ADAPTIVE ELECTRICAL LOAD FORECASTING FOR SMART GRIDS USING RANDOM FOREST REGRESSION WITH PERIODIC RETRAINING

Mr. JAGANNATH SATAPATHY
B. Tech Scholar
Department of Electrical and Electronics Engineering
Roland Institute of Technology, Berhampur

Mr. SHOBHAN KUMAR PATRO
Assistant Professor
Department of Electrical and Electronics Engineering
Roland Institute of Technology, Berhampur

Mr. S SUMAN PATRA
B. Tech Scholar
Department of Electrical and Electronics Engineering
Roland Institute of Technology, Berhampur

Mr. PRIYANKAR SWAIN
B. Tech Scholar
Department of Electrical and Electronics Engineering
Roland Institute of Technology, Berhampur

Mr. KANHA PANIGRAHY
B. Tech Scholar
Department of Electrical and Electronics Engineering
Roland Institute of Technology, Berhampur

Abstract*— Accurate electrical load forecasting supports efficient energy management and reliable smart-grid operation, but forecasting models may require updating as consumption patterns change. This study presents an adaptive load forecasting framework based on Random Forest regression with periodic retraining. Temperature, humidity, hour of the day, and historical demand from the preceding hour, day, and week are used as predictive inputs. The dataset is cleaned, arranged chronologically, and divided into training and testing subsets containing 80% and 20% of the observations, respectively. A static model is compared with an adaptive model that is retrained after every 100 testing observations using the latest 1,000 available records. Forecasting performance is assessed using mean absolute error, mean squared error, root-mean-square error, and mean absolute percentage error. The adaptive model achieves a mean absolute error of 2.339 kW and a mean absolute percentage error of 3.383%, compared with 2.369 kW and 3.456% for the static model. These results indicate modest reductions in average absolute and percentage errors through periodic updating. Feature-importance analysis identifies demand from the corresponding hour of the preceding week as the most influential predictor. The findings support periodic retraining as a practical approach to improving Random Forest load forecasts within the evaluated dataset, with potential applications in smart-grid energy management and demand scheduling.

Keywords— Adaptive load forecasting; smart grids; Random Forest regression; periodic retraining; historical demand; energy management.

I. INTRODUCTION

Accurate electrical load forecasting is essential for the reliable and economical operation of power systems. Anticipating electricity demand supports generation scheduling, reserve allocation, energy procurement, and the coordination of supply with consumption. Forecasting errors can lead to inefficient resource allocation, unnecessary operating expenditure, and difficulties in managing peak demand. Within smart-grid environments, demand forecasts also inform energy-storage scheduling and demand-side management. Consequently, forecasting models must represent both recurring consumption patterns and variations arising from weather and operating conditions. The suitability of a forecasting approach depends on the characteristics of the demand being predicted, the available information, and the intended forecasting horizon.[1][2]

Electrical demand exhibits temporal dependencies at multiple scales. Recent consumption reflects the immediate operating state, while observations from corresponding hours on preceding days and weeks provide information about recurring activity patterns. Weather conditions introduce additional variability, particularly where heating, cooling, and ventilation contribute substantially to electricity consumption. These influences interact, making the relationship between demand and its predictors more complex than a simple continuation of historical trends. Previous forecasting research has therefore incorporated calendar information, lagged demand, and temperature observations to represent the seasonal and weather-sensitive behaviour of electricity consumption.[3]

Statistical and machine-learning methods offer different approaches to modelling these relationships. Regression and time-series techniques provide established frameworks for representing trends, seasonality, and dependence on external factors. Flexible statistical models can also capture nonlinear effects without abandoning an interpretable structure. Machine-learning approaches extend the range of relationships that can be learned from historical observations, including interactions that may be difficult to specify explicitly. Recurrent neural networks, for example, have been investigated for residential load forecasting because of their ability to represent sequential dependencies. Nevertheless, greater model complexity does not by itself ensure better forecasting performance; model selection must remain sensitive to data quality, sample availability, and evaluation design.[4][5][6]

Random Forest regression provides an alternative for learning nonlinear relationships from structured electrical-demand data. By aggregating predictions from multiple decision trees, the method can represent interactions between historical consumption, weather conditions, and time-related information. Its ensemble structure reduces dependence on a single fitted tree and provides estimates of feature importance that can assist in examining model behaviour. These properties make it a suitable candidate for investigating load forecasting with a comparatively compact set of measured and derived inputs. However, feature-importance estimates require careful interpretation when historical demand inputs are strongly correlated.[7][8]

An additional consideration is whether a forecasting model should remain fixed after its initial training. A static model retains the relationships learned from its original training period, even as new observations become available. Where consumption behaviour changes, periodically updating the model offers a means of incorporating more recent information. The value of recent information has been examined in load-forecasting research, including the representation of recency effects in weather-related predictors. However, the benefit of retraining must be established through an explicit comparison with a static counterpart, since updating can introduce computational overhead without necessarily improving every aspect of forecast accuracy.[9]

Motivated by these considerations, this study develops an adaptive electrical load forecasting framework based on Random Forest regression. The framework combines temperature, humidity, hour-of-day information, and historical demand observations from the preceding hour, the corresponding hour of the previous day, and the corresponding hour of the previous week. These inputs are selected to represent environmental influences alongside immediate, daily, and weekly consumption dependencies. A static Random Forest model is established as the baseline, while an adaptive counterpart is periodically retrained using recent observations. The comparison examines whether updating the forecasting model provides a measurable improvement over retaining the initially trained model.[10]

The evaluation preserves the chronological ordering of observations to reflect the sequence in which forecasting information becomes available. Forecasting performance is assessed using absolute, squared, root-mean-square, and percentage-based error measures, enabling examination of average deviations, sensitivity to larger errors, and relative prediction accuracy. Feature-importance analysis further examines the contribution of the selected predictors. The contribution of the study lies in the implementation and assessment of a periodic retraining strategy within a consistent Random Forest forecasting workflow. Through comparison with a static baseline, the work investigates the practical value of model updating for smart-grid demand forecasting while providing a basis for subsequent evaluation under broader operating conditions.[11]

II. LITERATURE REVIEW

Electrical load forecasting has been extensively investigated as a means of supporting power-system scheduling, energy management, and demand planning. Existing approaches range from statistical time-series models to machine-learning and deep-learning techniques. Their effectiveness depends on the forecasting horizon, level of demand aggregation, availability of weather information, and characteristics of the consumption data. In their systematic review, Kuster, Rezgui, and Mourshed emphasized that forecasting methods should be selected according to the application context and evaluated within a consistent data-processing framework. This perspective is particularly relevant to adaptive forecasting, where performance depends on both the predictive model and the procedure used to incorporate newly available observations.[12][13]

Statistical methods provide an established foundation for electrical demand forecasting by representing temporal dependence, recurring seasonality, and the influence of external conditions. Fan and Hyndman developed a semiparametric additive approach incorporating calendar information, historical demand, and temperature information. Their work demonstrated how flexible statistical modelling can accommodate nonlinear relationships while retaining a structured representation of the factors influencing consumption. The study also considered prediction intervals, recognizing that an estimate of future demand alone does not fully describe forecasting uncertainty. Such approaches provide useful reference points for assessing whether more complex learning algorithms deliver meaningful improvements.[14]

The selection of historical and environmental predictors is a central issue across forecasting methodologies. Recent demand observations describe the immediate consumption state, whereas daily and weekly historical observations represent recurring patterns associated with human activity and operational schedules. Weather-related predictors require similar attention because demand may depend on preceding conditions as well as the temperature at the forecast time. Wang, Liu, and Hong investigated the recency effect in electrical load forecasting through an extensive examination of lagged and averaged temperature inputs. Their findings support the importance of representing historical weather influences appropriately. However, incorporating recent predictor information and retraining a forecasting model are distinct strategies, and evidence

supporting one should not be treated as direct validation of the other.[15][16]

Machine-learning methods have broadened the modelling options available for nonlinear demand prediction. Random Forest, introduced by Breiman, combines multiple randomized decision trees to produce an ensemble prediction. For load forecasting, this structure provides a means of learning relationships between weather conditions, time-related information, and historical consumption without prescribing a single functional relationship in advance. The method can represent interactions among predictors and is suitable for structured datasets containing different types of numerical inputs. Nevertheless, its effectiveness remains dependent on the quality and representativeness of the training observations, and its use does not remove the need for appropriate temporal validation.[17]

Deep-learning approaches have also been investigated for modelling the sequential characteristics of electricity consumption. Kong and colleagues applied a long short-term memory recurrent neural network to short-term residential load forecasting. Their work illustrates the use of recurrent architectures to learn temporal dependencies in demand sequences. Such models offer an alternative to explicitly constructed historical features, although model complexity, training requirements, and the amount of available data remain relevant considerations. A comparison between recurrent models and tree-based methods therefore requires consistent forecasting horizons, input availability, and testing periods to ensure that observed differences reflect the forecasting approaches being evaluated.[18]

Beyond prediction accuracy, the interpretation of influential inputs is important when assessing a forecasting model. Random Forest provides feature-importance estimates that can indicate how the fitted ensemble uses historical demand and environmental information. However, Strobl and colleagues demonstrated that conventional importance measures can be biased by predictor characteristics. Their subsequent work on conditional importance further showed that correlations between inputs affect the interpretation of their individual contributions. These findings are directly relevant to load forecasting because preceding-hour, preceding-day, and preceding-week demand measurements may contain overlapping information. Consequently, a highly ranked historical input should be interpreted as influential within the fitted model rather than as an independent explanation of electricity consumption.[19][20][21]

Model evaluation is equally important when determining whether a forecasting strategy offers a practical improvement. Cawley and Talbot showed that model selection can itself become overfitted, leading to optimistic estimates of predictive performance. In load forecasting, this concern reinforces the need to separate model development from final assessment and to preserve the information sequence associated with prediction. An adaptive model should therefore be updated using only observations available before the subsequent forecast is issued. Comparing static and adaptive models over the same testing period, with equivalent input information, helps isolate the effect of retraining from differences in experimental conditions.[22]

The reviewed literature establishes the importance of nonlinear modelling, historical dependencies, weather information, and careful validation in electrical load forecasting. It also provides a methodological basis for examining whether periodic retraining improves a forecasting model as additional observations become available. Within this scope, the present study compares static and periodically updated Random Forest regressors using the same set of environmental and historical demand features. Its focus is the incremental value of model updating under a consistent chronological evaluation procedure. The comparison is intended to determine whether the additional updating process produces measurable forecasting benefits, while feature analysis supports examination of the information used by the models.[23]

III. MATHEMATICAL MODELLING

The forecasting problem is formulated as the estimation of hourly electrical demand from weather conditions, time-of-day information, and previously observed consumption. The model combines immediate demand dependence with daily and weekly repetition. For an hourly dataset, the input representation is expressed as

$$\mathbf{x}_t = [T_t,\ H_t,\ h_t,\ L_{t-1},\ L_{t-24},\ L_{t-168}] \quad (1)$$

where T_t and H_t denote temperature and humidity, h_t represents the hour of the day, and the remaining terms represent demand observed one hour, twenty-four hours, and one hundred and sixty-eight hours earlier. The corresponding prediction is

$$\widehat{L}_t = f(\mathbf{x}_t). \quad (2)$$

Here, L_t denotes actual demand and $\hat{L}_t$ denotes predicted demand. All inputs must be available at the forecast issue time. When temperature and humidity for the target hour are unavailable in advance, operational forecasting requires their forecast values. Using measured target-hour weather instead corresponds to an evaluation conditional on observed weather.

Random Forest regression estimates demand by averaging the outputs of multiple decision trees. Each tree is trained using a bootstrap sample of the training observations, with a

randomly selected subset of features considered at each split. The ensemble prediction is

$$\widehat{L}_t = \frac{1}{M}\sum_{m=1}^{M} g_m(\mathbf{x}_t), \tag{3}$$

where M is the number of trees and g_m is the prediction produced by an individual tree. Within each terminal region, the tree prediction is the mean demand of the training observations assigned to that region. Averaging across trees reduces sensitivity to an individual tree and enables the ensemble to represent nonlinear relationships among the selected inputs.

For the static forecasting approach, the model is fitted once using the initial training period and retained throughout testing:

$$\widehat{L}_t^{\text{static}} = f_{\theta_0}(\mathbf{x}_t), \tag{4}$$

where θ_0 represents the fitted ensemble. This formulation provides a fixed reference against which the effect of periodic retraining can be assessed.

For the adaptive approach, the ensemble is periodically refitted using an updated training dataset. At an update time, the training set is defined as

$$\mathcal{D}_t = \{(\mathbf{x}_i, L_i) : i \in \mathcal{I}_t,\ i < t\}, \tag{5}$$

where I_t identifies the historical observations retained by the implemented updating procedure. This set may represent a recent rolling window or an expanding history, depending on the code configuration. Only observations whose demand values have already become available are eligible for retraining.

The adaptive prediction is subsequently obtained from

$$\theta_t = \text{TrainRF}(\mathcal{D}_t), \qquad \widehat{L}_t^{\text{adaptive}} = f_{\theta_t}(\mathbf{x}_t). \tag{6}$$

Between scheduled updates, the most recently fitted ensemble is retained. Adaptation therefore occurs through periodic model retraining rather than through continuous adjustment of individual tree parameters.

Forecasting performance is evaluated over the same chronological testing period for both approaches. The principal error measures are

$$\text{MAE} = \frac{1}{N}\sum_{t=1}^{N}|L_t - \widehat{L}_t|,$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(L_t - \widehat{L}_t)^2}, \tag{7}$$

$$\text{MAPE} = \frac{100}{N}\sum_{t=1}^{N}\left|\frac{L_t - \widehat{L}_t}{L_t}\right|.$$

Here, N represents the number of evaluated observations. Mean absolute error describes the average magnitude of forecasting deviations, while root-mean-square error assigns greater influence to larger deviations. Mean absolute percentage error expresses deviations relative to actual demand and requires nonzero actual values; observations with zero demand must be handled explicitly. Mean squared error, when reported, is the square of root-mean-square error.

The relative benefit of adaptation is quantified as

$$\text{Improvement}(\%) = \frac{E_{\text{static}} - E_{\text{adaptive}}}{E_{\text{static}}} \times 100, \tag{8}$$

where the same error measure is used for both models. A positive value indicates that periodic retraining reduced the forecasting error. This comparison isolates the contribution of updating when the input features, testing observations, and forecast information are otherwise equivalent.[24][25][26][27][28]

IV. MODEL IMPLEMENTATION

	A	B	C	D	E	F	G	H	I	J	K
	Date	Time	Temperat	Humidity_	Hour	Day_of_W	Day_Type	Previous_l	Load_24h	Load_168l	Actual_Load
	08-01-2026	00:00	21.64	68.33	0	Thursday	Working	64.08	58.54	60.23	56.15
	08-01-2026	01:00	19.82	74.78	1	Thursday	Working	56.15	63.89	61.29	58.67
	08-01-2026	02:00	18.37	74.88	2	Thursday	Working	58.67	62.92	58.28	62.83
	08-01-2026	03:00	17.16	79.12	3	Thursday	Working	62.83	64.31	65.26	61.53
	08-01-2026	04:00	16.84	80.44	4	Thursday	Working	61.53	64.62	58.91	66.85
	08-01-2026	05:00	16.31	78.06	5	Thursday	Working	66.85	68.84	70.61	67.04
	08-01-2026	06:00	15.44	85.78	6	Thursday	Working	67.04	70.88	73.1	75.19
	08-01-2026	07:00	15.6	83.88	7	Thursday	Working	75.19	80.97	80.41	77.47
	08-01-2026	08:00	14.45	80.39	8	Thursday	Working	77.47	82.86	82.18	80.23
	08-01-2026	09:00	16.35	93.04	9	Thursday	Working	80.23	84.5	83.4	83.5
	08-01-2026	10:00	14.79	85.18	10	Thursday	Working	83.5	85.93	82.51	80.71
	08-01-2026	11:00	19.16	81.39	11	Thursday	Working	80.71	78.97	81.45	78.06
	08-01-2026	12:00	17.69	86.01	12	Thursday	Working	78.06	79.71	78.07	79.34
	08-01-2026	13:00	17.11	85.78	13	Thursday	Working	79.34	80.09	80.22	79.32
	08-01-2026	14:00	18.15	75.77	14	Thursday	Working	79.32	77.56	76.37	82.57
	08-01-2026	15:00	21.31	75.67	15	Thursday	Working	82.57	81.35	81.04	83.93
	08-01-2026	16:00	21.67	68.4	16	Thursday	Working	83.93	85.89	86	87.6
	08-01-2026	17:00	23.83	70.1	17	Thursday	Working	87.6	87.97	90.23	90.38
	08-01-2026	18:00	24.33	68.62	18	Thursday	Working	90.38	94.46	91.95	92.43
	08-01-2026	19:00	24.18	69.36	19	Thursday	Working	92.43	87.98	89.78	88.71
	08-01-2026	20:00	23.42	69.68	20	Thursday	Working	88.71	89.39	88.25	87.37
	08-01-2026	21:00	22.45	70.86	21	Thursday	Working	87.37	81.14	77.56	81.11
	08-01-2026	22:00	23.23	70.37	22	Thursday	Working	81.11	70.82	75.27	75.39
	08-01-2026	23:00	24	65.85	23	Thursday	Working	75.39	64.08	65.28	67.74

Figure 1: SCREENSHOT OF THE DATASET USED FOR MACHINE LEARNING BASED ADAPTIVE LOAD FORECASTING FOR SMART GRID

The dataset as shown in Figure 1 contains 8,700 records and 16 columns. It provides time, weather, and historical electricity-load information, including temperature, humidity, hour, day of the week, previous load, 24-hour and 168-hour historical load, and actual load. These variables are used to predict the Actual_Load_kW using the Random Forest forecasting model.

The code implements a machine-learning-based adaptive electrical load forecasting system using Random Forest Regression. It begins by importing the required Python libraries for data handling, numerical calculations, visualization, machine learning, data scaling, and performance evaluation. Pandas is used to manage the dataset, NumPy performs numerical operations, and Matplotlib generates graphs. The Random Forest Regressor is used as the forecasting algorithm, StandardScaler is used for feature standardization, and MAE, MSE, and MAPE are used to evaluate forecasting accuracy.[29][30]

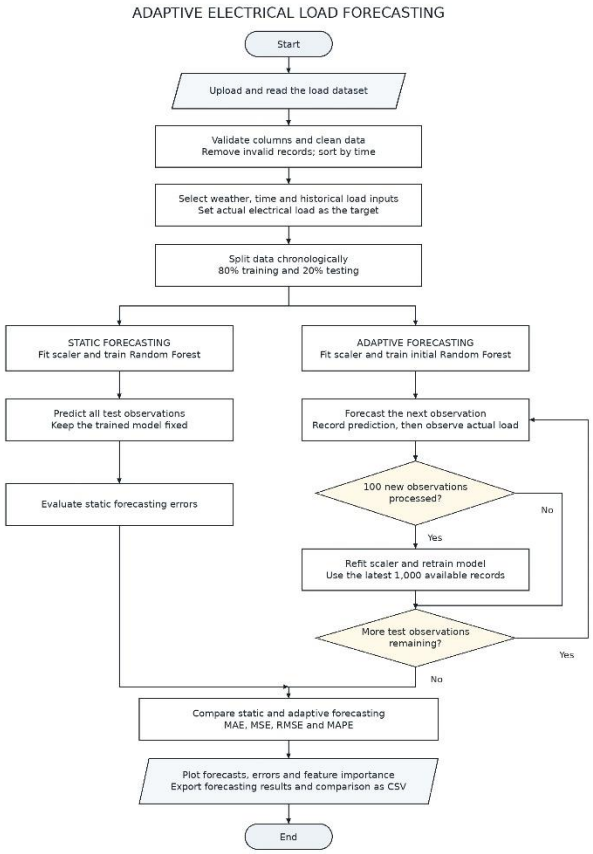


Figure 2: FLOWCHART OF THE PYTHON CODE IMPLEMENTING MACHINE-LEARNING-BASED ADAPTIVE ELECTRICAL LOAD FORECASTING SYSTEM USING RANDOM FOREST REGRESSION

The program allows the dataset to be uploaded and then reads it into a Pandas DataFrame. After loading the data, it displays the dataset dimensions so that the number of records and variables can be verified. Unnecessary columns whose names begin with "Unnamed" are removed because these columns are generally not useful for forecasting. The program then displays the remaining columns and checks whether all the required variables are available. If any required variable is missing, the program identifies the missing columns and stops execution to prevent errors during model development.

The numerical variables are then converted into appropriate numeric data types. Any invalid numerical values are converted into missing values so that they can be handled during the cleaning process. The separate date and time information are combined to create a single DateTime variable. This allows the observations to be arranged chronologically, which is important for electrical load forecasting because load values depend strongly on their position in time. Rows containing missing values in important forecasting variables are removed, and the cleaned dataset is sorted according to DateTime.[31]

After cleaning, the program displays the first few records and checks for remaining missing values in the major forecasting variables. The input features selected for the machine-learning model are temperature, humidity, hour of the day, previous electrical load, load from 24 hours earlier, and load from 168 hours earlier. These variables represent environmental conditions, daily time patterns, recent load behavior, previous-day load behavior, and previous-week load behavior. The actual electrical load is selected as the target variable that the model needs to predict.[32]

The dataset is divided into training and testing portions using an 80:20 ratio. The first 80% of the observations are used for model training, while the remaining 20% are used for testing. The chronological order of the data is maintained rather than randomly mixing the observations. This is appropriate for time-dependent forecasting because the model is trained using earlier observations and evaluated using later observations, which better represents a real forecasting situation.[33]

The selected input features are standardized using StandardScaler. The scaler is fitted using only the training data and is subsequently applied to both the training and testing data. This provides a consistent transformation of the input variables before they are supplied to the forecasting model.[34]

A Random Forest Regression model is then created as the static forecasting model. The model contains 100 decision trees, with a maximum tree depth of 15. A fixed random state is used to make the results reproducible, while all available processing cores are utilized to improve computational efficiency. The model is trained using the scaled training features and the corresponding actual load values. After training, it predicts the electrical load for the testing period.

The performance of the static model is evaluated using MAE, MSE, RMSE, and MAPE. MAE measures the average magnitude of the forecasting error without considering its direction. MSE squares the forecasting errors, giving greater importance to large errors. RMSE is obtained by taking the square root of MSE and expresses the error in the same unit as electrical load, which is kilowatts. MAPE expresses the forecasting error as a percentage and provides an indication of the relative prediction accuracy.[35]

The program then creates a set of forecasting results containing the actual electrical load, predicted electrical load, forecasting error, and absolute forecasting error. The forecasting error is calculated by subtracting the predicted load from the actual load. These results make it possible to examine how closely the static model follows the actual load values during the testing period.[36]

The adaptive forecasting stage begins by defining a window size of 1000 samples and an update frequency of 100 samples. An initial Random Forest model is trained using the available training data. Unlike the static model, however, this adaptive model is not kept unchanged throughout the entire testing period. Instead, it continuously processes the test observations one at a time and produces a prediction for each observation. The predicted value and corresponding actual value are stored for later performance evaluation.[37]

After every 100 test samples, the adaptive model is updated. During each update, the model selects the most recent 1000 observations, or all available observations if fewer than 1000 are available. These recent observations are used to create a new training dataset. A new StandardScaler is fitted to this recent data, and the Random Forest model is retrained using the updated information. This process allows the forecasting system to give greater importance to recent electrical-load behavior and adapt to changes in consumption patterns over time.[38]

Once all testing observations have been processed, the adaptive predictions and actual load values are converted into arrays. The adaptive model is then evaluated using MAE, MSE, RMSE, and MAPE. These metrics are calculated in the same manner as those of the static model so that the two forecasting approaches can be compared fairly. Lower values of these error metrics generally indicate better forecasting performance.[39]

The program creates a comparison of the static and adaptive Random Forest models using the four evaluation metrics. It also calculates the percentage improvement in MAE and MAPE and the percentage change in RMSE. These calculations help determine whether adaptive retraining improves the forecasting accuracy compared with a model that remains unchanged after its initial training.[40]

The adaptive forecasting results are then organized together with the corresponding date and time, actual load, predicted load, forecasting error, and absolute error. The first 20 results are displayed to provide a detailed view of the adaptive model's forecasting behavior.[41]

The program also produces several graphical representations of the forecasting performance. The first graph compares the actual electrical load with the adaptive predicted load for the first 500 testing observations. If the predicted curve follows the actual curve closely, it indicates that the forecasting model is successfully capturing the variations in electrical demand.

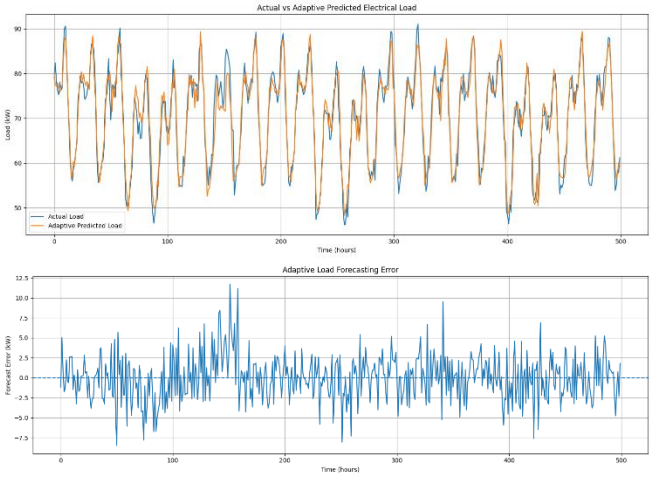
The second graph shows the forecasting error for the first 500 testing observations. The error is calculated as the actual load minus the predicted load. A zero reference line is included to indicate perfect forecasting. Positive errors represent situations where the model underestimates the actual load, while negative errors represent situations where the model overestimates the actual load.[42]

The third graph is a scatter plot comparing actual load values with predicted load values. A diagonal reference line is plotted to represent perfect prediction. Points located close to this line indicate that the predicted values are close to the actual values, whereas points farther away indicate larger forecasting errors.[43]

The program then calculates the importance of each input feature using the feature importance values generated by the final adaptive Random Forest model. These values indicate the relative contribution of temperature, humidity, hour, previous load, 24-hour historical load, and 168-hour historical load to the model's predictions. The features are arranged from the most important to the least important, and a bar graph is generated to provide a visual representation of their relative importance.[44]

Finally, the program saves the adaptive forecasting results and the comparison of the two models as separate CSV outputs. This allows the forecasting results, errors, and performance metrics to be examined later or used for further analysis. The program concludes by displaying confirmation messages indicating that the required outputs have been generated and that the complete adaptive electrical load forecasting process has been successfully executed.

V. RESULTS AND DISCUSSION



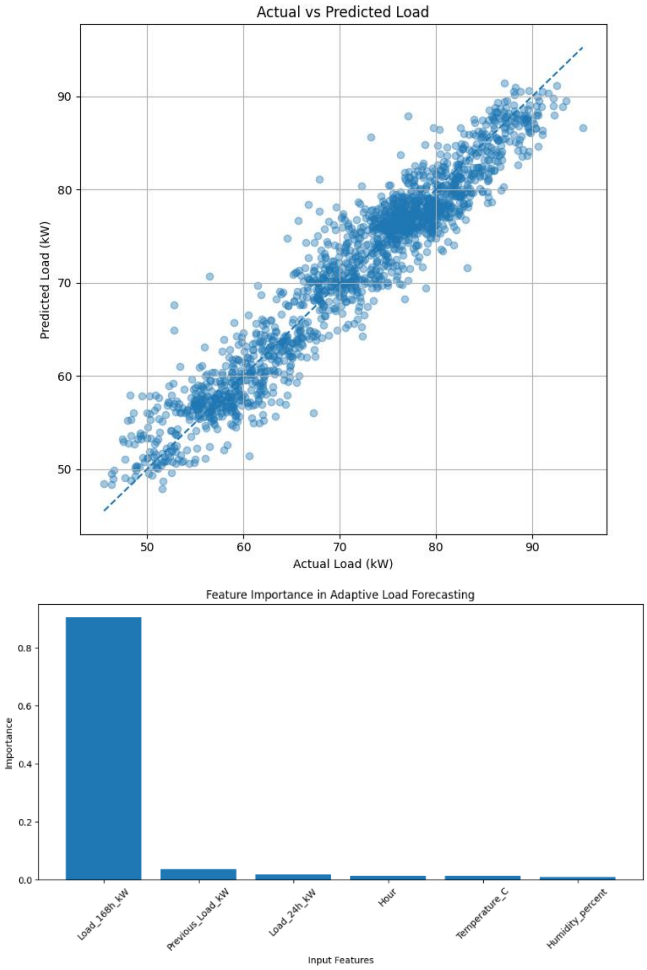


Figure 3: OUTPUT OF THE PYTHON CODE IMPLEMENTING MACHINE LEARNING BASED ADAPTIVE LOAD FORECASTING FOR SMART GRID

The output as shown in Figure 3 shows that the adaptive Random Forest model successfully performed electrical load forecasting. The dataset initially contained 8,700 records and 16 columns. After removing unnecessary columns and cleaning the data, 8,592 records remained. There were no missing values in the important forecasting variables, so the dataset was ready for model training.

The cleaned data was divided into 80% training data and 20% testing data. This resulted in 6,873 training samples and 1,719 testing samples. The Random Forest model was successfully trained using temperature, humidity, hour, previous load, 24-hour load, and 168-hour load as input variables.

For the static Random Forest model, the MAE was 2.369 kW, RMSE was 3.009 kW, and MAPE was 3.456%. This means that, on average, the predicted electrical load differed from the actual load by about 2.37 kW, while the average percentage error was about 3.46%. These results indicate reasonably accurate load prediction.

The adaptive model was updated after every 100 testing samples, up to 1,700 samples. This means the model repeatedly learned from recent load information instead of remaining unchanged throughout the testing period. The adaptive model achieved an MAE of 2.339 kW and MAPE of 3.383%. Compared with the static model, the MAE improved by 1.26% and MAPE improved by 2.13%. However, RMSE increased slightly by 0.27%, meaning that the adaptive model did not reduce the influence of larger individual errors.

The final forecasting results show that the predicted values are generally close to the actual load values. For example, when the actual load was 78.91 kW, the predicted value was 80.08 kW, giving an error of approximately −1.17 kW. Similarly, for an actual load of 90.00 kW, the predicted load was approximately 87.40 kW, giving an error of about 2.60 kW. Therefore, the model is able to follow the general variation in electrical demand.

The feature-importance result shows that Load_168h_kW is the most important input feature, with an importance of approximately 90.52%. This means that the load from 168 hours earlier, or approximately one week earlier, has the strongest influence on the model's prediction. Previous load has about 3.71% importance, followed by 24-hour historical load at 1.81%, hour at 1.50%, temperature at 1.38%, and humidity at 1.07%.

Overall, the output indicates that the adaptive Random Forest provides a small improvement over the static Random Forest, particularly in MAE and MAPE. The model successfully completed the forecasting process, generated predictions and error values, performed model comparison and feature-importance analysis, and successfully saved the required results.

VI. CONCLUSION

The project successfully developed an adaptive electrical load forecasting system using Random Forest Regression. The system uses historical load data along with temperature, humidity, and time-related information to predict future electrical demand. The dataset was cleaned and divided into training and testing data, and both static and adaptive Random Forest models were evaluated using MAE, MSE, RMSE, and MAPE.

The results show that the adaptive model achieved a MAE of 2.339 kW and MAPE of 3.383%, compared with 2.369 kW and 3.456% for the static model. This represents an improvement of 1.26% in MAE and 2.13% in MAPE, demonstrating that periodic model updating can provide a small improvement in forecasting accuracy. The feature-importance analysis also showed that the 168-hour historical load was the most influential factor in predicting future electrical demand.

Overall, the project demonstrates that an adaptive Random Forest approach can effectively forecast electrical load and respond to changing consumption patterns. Although the improvement over the static model is relatively small and some larger prediction errors remain, the proposed system can be useful for smart grids, energy management, demand-side management, peak-load management, microgrids, and efficient power-generation planning. Thus, the project provides a practical foundation for developing intelligent and adaptive electrical energy forecasting systems.

REFERENCES

[1] G. Gross and F. D. Galiana, "Short-term load forecasting," Proceedings of the IEEE, vol. 75, no. 12, pp. 1558–1573, 1987.
[2] H. Hahn, S. Meyer-Nieberg, and S. Pickl, "Electric load forecasting methods: Tools for decision making," European Journal of Operational Research, vol. 199, no. 3, pp. 902–907, 2009.
[3] C. Kuster, Y. Rezgui, and M. Mourshed, "Electrical load forecasting models: A critical systematic review," Sustainable Cities and Society, vol. 35, pp. 257–270, 2017.
[4] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," International Journal of Forecasting, vol. 32, no. 3, pp. 914–938, 2016.
[5] H. S. Hippert, C. E. Pedreira, and R. C. Souza, "Neural networks for short-term load forecasting: A review and evaluation," IEEE Transactions on Power Systems, vol. 16, no. 1, pp. 44–55, 2001.
[6] S. Fan and R. J. Hyndman, "Short-term load forecasting based on a semi-parametric additive model," IEEE Transactions on Power Systems, vol. 27, no. 1, pp. 134–141, 2012.
[7] J. W. Taylor, L. M. de Menezes, and P. E. McSharry, "A comparison of univariate methods for forecasting electricity demand up to a day ahead," International Journal of Forecasting, vol. 22, no. 1, pp. 1–16, 2006.
[8] J. W. Taylor and R. Buizza, "Using weather ensemble predictions in electricity demand forecasting," International Journal of Forecasting, vol. 19, no. 1, pp. 57–70, 2003.
[9] Y. Goude, R. Nedellec, and N. Kong, "Local short and middle term electricity load forecasting with semi-parametric additive models," IEEE Transactions on Smart Grid, vol. 5, no. 1, pp. 440–446, 2014.
[10] P. Wang, B. Liu, and T. Hong, "Electric load forecasting with recency effect: A big data approach," International Journal of Forecasting, vol. 32, no. 3, pp. 585–597, 2016.
[11] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, and R. J. Hyndman, "Probabilistic energy forecasting: Global Energy Forecasting Competition 2014 and beyond," International Journal of Forecasting, vol. 32, no. 3, pp. 896–913, 2016.
[12] T. Hong, P. Wang, and L. White, "Weather station selection for electric load forecasting," International Journal of Forecasting, vol. 31, no. 2, pp. 286–295, 2015.
[13] J. Xie and T. Hong, "GEFCom2014 probabilistic electric load forecasting: An integrated solution with forecast combination and residual simulation," International Journal of Forecasting, vol. 32, no. 3, pp. 1012–1016, 2016.
[14] J. Nowotarski, B. Liu, R. Weron, and T. Hong, "Improving short term load forecast accuracy via combining sister forecasts," Energy, vol. 98, pp. 40–49, 2016.
[15] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," IEEE Transactions on Smart Grid, vol. 10, no. 1, pp. 841–851, 2019.
[16] E. Mocanu, P. H. Nguyen, M. Gibescu, and W. L. Kling, "Deep learning for estimating building energy consumption," Sustainable Energy, Grids and Networks, vol. 6, pp. 91–99, 2016.
[17] D. L. Marino, K. Amarasinghe, and M. Manic, "Building energy load forecasting using deep neural networks," in Proc. IECON 2016—42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 2016, pp. 7046–7051.
[18] L. Suganthi and A. A. Samuel, "Energy models for demand forecasting—A review," Renewable and Sustainable Energy Reviews, vol. 16, no. 2, pp. 1223–1240, 2012.
[19] R. Weron, Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach. Chichester, U.K.: Wiley, 2006.
[20] E. A. Feinberg and D. Genethliou, "Load forecasting," in Applied Mathematics for Restructured Electric Power Systems: Optimization, Control, and Computational Intelligence, J. H. Chow, F. F. Wu, and J. J. Momoh, Eds. New York, NY, USA: Springer, 2005, pp. 269–285.
[21] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[22] L. Breiman, "Bagging predictors," Machine Learning, vol. 24, no. 2, pp. 123–140, 1996.
[23] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and Regression Trees. Belmont, CA, USA: Wadsworth International Group, 1984.
[24] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," Machine Learning, vol. 63, no. 1, pp. 3–42, 2006.
[25] T. G. Dietterich, "Ensemble methods in machine learning," in Multiple Classifier Systems, vol. 1857, Lecture Notes in Computer Science. Berlin, Germany: Springer, 2000, pp. 1–15.
[26] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. New York, NY, USA: Springer, 2009.
[27] C. M. Bishop, Pattern Recognition and Machine Learning. New York, NY, USA: Springer, 2006.
[28] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time Series Analysis: Forecasting and Control, 5th ed. Hoboken, NJ, USA: Wiley, 2015.
[29] R. J. Hyndman and G. Athanasopoulos, Forecasting: Principles and Practice, 3rd ed. Melbourne, Australia: OTexts, 2021.
[30] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," ACM Computing Surveys, vol. 46, no. 4, Art. no. 44, 2014.
[31] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in Proc. 2007 SIAM International Conference on Data Mining, Minneapolis, MN, USA, 2007, pp. 443–448.
[32] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," International Journal of Forecasting, vol. 22, no. 4, pp. 679–688, 2006.

[33] F. X. Diebold and R. S. Mariano, “Comparing predictive accuracy,” Journal of Business & Economic Statistics, vol. 13, no. 3, pp. 253–263, 1995.
[34] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” Journal of Machine Learning Research, vol. 13, pp. 281–305, 2012.
[35] G. C. Cawley and N. L. C. Talbot, “On over-fitting in model selection and subsequent selection bias in performance evaluation,” Journal of Machine Learning Research, vol. 11, pp. 2079–2107, 2010.
[36] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” Journal of Machine Learning Research, vol. 3, pp. 1157–1182, 2003.
[37] C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn, “Bias in random forest variable importance measures: Illustrations, sources and a solution,” BMC Bioinformatics, vol. 8, Art. no. 25, 2007.
[38] C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, “Conditional variable importance for random forests,” BMC Bioinformatics, vol. 9, Art. no. 307, 2008.
[39] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[40] C. R. Harris et al., “Array programming with NumPy,” Nature, vol. 585, no. 7825, pp. 357–362, 2020.
[41] W. McKinney, “Data structures for statistical computing in Python,” in Proc. 9th Python in Science Conference, Austin, TX, USA, 2010, pp. 51–56.
[42] J. D. Hunter, “Matplotlib: A 2D graphics environment,” Computing in Science & Engineering, vol. 9, no. 3, pp. 90–95, 2007.
[43] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[44] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in Advances in Neural Information Processing Systems, vol. 30, pp. 4765–4774, 2017.