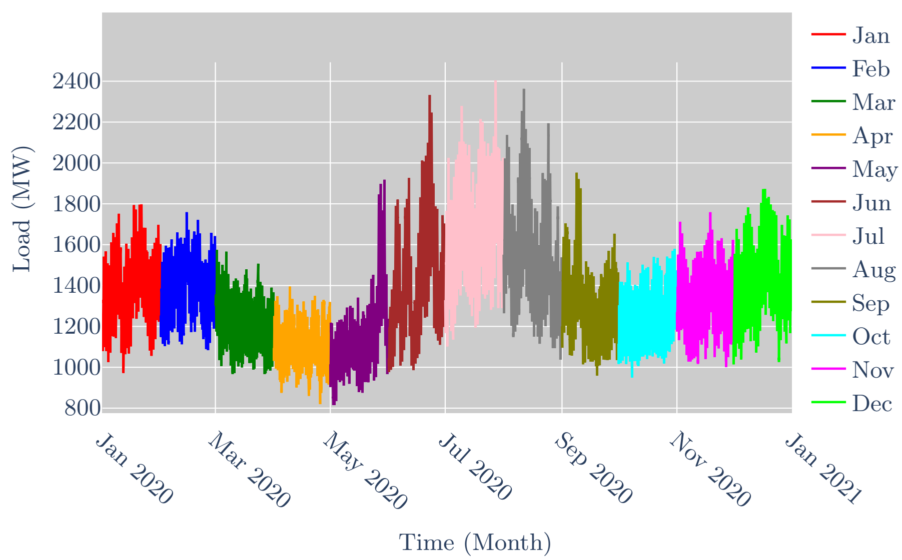

Fig. 1. Load fluctuation over one year period

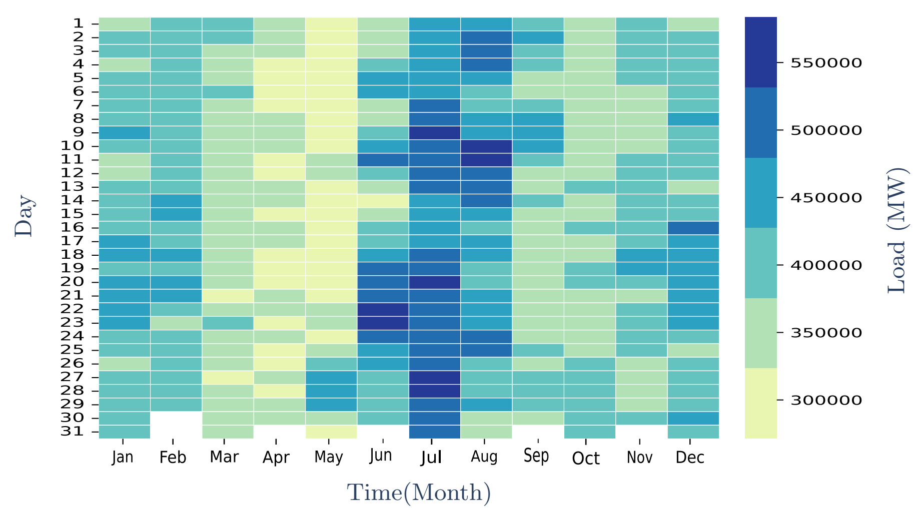

Fig. 2. Monthly load dispersion in Heatmap

Data preparation: The preparation of the dataset involved a series of meticulous data preprocessing steps, crucial for ensuring the quality and reliability of subsequent model predictions. Each approach, in particular, underwent unique data preprocessing methods adapted to its particular needs.

Features and Target: The dataset is distinguished by two key components: the feature and the target. The single time series feature incorporates temporal information about the load demand instances, acting as the foundation for prediction models. Concurrently, the target variable represents the actual load demand values, serving as the standard against which model predictions are compared.

Null Value Handling: Finding and addressing any null values that were present in the dataset was one of the steps. Only two instances of the 107,000 records had null values. These null values were seamlessly substituted using the interpolation method, maintaining the dataset's integrity.

Normalization: The dataset was normalized to provide uniform scale and convergence during model training. In order to prevent the dominance of certain features in the modeling process, this technique ensured that all features shared similar scale properties.

4 Methodology

To address the issues provided by dynamic load demand, a plethora of solutions in the domain of Short-Term Load Forecasting (STLF) have arisen. In our dataset, we evaluated six models. Nonetheless, given to their greater predictive performance, KNN, RF, and GB were chosen for comparison and study. Some major theoretical concepts and parameters used in load forecasting analysis are covered for each technique. Table 1 describes the input parameters under consideration.

4.1 Gradient Boosting (GB) Model

Gradient Boosting [27] is an effective boosting technique that transforms multiple weak learners into strong ones. Each subsequent model is trained using gradient descent [25] to minimize the loss function of the previous model, which can be metrics like mean square error or cross-entropy[12]. The name "Gradient Boosting" comes from the fact that the technique is focused on minimizing a loss function [11].

A description of the operation of gradient boosting for regression Gradient boosting Regressor [37] starts with the setup of a fundamental model, frequently a simple decision tree referred to as a "weak learner" [39]. The model is then used to compute residuals, which indicate prediction mistakes. These residuals direct the training of a new weak learner, which seeks to identify patterns or mistakes that have not yet been addressed [27]. The outputs of the new learner are added to the predictions made by the current model after being weighted by a small learning rate. Multiple weak learners participate in this iterative process, which emphasizes residual minimization and goes on until a predetermined criterion is reached. This leads to less bias and better predictive accuracy as contributions from all poor learners are combined for the final prediction. To understand Gradient Boosting for regression better, refer to [23,17,27].

4.2 Random Forest (RF) Model

Random forests represent a significant enhancement of bagging by constructing a sizable ensemble of trees that are uncorrelated with each other, followed by averaging their predictions [34]. The main concept of bagging is to reduce variance by averaging several noisy but roughly unbiased base models. Hence, it is an ensemble machine-learning approach that combines different decision trees to increase prediction accuracy and decrease overfitting. When compared to individual decision trees, RF is shown to have better prediction performance and robustness. It doesn't experience the commonly known issues found in individual trees, such as unstable divisions and a lack of smoothness[42].RF can be employed for classification as well as regression. We advise studying references [34,42,35] for a more thorough understanding of the inner workings of the RF algorithm.

In this study, we used Random Forest for regression analysis, concentrating on its use for Short-Term Load Forecasting (STLF). In the following flow chart 3, we illustrate the Random Forest (RF) process, which incorporates an essential technique known as 'bagging.' The term Bagging originates from the acronym formed by Bootstrap Aggregating [42]. As its name suggests, Bagging primarily consists of two essential elements: bootstrap and aggregation. Bootstrapping [23,17] involves the random sampling of the training dataset with replacement to create multiple subsets (bootstrap samples). These subsets are used to train individual decision trees within the Random Forest. In order to provide a more reliable and accurate overall prediction, many decision trees' predictions are combined in a process known as "aggregation." Aggregation [30,16] enhances the model's capability by lowering the variance of the predictions made by each individual decision tree in the RF, which each makes its own forecast based on the training data. Furthermore, the incorporation of Bagging in Random Forest minimizes the risk of overfitting [36], as each decision tree is trained on a different subset of the data. This diversity in training improves the model's generalization to unseen data, contributing to its robust performance in Short-Term Load Forecasting. The effectiveness of the Random Forest approach lies in its ability to capture complex relationships in the data, making it a powerful tool for accurate and reliable predictions in energy forecasting applications.

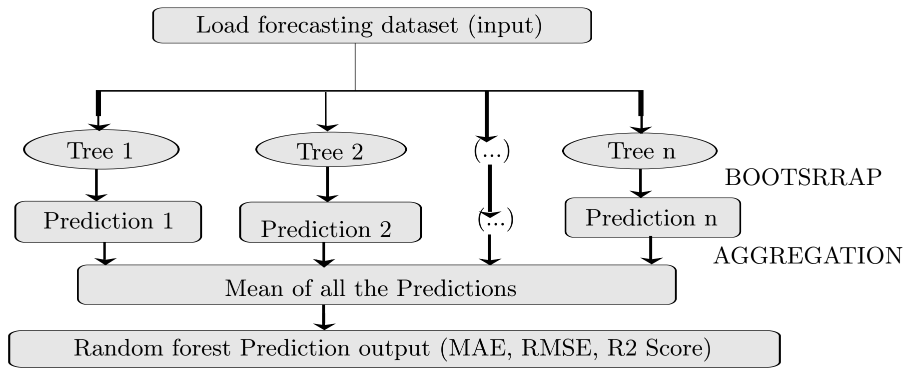

Fig. 3. Flow chart of RF algorithm

4.3 K-Nearest Neighbors (KNN) Model

Nearest Neighbor algorithms are among the simplest of all machine learning algorithms. K-Nearest Neighbors (KNN) [41] regression is a non-parametric [5], instance-based learning algorithm used for predictive modeling and regression analysis. It is a simple yet effective technique used for short-term load forecasting (STLF) due to its intuitive nature and adaptability to various data patterns. The method searches for the 'nearest neighbors,' which are the data points closest to the new data point [30] in the training dataset. Predicting the target value for a fresh data point involves averaging the target values [21] of its k closest neighbors in the training dataset. The general KNN regression formula is:

$$\hat{y} = \frac{1}{k} \sum_{i=1}^{k} y_i \tag{1}$$

Where:

$\hat{y}$ is the predicted target value for the new data point.

k is the number of closest neighbors to consider.

y_i represents the target values of the k-nearest neighbors.

A distance function determines how close two points are to one another.This measure is crucial in KNN algorithms, where it helps identify the nearest neighbors of a given data point, enabling effective classification and regression tasks. The subsequent equation is employed for calculating the Euclidean distance between two data points, X and Y, in a multidimensional space [38].

$$\text{Euclidean Distance}(D_i) = \sqrt{\sum_{i=1}^{n}(X_i - Y_i)^2} \tag{2}$$

Where:

n is the count of dimensions (attributes) in the data.

X_i and Y_i denote the values of the i-th dimension for data points X and Y respectively.

The formulas provided above are the core mechanism behind its predictive power and make it a valuable tool in data-driven decision-making for load forecasting. For load forecasting, KNN analyzes the similarity between the input time series and past occurrences to forecast future load demand. For a more in-depth understanding of how the KNN algorithm operates we encourage readers to refer to [30,38]. The workflow of a K-Nearest Neighbors (KNN) regression model for load forecasting is illustrated visually in figure 4. It outlines the steps involved in preparing data, setting up a model, training it, making a prediction, evaluating it, and reporting evaluation metrics. With its capacity to unearth hidden patterns and adapt to shifting load demand circumstances, KNN regression gives a significant edge in load forecasting. It thrives in capturing the complex interactions between the input time series and the historical data, producing precise forecasts. Furthermore, the model's interpretability and simplicity [18] make it an attractive choice for practitioners seeking transparent insights into the load forecasting process. Additionally, its non-parametric nature allows for flexibility in handling diverse datasets. It has applications in areas including recommendation systems, anomaly detection, and pattern recognition, demonstrating its adaptability beyond load forecasting [2].

In the course of this research, a rigorous dataset preprocessing methodology is implemented. This involves the conversion of the date-time feature into a numerical representation and the normalization of both features and the target variable through Min-Max scaling, preventing bias toward variables with larger magnitudes and fostering improved model performance. The dataset is subsequently partitioned into distinct training and test sets, a standard practice in machine learning evaluation. For model development, we employ the K-Nearest Neighbors (KNN) algorithm with a configuration of twenty-four nearest neighbors, selected through hyperparameter tuning. The careful selection of hyperparameters, guided by tuning, enhances the K-Nearest Neighbors algorithm's performance, optimizing its ability to capture intricate patterns in the data. Post-training, a crucial step involves reverse-transforming the predicted values to their original scale, facilitating a more meaningful interpretation of the results in the context of the original dataset. This comprehensive approach, spanning data preprocessing, model training, and result interpretation, serves as the foundation for a robust and reliable comparative analysis in this research endeavor.

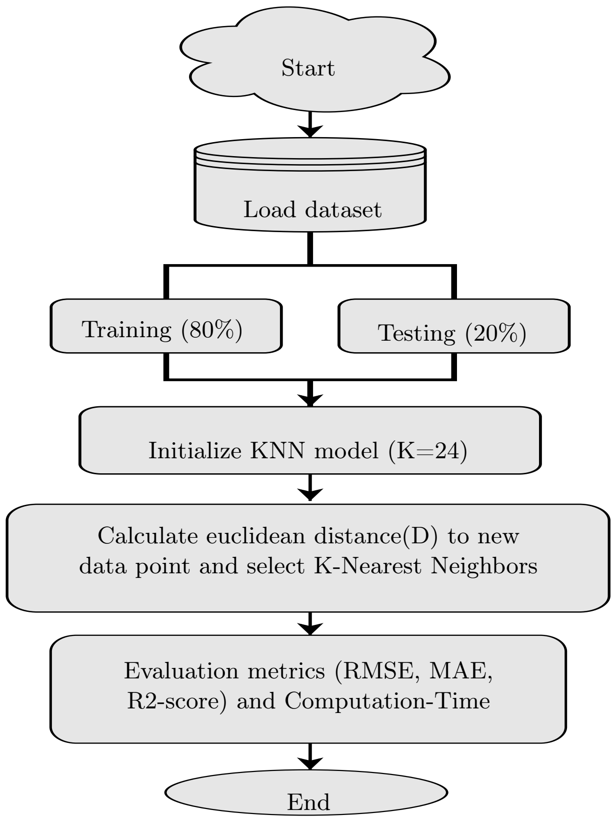

Fig. 4. Flow chart of KNN algorithm

Table 1. Model Parameters

Model	Parameters
Gradient Boosting (GB)	`Number of estimators=100, Maximum depth=11, Random state=42`
Random Forest (RF)	`Number of estimators=100, Maximum depth=16, Random state=42`
K-Nearest Neighbors (KNN)	`Number of neighbors=24, p=3, Random state=42`

5 Result and Discussion

5.1 Assessment Measures

To evaluate the model's precision in regression analysis, the metrics Mean Squared Error, Root Mean Square Error, Mean Absolute Error, and R2-score or Coefficient of Determination are used [8]. The mathematical expression of the metrics is provided as follows where, $\hat{y}$ represents the anticipated or forecasted value of y, and $\bar{y}$ denotes the average or mean value of y.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2 \tag{3}$$

MSE (Mean Squared Error) [10] quantifies this difference by averaging the squared values.

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2} \tag{4}$$

RMSE (Root Mean Squared Error) [8] is the square root of MSE, providing another measure of error.

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}| \tag{5}$$

MAE (Mean Absolute Error) [10] quantifies the average absolute difference between predicted and actual values.

$$R2 - score = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2} \tag{6}$$

R2-score (Coefficient of Determination) gauges how well the model fits the data, with values between 0 and 1, higher values indicate a better fit [10,38].

5.2 Experimental Result Analysis

The performance of our three models - GB, RF, and KNN - in short-term load forecasting was evaluated using four key metrics: RMSE, MAE, R2-score, and Computation-Time. Predictions are made for all months and for one random month (November). The models underwent training on a dataset through the use of 5-fold cross-validation and were evaluated using NMSE metric to prevent overfitting or underfitting. Table 2 summarizes the performance metrics, and the subsequent figures visually compare actual and predicted load consumption. GB exhibited the weakest fit to actual load consumption, as the red and blue lines shown in figure 5 and 6 consistently diverged. This stark divergence accentuated GB's lowest R2-score (0.9552), reflecting low performance in comparison to the other models.GB is also taking more computational time, specifically 34.5761 seconds.

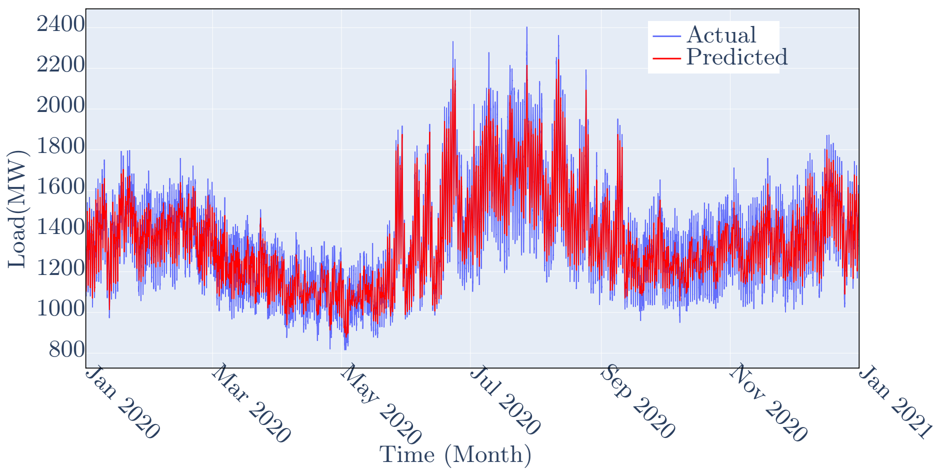

Fig. 5. Actual vs predicted load consumption using GB model

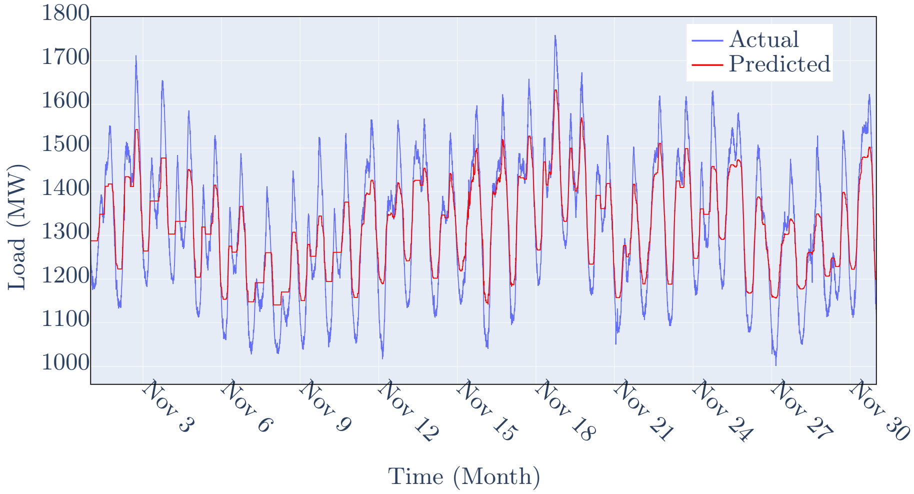

Fig. 6. Actual vs predicted load for one month with GB model

RF, though not matching KNN's performance, demonstrated a commendable fit to real load usage, with a reasonably low RMSE (0.0263), MAE (0.0161), and a good R2-score of 0.9712. The graph in figure 7 and 8 indicates that while RF has some inconsistencies and variations, the blue and red lines generally remain close. However, RF displayed higher fluctuations in predicted values compared to KNN, indicating slightly lower stability. It should be noted that RF required more computational time, at 29.4186 seconds.While RF produced favorable results, particularly in certain months, it displayed inconsistency in the summer months.

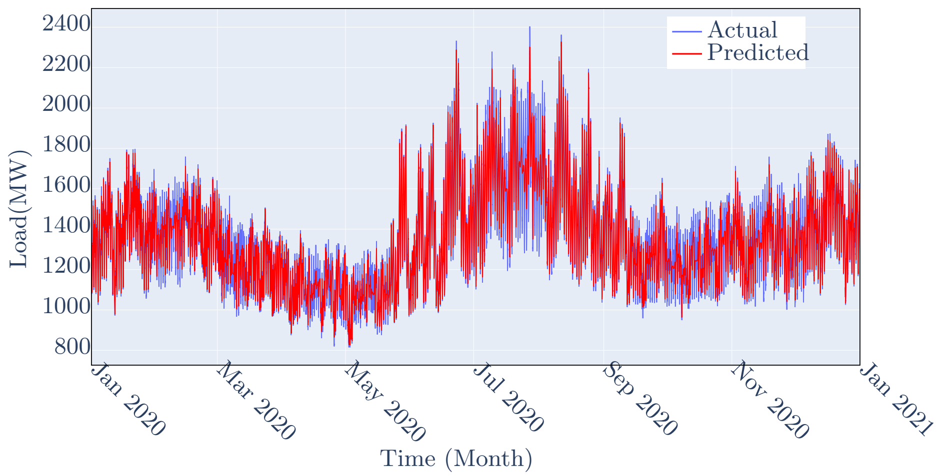

Fig. 7. Actual vs predicted load consumption using RF model

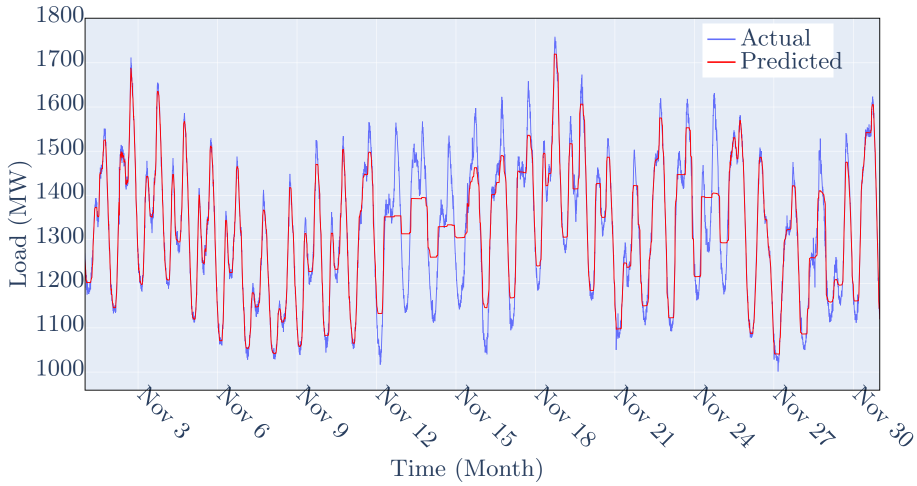

Fig. 8. Actual vs predicted load for one month with RF model

KNN emerged as the top-performing model, boasting the lowest RMSE (0.0194), MAE (0.0156), and the highest R2-score (0.9843). The close alignment of the red line (predicted) and the blue line (actual) in the graph 9 and 10 demonstrates its outstanding accuracy and precision, as well as its stability, with minimal predicted value variations. Additionally, KNN exhibited remarkable computational efficiency, taking only 0.0637 seconds. This impressive combination of RMSE, MAE, R2-score, and Computation-Time makes KNN an ideal choice for our predictive modeling needs.

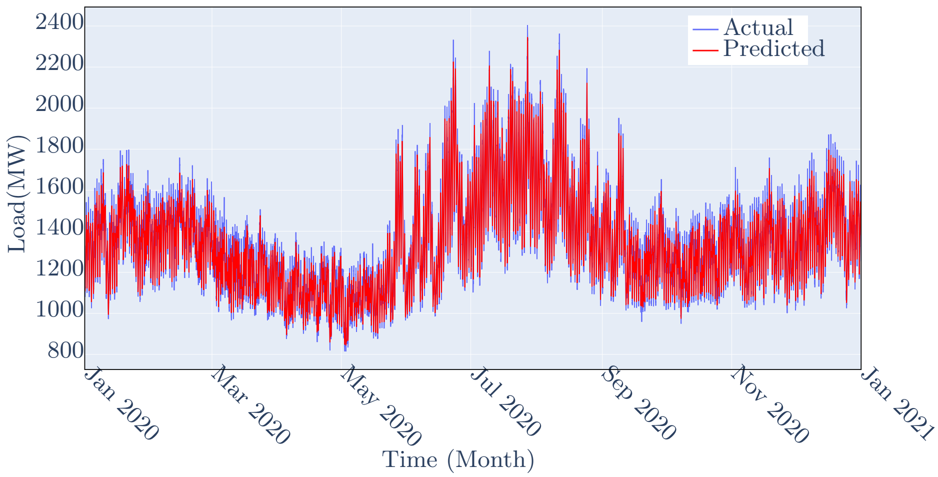

Fig. 9. Actual vs predicted load consumption using KNN model

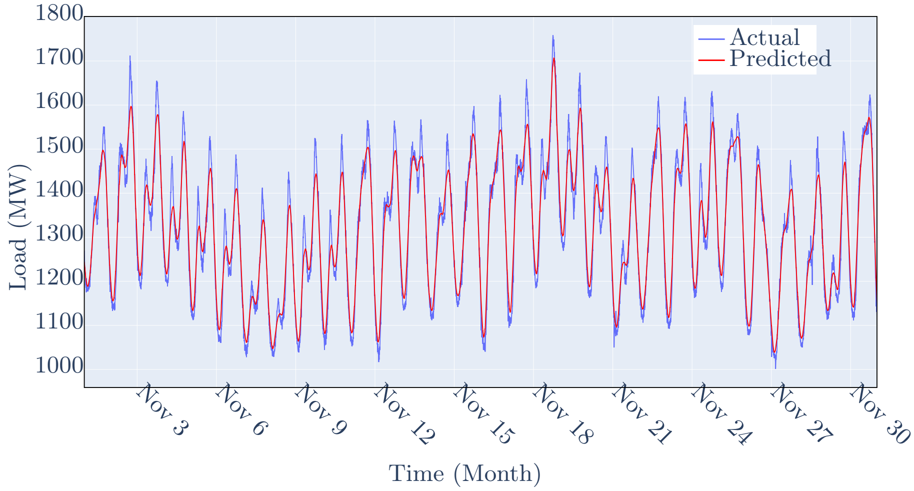

Fig. 10. Actual vs predicted load for one month with KNN model

Table 2. Performance Comparison of Machine Learning Models

Model	RMSE	MAE	R2-score	Computation-Time (sec)
GB	0.0328	0.0249	0.9552	34.5761
RF	0.02633	0.0161	0.9712	29.4186
KNN	0.0194	0.0156	0.9843	0.0637

6 Conclusion

This study aims to forecast electricity demand using machine learning algorithms, with a primary focus on achieving the highest possible R2-score while maintaining computational time as a crucial consideration in the prediction process. The dataset employed for our analysis was sourced from NYISO, comprising Real-Time Consumption Data. In this research project, we utilized three machine learning models: Gradient Boosting (GB), Random Forest, and K-Nearest Neighbors (KNN). The optimal method has been determined through a rigorous analysis of various statistical parameters, Incorporating Root Mean Square Error (RMSE), Mean Absolute Error (MAE), R2-score, and the time taken for computation. During the summer season, Random Forest (RF) showed limited performance, while Gradient Boosting (GB) exhibited underperformance during the fall. In contrast, K-Nearest Neighbors (KNN) consistently outperformed both RF and GB, delivering superior results with an impressive R2-score of 98.43% across all months, alongside shorter computational times. This study highlights the potential of KNN for enhancing load forecasting in real-time applications, offering valuable insights for the energy sector's future forecasting endeavors.

In forthcoming studies, we will broaden the scope of our forecasting models by expanding their applicability. This will involve enhancing our existing models and exploring additional features, such as temperature, to achieve greater accuracy in real-time predictions. Additionally, we plan to extend our models to forecast PV (photovoltaic) power and wind power, enabling a comprehensive and integrated approach to energy demand and supply forecasting.

Acknowledgement The authors thank NYISO for granting the scientific community's use of the load data for research purposes. Additionally, in order to support such a sophisticated computational study, the authors would like to thank Marwadi University, Rajkot for its computing resources.

References

1. Alqatawna, A., Abu-Salih, B., Obeid, N., Almiani, M.: Incorporating time-series forecasting techniques to predict logistics companies' staffing needs and order volume. Computation **11**(7), 141 (2023). https://doi.org/https://doi.org/10.3390/computation11070141
2. Anwar, T., Uma, V., Hussain, M.I., Pantula, M.: Collaborative filtering and knn based recommendation to overcome cold start and sparsity issues: A comparative analysis. Multimedia Tools and Applications **81**(25), 35693–35711 (2022). https://doi.org/10.1007/s11042-021-11883-z
3. Arvanitidis, A.I., Bargiotas, D., Daskalopulu, A., Laitsos, V.M., Tsoukalas, L.H.: Enhanced short-term load forecasting using artificial neural networks. Energies **14**(22), 7788 (2021). https://doi.org/https://doi.org/10.3390/en14227788
4. Ashfaq, T., Javaid, N.: Short-term electricity load and price forecasting using enhanced knn. In: 2019 International Conference on Frontiers of Information Technology (FIT). pp. 266–2665. IEEE (2019). https://doi.org/https://doi.org/10.1109/FIT47737.2019.00057

5. Atanasovski, M., Kostov, M., Arapinoski, B., Spirovski, M.: K-nearest neighbor regression for forecasting electricity demand. In: 2020 55th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST). pp. 110–113. IEEE (2020)
6. Azeem, A., Ismail, I., Jameel, S.M., Harindran, V.R.: Electrical load forecasting models for different generation modalities: a review. IEEE Access **9**, 142239–142263 (2021). https://doi.org/10.1109/ACCESS.2021.3120731
7. Chen, Z., Zhang, D., Jiang, H., Wang, L., Chen, Y., Xiao, Y., Liu, J., Zhang, Y., Li, M.: Load forecasting based on lstm neural network and applicable to loads of "replacement of coal with electricity". Journal of Electrical Engineering & Technology **16**(5), 2333–2342 (2021). https://doi.org/https://doi.org/10.1007/s42835-021-00768-8
8. Chicco, D., Warrens, M.J., Jurman, G.: The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. PeerJ Computer Science **7**, e623 (2021). https://doi.org/https://doi.org/10.7717/peerj-cs.623
9. Cordeiro-Costas, M., Villanueva, D., Eguía-Oller, P., Martínez-Comesaña, M., Ramos, S.: Load forecasting with machine learning and deep learning methods. Applied Sciences **13**(13), 7933 (2023). https://doi.org/https://doi.org/10.3390/app13137933
10. DataTechNotes: Regression model accuracy (mae, mse, rmse, r-squared) check in r, https://www.datatechnotes.com/2019/02/regression-model-accuracy-mae-mse-rmse.html
11. Di Persio, L., Fraccarolo, N.: Energy consumption forecasts by gradient boosting regression trees. Mathematics **11**(5), 1068 (2023). https://doi.org/https://doi.org/10.3390/math11051068
12. GeeksforGeeks: Gradient boosting in ml, https://www.geeksforgeeks.org/ml-gradient-boosting/
13. Habbak, H., Mahmoud, M., Metwally, K., Fouda, M.M., Ibrahem, M.I.: Load forecasting techniques and their applications in smart grids. Energies **16**(3), 1480 (2023). https://doi.org/https://doi.org/10.3390/en16031480
14. Hammad, M.A., Jereb, B., Rosi, B., Dragan, D., et al.: Methods and models for electric load forecasting: a comprehensive review. Logist. Sustain. Transp **11**(1), 51–76 (2020). https://doi.org/10.2478/jlst-2020-0004
15. Hu, L., Zhang, L., Wang, T., Li, K.: Short-term load forecasting based on support vector regression considering cooling load in summer. In: 2020 Chinese Control And Decision Conference (CCDC). pp. 5495–5498. IEEE (2020). https://doi.org/10.1109/CCDC49329.2020.9164387
16. Ibrahim, B., Rabelo, L., Gutierrez-Franco, E., Clavijo-Buritica, N.: Machine learning for short-term load forecasting in smart grids. Energies **15**(21), 8079 (2022). https://doi.org/https://doi.org/10.3390/en15218079
17. Lee, T.H., Ullah, A., Wang, R.: Bootstrap aggregating and random forest. Macroeconomic forecasting in the era of big data: Theory and practice pp. 389–429 (2020)
18. Li, F., Jin, G.: Research on power energy load forecasting method based on knn. International Journal of Ambient Energy **43**(1), 946–951 (2022). https://doi.org/https://doi.org/10.1080/01430750.2019.1682041
19. Lv, X., Cheng, X., Tang, Y.m., et al.: Short-term power load forecasting based on balanced knn. In: IOP Conference series: materials science and engineering. vol. 322, p. 072058. IOP Publishing (2018). https://doi.org/10.1088/1757-899X/322/7/072058

20. Marrero, L., García-Santander, L., Carrizo, D., Ulloa, F.: An application of load forecasting based on arima models and particle swarm optimization. In: 2019 11th International Symposium on Advanced Topics in Electrical Engineering (ATEE). pp. 1–6. IEEE (2019). https://doi.org/https://doi.org/10.1109/ATEE.2019.8724891
21. Martínez, F., Frías, M.P., Pérez, M.D., Rivera, A.J.: A methodology for applying k-nearest neighbor to time series forecasting. Artificial Intelligence Review **52**(3), 2019–2037 (2019). https://doi.org/https://doi.org/10.1007/s10462-017-9593-z
22. Mediavilla, M.A., Dietrich, F., Palm, D.: Review and analysis of artificial intelligence methods for demand forecasting in supply chain management. Procedia CIRP **107**, 1126–1131 (2022). https://doi.org/https://doi.org/10.1016/j.procir.2022.05.119
23. Mienye, I.D., Sun, Y.: A survey of ensemble learning: Concepts, algorithms, applications, and prospects. IEEE Access **10**, 99129–99149 (2022). https://doi.org/10.1109/ACCESS.2022.3207287
24. Musbah, H., El-Hawary, M.: Sarima model forecasting of short-term electrical load data augmented by fast fourier transform seasonality detection. In: 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE). pp. 1–4. IEEE (2019). https://doi.org/https://doi.org/10.1109/CCECE.2019.8861542
25. Mustapha, A., Mohamed, L., Ali, K.: An overview of gradient descent algorithm optimization in machine learning: Application in the ophthalmology field. In: Smart Applications and Data Analysis: Third International Conference, SADASC 2020, Marrakesh, Morocco, June 25–26, 2020, Proceedings 3. pp. 349–359. Springer (2020). https://doi.org/https://doi.org/10.1007/978-3-030-45183-7_27
26. Neeraj, Gupta, P., Tomar, A.: Deep learning techniques for load forecasting. In: Prediction Techniques for Renewable Energy Generation and Load Demand Forecasting, pp. 177–198. Springer (2023). https://doi.org/https://doi.org/10.1007
27. Nie, P., Roccotelli, M., Fanti, M.P., Ming, Z., Li, Z.: Prediction of home energy consumption based on gradient boosting regression tree. Energy Reports **7**, 1246–1255 (2021). https://doi.org/https://doi.org/10.1016/j.egyr.2021.02.006
28. Pasari, S., Shah, A.: Time series auto-regressive integrated moving average model for renewable energy forecasting. In: Enhancing Future Skills and Entrepreneurship: 3rd Indo-German Conference on Sustainability in Engineering. pp. 71–77. Springer International Publishing Cham (2020). https://doi.org/https://doi.org/10.1007/978-3-030-44248-4_7
29. Paudel, S., Elmitri, M., Couturier, S., Nguyen, P.H., Kamphuis, R., Lacarrière, B., Le Corre, O.: A relevant data selection method for energy consumption prediction of low energy building based on support vector machine. Energy and Buildings **138**, 240–256 (2017). https://doi.org/https://doi.org/10.1016/j.enbuild.2016.11.009
30. Ruiz-Abellón, M.D.C., Gabaldón, A., Guillamón, A.: Load forecasting for a campus university using ensemble methods based on regression trees. Energies **11**(8), 2038 (2018). https://doi.org/https://doi.org/10.3390/en11082038
31. Salehimehr, S., Taheri, B., Sedighizadeh, M.: Short-term load forecasting in smart grids using artificial intelligence methods: A survey. The Journal of Engineering **2022**(12), 1133–1142 (2022). https://doi.org/https://doi.org/10.1049/tje2.12183
32. Sallam, A., Malik, O.: Load Forecasting, pp. 41–71 (10 2018). https://doi.org/10.1002/9781119509332.ch4

33. Santos, M.L., García, S.D., García-Santiago, X., Ogando-Martínez, A., Camarero, F.E., Gil, G.B., Ortega, P.C.: Deep learning and transfer learning techniques applied to short-term load forecasting of data-poor buildings in local energy communities. Energy and Buildings **292**, 113164 (2023). https://doi.org/https://doi.org/10.1016/j.enbuild.2023.113164
34. Sheppard, C.: Tree-based machine learning algorithms: Decision trees, random forests, and boosting. Clinton Sheppard (2019)
35. Silva, J., Praça, I., Pinto, T., Vale, Z.: Energy consumption forecasting using ensemble learning algorithms. In: Distributed Computing and Artificial Intelligence, 16th International Conference, Special Sessions. pp. 5–13. Springer (2020). https://doi.org/https://doi.org/10.1007/978-3-030-23946-6_1
36. Subbiah, S.S., Chinnappan, J.: An improved short term load forecasting with ranker based feature selection technique. Journal of Intelligent & Fuzzy Systems **39**(5), 6783–6800 (2020). https://doi.org/10.3233/JIFS-191568
37. Sundaram, R.B.: Gradient boosting algorithm: A complete guide for beginners. analyticsvidhya (2021)
38. Suwanda, R., Syahputra, Z., Zamzami, E.M.: Analysis of euclidean distance and manhattan distance in the k-means algorithm for variations number of centroid k. In: Journal of Physics: Conference Series. vol. 1566, p. 012058. IOP Publishing (2020). https://doi.org/10.1088/1742-6596/1566/1/012058
39. Touzani, S., Granderson, J., Fernandes, S.: Gradient boosting machine for modeling the energy consumption of commercial buildings. Energy and Buildings **158**, 1533–1543 (2018). https://doi.org/https://doi.org/10.1016/j.enbuild.2017.11.039
40. Tudose, A.M., Picioroaga, I.I., Sidea, D.O., Bulac, C., Boicea, V.A.: Short-term load forecasting using convolutional neural networks in covid-19 context: the romanian case study. Energies **14**(13), 4046 (2021). https://doi.org/https://doi.org/10.3390/en14134046
41. Valgaev, O., Kupzog, F., Schmeck, H.: Building power demand forecasting using k-nearest neighbours model–practical application in smart city demo aspern project. CIRED-Open Access Proceedings Journal **2017**(1), 1601–1604 (2017)
42. Yiling, H., Shaofeng, H.: A short-term load forecasting model based on improved random forest algorithm. In: 2020 7th International Forum on Electrical Engineering and Automation (IFEEA). pp. 928–931. IEEE (2020). https://doi.org/10.1109/IFEEA51475.2020.00195