

Integrated AI-Driven Smart Power Grid for Real-Time Monitoring and Forecasting

S Venkatasubramanian¹, Uday R², Vasanth V³, Suwetha A⁴

¹ Associate professor, Dept. of CSBS, Saranathan College of Engineering, Trichy, Tamil Nadu, India

^{2,3,4} UG Scholar, Dept. of CSBS, Saranathan College of Engineering, Trichy, Tamil Nadu, India

Emails: csbs.hod@Saranathan.ac.in¹, uday12704@gmail.com², vasanth232714@gmail.com³, suwethaayyappan13@gmail.com⁴

Abstract

In the context of the present day, the availability of non-renewable energy resources is declining, and the energy demand is on the rise with the increase in population and technological advancements. Although the use of solar energy and other forms of renewable energy has been integrated with the conventional energy grid, the intermittent nature of these forms of energy often creates an energy supply-demand imbalance and inefficient use of energy. To overcome the problems associated with the efficient use of energy, the present work proposes the development of an AI-based smart grid energy prediction system called NeuroWattic, which optimizes the use of energy, prevents wastage of energy, and promotes the efficient use of renewable energy. The model uses the stacking ensemble learning method, which combines the Random Forest, Extreme Gradient Boosting (XGBoost), and Histogram-based Gradient Boosting Machine (HistGBM) models as the base models, and the Ridge Regression model as the meta-model for efficient energy prediction. The model has been trained with high-resolution, multi-zone power consumption data, with a time resolution of ¹⁰ minutes, and meteorological data such as temperature, humidity, wind speed, and solar irradiance. Feature engineering has been done on the dataset, which creates temporal features, lag features, and weather-based features, which are used to model the complex energy consumption patterns. The model has been designed to perform iterative multi-step forecasting up to ¹⁴ days into the future, which enables efficient energy management. The performance of the model has been tested, and the results indicate significant improvement in the efficiency of the model, with an RMSE of ^{183.18} kW, MAE of ^{93.83} kW, MAPE of ^{0.1537}%, and R² score of ^{0.9998}. Moreover, the model has been integrated with an interactive dashboard that enables real-time monitoring, visualization of the duck curve, and cost optimization. The proposed system provides a scalable and intelligent solution for improving energy efficiency and enabling reliable renewable energy integration in modern smart grids.

Keywords: Smart Grid, Machine Learning, Load Forecasting, Artificial Intelligence, Demand-Side Management

1. Introduction

The global transition toward renewable energy sources, particularly solar photovoltaic (PV) systems, has introduced significant complexity into modern power grids. Unlike traditional forms of electricity generation, solar power generation varies inherently and depends on the weather, thereby creating a mismatch between supply and demand that affects the world over. This can be depicted as a curve known as the "Duck Curve." To address this issue, forecasting techniques need to be employed. Accurate forecasting of loads is the backbone of a

sustainable grid management system. It is essential for scheduling the grid's generating resources, reducing energy wastage, peak demand management, and integrating renewable energy without compromising grid security. The consequences of poor forecasting are far-reaching, and a series of power restrictions were witnessed across various provinces in China during late 2020, which is a consequence of a lack of accurate forecasting capability. Traditional models such as ARIMA and exponential smoothing are easy to understand but do not account for the non-linear

relationships between meteorological factors and electricity demand. Machine learning models such as Random Forest [1], XGBoost [2], and deep learning models such as LSTM [3] have been found to perform better in terms of accuracy in recent studies. However, there exists no best model that can perform under all circumstances. Ensemble methods are used to overcome this problem; stacking generalization is an example of an ensemble method. This paper introduces NeuroWattic, a holistic AI-assisted smart power grid system that leverages a stacking ensemble model with three models and an interactive real-time dashboard. The system was trained on a high-resolution power consumption data set with a multi-zone power consumption profile, which was supplemented with over 55+ engineered features. The stacking ensemble model was able to achieve a MAPE score of 0.1537% and an R^2 score of 0.9998 on a separate test data set, surpassing all the base models and blending models. In addition to its forecasting prowess, the system offers a variety of actionable energy-related insights. The rest of this paper is organized as follows: Section II provides a review of related literature on various load forecasting methodologies. Section III introduces the dataset and feature engineering strategy. Section IV explains the stacking ensemble architecture. Section V explains the interactive dashboard design. Section VI reports experimental results and comparative analysis. Section VII draws practical implications and conclusions.

2. Related Work

Significant advancements in load forecasting have been achieved from traditional statistical methods to advanced machine learning and deep learning approaches. ARIMA models, for example, were initially based on linear assumptions and stationary data, which were unable to accurately predict complex consumption behaviours of smart grids. To improve the limitations of traditional models, medium- and long-term forecasting models, known as time series models, were introduced. Prophet [5] is a popular model for medium- and long-term forecasting. This model accurately predicts the trend and seasonality of the data; however, it is unable to account for other factors, which makes it less adaptable for smart grids. To improve the limitations

of traditional models, deep learning models were introduced, achieving better results in terms of accuracy. [1] introduced a CNN-LSTM hybrid model, achieving better results in short-term load predictions. Similarly, an improved model was introduced by [3], which improved the accuracy of the results by minimizing residual error. Furthermore, improved results were achieved by [4], introducing a high-precision LSTM model that is capable of accurately predicting nonlinear dependencies. In addition to individual models, a series of ensemble learning methods have also been developed for better performance in forecasting. A stacking ensemble framework was proposed by Shi and Zhang [2], where multiple models are combined for better accuracy. Yang et al. [8] extended the idea of the stacking model with a blending model that incorporates Prophet, LSTM, and XGBoost, resulting in better predictive performance for mid-term load forecasting. Recent research also investigated the significance of artificial intelligence in smart energy systems. For example, Wang et al. [6] discussed the importance of artificial intelligence in smart energy systems, focusing on load forecasting, anomaly detection, and demand response. Similarly, Kiasari et al. [10] and Deshpande [11] emphasized the importance of machine learning in smart grids, energy efficiency, and demand response. Sulaiman et al. [7] discussed the importance of artificial intelligence in secure power grid protocols. Furthermore, AI-based renewable energy forecasting has attracted interest in recent studies. Wen et al. [9], [13] have proposed integrated frameworks for solar energy generation forecasting and optimization of the smart grid, enhancing the efficiency of the grid. In addition, the studies carried out by Rajaperumal and Columbus [14] and Strielkowski and Smutka [15] emphasize the significance of predictive analytics in the development of the grid's resilience, efficiency, and sustainability. However, it has to be mentioned that despite such advancements, most of the existing studies have been focused on improving model development as well as accuracy enhancement. Not much has been explored in terms of making such models more interactive as well as real-time enabled systems. Hence, this study has attempted to bridge

this gap to some extent through the development of a stacking ensemble method-based forecasting system that can also be integrated into an interactive system. This would not only help in improving the accuracy of such a system but also make it more interactive as a real-time system in smart energy systems.

3. Dataset Description And Pre-Processing

3.1. Dataset Description And Data Source

The research is based on high-resolution power consumption data collected every 10 minutes and covers three different geographic zones. The data collected covers more than one year and has the following feature groups:

- Environmental Attributes: Temperature (°C), Humidity (%), Wind Speed (m/s), General Diffuse Flows (proxy for solar irradiance), and Diffuse Flows.
- Target Variables: Power Consumption in Zone 1, Zone 2, and Zone 3 (Watts), with Total Power Consumption computed as their sum.
- Temporal Attributes: Datetime stamps from which structured temporal features are derived.

Having multiple zones is essential for zone-level forecasting, which is crucial for zone-level grid control. Using diffuse flows as a proxy for solar irradiance is also important for predicting the Duck Curve without relying on direct irradiance measurements.

3.2. Data Pre-Processing And Feature Engineering

The data went through rigorous pre-processing using multiple stages to ensure the best convergence and accuracy for the model. More than 55 features were extracted using the basic variables and are categorized into the following:

3.2.1. Temporal Feature Extraction

The Datetime variable was used to derive different features. The Hour variable was split into both linear and cyclical representations. Cyclical encoding using sine and cosine functions is used to maintain continuous periodicity for the features without discontinuity at boundary transitions:

$$Hour_{sin} = \sin(2\pi \cdot Hour/24)$$

$$Hour_{cos} = \cos(2\pi \cdot Hour/24)$$

The Datetime variable is split into both linear and cyclical representations. Cyclical encoding using sine and cosine functions is used to maintain continuous periodicity for the features without discontinuity at boundary transitions.

3.2.2. Meteorological Derived Features

Additional advanced weather-derived indices were calculated to improve the model's understanding of demand-driving conditions:

- Heat Index a combination of temperature and humidity that more closely represents the perceived temperature.
- Cooling Degree Hour (CDH) and Heating Degree Hour (HDH): CDH = max(0, T - 18) and HDH = max(0, 18 - T) quantify the energy required for climate control above or below the thermal comfort threshold of 18°C.
- Solar Irradiance and Solar Power: Computed from diffuse flow measurements weighted by a modelled solar elevation angle that varies with hour of day and day of year.
- Wind Chill: Captures the combined effect of temperature and wind speed on heating demand.
- Interaction Features: Temperature×Humidity, Temperature×WindSpeed, and Temperature² capture non-linear environmental interaction effects.

3.2.3. Temporal Lag And Rolling Statistics

Autoregressive lag features capture temporal dependencies in consumption patterns:

- Power Lag Features: Lag values at 1h, 2h, 3h, 6h, 12h, 24h, 48h, and 168h (one week) enable the model to learn within-day, day-ahead, and week-ahead recurrence patterns.
- Rolling Statistics: 6h, 24h, and 168h rolling mean and standard deviation of power consumption; 24h rolling mean of temperature.
- Exponential Moving Averages (EMA): 6h and 24h EMA of power consumption to capture recent trend momentum.
- Rate-of-Change Features: 1h and 24h

differenced power consumption; 1h differenced temperature, capturing the velocity of demand and weather changes.

- Zone Ratio Features: Per-zone share of total consumption (Zone1_Ratio, Zone2_Ratio, Zone3_Ratio) to support zone-level disaggregation.

3.2.4. Missing Value Handling And Scaling

Missing values were addressed using linear interpolation, preserving temporal continuity better than mean imputation for time-series data. All features were subsequently scaled using Robust Scaler, which normalizes using median and interquartile range, providing resilience to the outliers common in sensor-based power consumption datasets. The complete feature engineering pipeline yields 55+ input features per observation.

4. Predictive Modelling Framework

4.1. Stacking Ensemble Architecture

The predictive core of NeuroWattic is a three-level stacking generalization architecture. This meta-learning framework trains a secondary model on the out-of-fold (OOF) predictions of multiple base learners, allowing the meta-model to learn the optimal combination of base model strengths without overfitting. The architecture is summarised as follows:

$$\hat{y}_{final} = w_1 \hat{y}_{RF} + w_2 \hat{y}_{XGB} + w_3 \hat{y}_{GBM} + b$$

where the weights w_1, w_2, w_3 are learned by the Ridge meta-learner from OOF predictions, rather than being set heuristically as in simple blending. Shown in Figure 1.

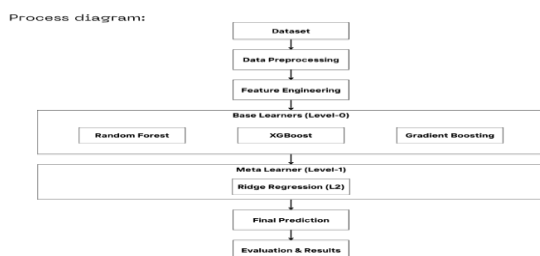


Figure 1 Process Diagram

4.2. Level-0 Base Learners

4.2.1. Random Forest (Rf)

Random Forest constructs B independent decision trees on bootstrap samples of the training data, with

random feature subsets considered at each split. For regression, the prediction is the ensemble mean:

- Mechanism: It draws multiple bootstrap samples from the training data and trains a standalone decision tree on each. For regression, the final output is the average of all individual tree predictions:

$$\hat{y} = \frac{1}{B} \sum_{b=1}^B T_b(x)$$

- The randomization over both samples and features reduces variance substantially compared to single trees, producing robust predictions on noisy sensor data. In NeuroWattic, Random Forest is configured with 100–200 trees, max depth of 15, and min_samples_leaf of 3.
- Strength of Power Systems: RF's method of bagging is still effective against outliers and noise inherent in actual sensor-based power consumption data. It is also able to effectively capture complex relationships among various weather parameters and zone consumption without requiring feature selection.

4.2.2. Extreme Gradient Boosting (Xgboost)

XGBoost is a fast and efficient tree boosting method that combines gradient boosting and decision trees. In this technique, we create an additive model in which each tree is learned from the residuals of the previous tree.

The regularized objective function is:

$$L = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k), \Omega(f) = \gamma T + \frac{1}{2} \lambda \sum w_j^2$$

where T is the number of leaf nodes and w_j are leaf scores.

It also includes γ and λ regularization to prevent overfitting. XGBoost makes use of a histogram-based splitting algorithm to increase efficiency and can handle missing values. For our NeuroWattic model, XGBoost parameters are usually set to 150 to 300 trees (controlled by speed mode), a learning rate of 0.05 to 0.12, a maximum depth of 5 to 7, and subsample/colsample_bytree values of 0.8.

Strength in the power system: It handles non-linear spikes in power demand well and can adapt to quick

changes in weather.

4.2.3. HistGradient Boosting Machine (GBM)

- The Histogram-based Gradient Boosting Machine is a step-by-step method for building decision trees that pursue remaining error gradients left over from previous trees. This is unlike other gradient boosters, which use a continuous variable method to speed up computation without sacrificing accuracy. This method also handles missing values and incorporates a validation set for early stopping to prevent overfitting.
- The mechanism of the Gradient Boosting Machine is such that, unlike in Random Forests, trees are built sequentially. Each tree is built to classify the residuals of the previous tree:

$$F_m(x) = F_{m-1}(x) + h_m(x)$$

where will be the new weak learner. h_m

- Strength in Power Systems: The Gradient Boosting Machine is particularly effective in identifying smooth, slowly changing load trends and patterns in a power system. It is particularly effective in identifying slowly changing seasonal and diurnal patterns in load data, which is characteristic of consumption data. The decremental error minimization method of GBM complements XGBoost's aggressive correction method.

4.3. Stacked Ensemble

4.3.1. Level-1: Ridge Regression Meta-Learner

The Ridge Regression meta-learner receives the out-of-fold predictions of all three base models as input features, and learns their optimal linear combination. Ridge adds an L2 penalty to prevent the meta-model from over-weighting any single base learner:

4.3.2. 4.3.2 Ridge Regression (L2 Regularization)

Ridge Regression is used as the meta-learner in the stacking framework. It introduces L2 regularization to penalize large coefficients and prevent overfitting. The objective function is:

$$\min \left(\sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p w_j^2 \right)$$

where is the regularization parameter. λ . The use of OOF predictions rather than training-set predictions is critical: it ensures the meta-learner is trained on predictions made on data the base models have not seen, preventing information leakage and overfitting. This is a key advantage of stacking over simple weighted averaging (blending), where weights are typically set heuristically or optimized on the same training data used to train the base models.

Actual vs All Predictions

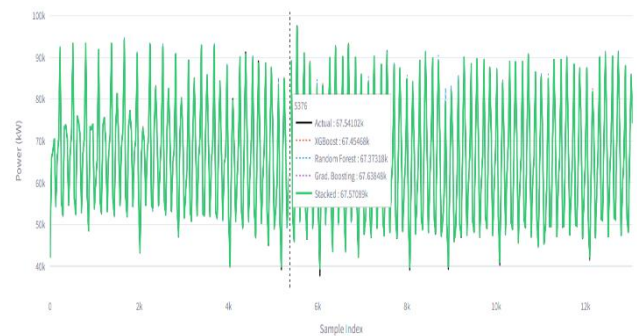


Figure 2 Actual Vs All Predictions

4.4. K-Fold Out-Of-Fold Training Protocol

Base model predictions for meta-learner training are generated using K-Fold cross-validation (K=3 for Fast/Balanced modes, K=5 for Accurate mode) with no shuffling, preserving temporal order. For each fold, the base model is retrained on the remaining folds and generates predictions for the held-out fold. These OOF predictions form the meta-features, while average test-set predictions across folds are used as the test meta-features. This ensures the meta-learner generalizes to unseen operational conditions.

4.5. Iterative Future Forecasting

For multi-step-ahead forecasting, NeuroWattic employs an autoregressive (iterative) strategy. Each hour's prediction is immediately incorporated as lag features (Power_Lag1, Power_Lag2, etc.) and rolling statistics for the subsequent prediction step. Future meteorological variables are created by using historical patterns of hourly data and then adding seasonal noise and user-specified temperature shifts for scenario analysis. This approach allows for forecasting out to 14 days (336 hours) with uncertainty as confidence intervals broaden.

5. Interactive Dashboard And Visualization

To translate the forecast into actual decision-making, the NeuroWattic outputs are integrated into a wide range of analysis dashboards built with Streamlit. The analysis dashboard consists of seven modules:

5.1. Power Consumption Overview

The main view of the Power Consumption Overview will feature real-time power consumption, peak and average loads, minimum load, and load factor. It will also allow selecting the date range for the data to be shown. The 24-hour average load profile, zone-by-zone load (shown as a pie chart), weather impact (shown as scatter plots), and results of time series analysis, such as two-axis temperature plots, peak and average load charts, and seasonal trends, will be generated with Plotly.

5.2. Demand Analysis Module

The derived demand patterns include hourly load profiles, daily demand tendencies, time-of-day load variation, weekday/weekend demand variation, correlation between meteorological and engineered features, temperature sensitivity, and heat index sensitivity. These will offer insights into the feature engineering process used in the model development.

5.3. Stacking Ensemble Training And Evaluation

The user will be able to choose the target zone (Total/Zone1/Zone2/Zone3), the speed mode (Fast/Balanced/Accurate), and the test size. The results will include: a comparison table of the five models (XGBoost, RF, GBM, Simple Blend, and Stacked Ensemble), Ridge meta-learner weight visualization, actual and predicted time series plots for all models, residuals, and top 20 feature importances comparing XGBoost and RF models.

5.4. Enhanced Forecasting Module (8 Analytical Views)

The Forecast Tab offers eight different types of analytical views:

- Confidence Interval Bands: Providing 80%, 90%, Or 95% Confidence Interval Bands For Forecasts That Increase With Horizon.
- Forecast Heatmap: Displaying A Heatmap Of Hours Of The Day Against Date To Visualize Demand Patterns.
- Anomaly And Spike Detection: Detecting Anomalies In Demand Compared To A 24-

Hour Moving Average Via Z-Score Analysis.

- Scenario Analysis: Providing forecasts for Optimistic Scenario, Base Scenario, and Pessimistic Scenario with user-specified temperature offsets (default is $\pm 2^{\circ}\text{C}$).
- Zone Level Forecast Breakdown: Smoothing Total Demand Into Zone 1, Zone 2, And Zone 3 Using Historical Ratios.
- Temperature Sensitivity Table: Analyzing temperature sensitivity between -4°C and $+4^{\circ}\text{C}$ to calculate average demand, peak demand, total energy, and variations compared to the original scenario.
- Historical Comparison: Displaying side-by-side forecasts and historical data for the same period (1 week, 2 weeks, or 1 month prior).
- Ramp Rate Analysis: Displaying The Rate Of Change In Demand (Kw/Hr) With The Top 10 Events Highlighted.

5.5. Optimization And Power Purchase Modules

The optimization module detects the times of day when demand is high or low, measures the amount of demand shift, performs simulations for battery storage, and designs demand response strategies. On the other hand, the power purchase planning module creates a second blended ensemble model for net purchase requirements, makes 7- to 90-day forecasts of procurement needs with time-of-use pricing, and separates costs, monthly summary, optimization, and ROI for expansion of renewable energy and battery storage.

5.6. Evaluation Metrics

There are several metrics to evaluate the performance of the model, including the following common metrics for calculating errors:

- Mean Absolute Error (MAE): This is a linear calculation of the error in kilowatts, which is easy to understand and interpret.

$$\text{MAE} = \frac{1}{n} \sum_{t=1}^n |y_t - \hat{y}_t|$$

where, y_t = actual value, \hat{y}_t = predicted value, n = number of observations

- **Root Mean Square Error (RMSE):** This measures the sum of the squares of the values, penalizing greater values, so it can detect large deviations.

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2}$$

- **Mean Absolute Percentage Error (MAPE):** It is expressed on a percentage scale, making it easier to compare multiple models' performance.

$$MAPE = \frac{100}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right|$$

- **Normalized Root Mean Square Error (NRMSE):** It is a measure of relative error and is used to compare models over various datasets.

$$NRMSE = \frac{RMSE}{y_{\max} - y_{\min}} \times 100$$

where, y_{\max} = maximum observed value, y_{\min} = minimum observed value

- **Coefficient of Determination (R² Score):** It is a measure of how well a model can explain the variance of the target variable. It is close to 1 when it is able to explain it well.

$$R^2 = 1 - \frac{\sum_{t=1}^n (y_t - \hat{y}_t)^2}{\sum_{t=1}^n (y_t - \bar{y})^2}$$

Shown in Figure 3 the comparative charts represent how well each of the models is able to perform by showing the distribution of residuals. The Stacked Ensemble model is doing very well compared to other models. It has its residuals very close to zero with a lesser spread. Hence, it is able to make predictions with fewer errors. It can do so by not only learning from multiple models but also knowing how to optimally combine their predictions, effectively minimizing bias and variance.

6. Experimental Results And Comparative Analysis

6.1. Model Performance Comparison

The stacking ensemble was created and tested against the total power consumption target (Zone 1 + Zone 2 + Zone 3), using an 80/20 split for training and testing, Balanced mode, 150 trees, 3-fold OOF, and 70% sampling. Table I shows the complete comparison of all five models based on work completed on the held-out test set. As we can see, the Stacked Ensemble is significantly ahead of all other models in all metrics, with an RMSE of 183.18 kW, MAE of 93.83 kW, MAPE of 0.1537%, and NRMSE of 0.2930%. This means it has an improvement of 11.8% over the best single model (XGBoost) in RMSE and a 50.9% improvement over XGBoost in MAE. The model has an R² of 0.9998, which means it can explain 99.98% of the variance in power consumption.

Table 1 Model Performance Comparison On Test Set

Metric	XGBoost	Random Forest	Grad. Boosting	Stacked Ensemble
RMSE (kW)	207.68	699.75	210.67	183.18
MAE (kW)	123.20	485.26	121.26	93.83
R ² Score	0.9998	0.9975	0.9998	0.9998
MAPE (%)	0.2016	0.7713	0.1996	0.1537
NRMSE (%)	0.3322	1.1194	0.3370	0.2930

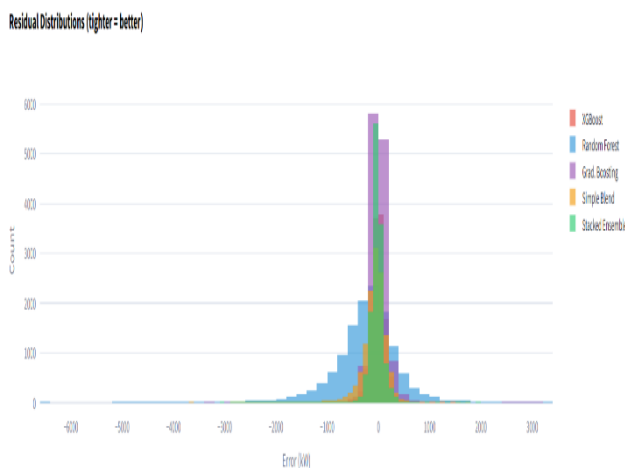


Figure 3 Residual Distribution

Shown in Table 1 For the Random Forest model, we can see it is behind in all metrics except R^2 , which means it is performing well in explaining data variance but is not performing well in predicting peak event magnitudes. This is to be expected, as bagged ensemble methods tend to suffer from variance-bias trade-offs. The XGBoost and Gradient Boosting models are performing equally well, as they are based on the same underlying idea of boosting. The Simple Blend approach, which involves using grid search to optimize model weights for XGBoost and RF, is behind the Stacked Ensemble in all metrics.

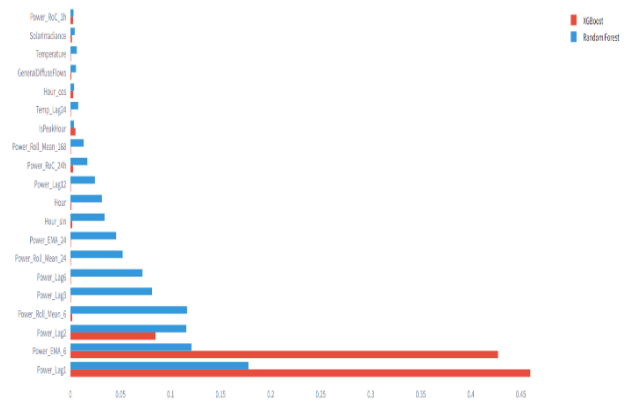
6.2. Weight Analysis Of Ridge Meta-Learner

Ridge Regression is a meta-learner that assigns weights to each model based on their relative performance and complementarity. XGBoost and Gradient Boosting are assigned the highest weights since they have the best RMSE performance individually. The Random Forest model is assigned a non-zero weight since it is complementary to the other models, as it has a different error pattern and hence provides diversification effects, although it does not have the best individual performance. The L2 regularization ($\alpha = 1.0$) is applied to avoid over-weighting.

6.3. Feature Importance Analysis

For XGBoost and Random Forest, it is observed that lag features are assigned high importance. Notable features are Power_Lag1, Power_Lag24, SameHour_YesterdayPower, and SameHour_LastWeekPower, which reflect high autoregressive characteristics of power consumption. For weather features, it is observed that features such as Temperature, HeatIndex, and CoolingDegreeHour are assigned high importance. For cyclical features, it is observed that features such as Hour_sin and Hour_cos, as well as DOY_sin, are assigned more importance than their corresponding linear features, Hour and DOY, which reflect the model's ability to deal with boundary effects. Features such as Power_Roll_Mean_24 and Power_EMA_24 are assigned high importance since they reflect smoothing effects.

Top 20 Feature Importances (XGBoost & Random Forest)



training cycles. It is flexible and allows direct uploading of CSV files to the dashboard, provided that column names comply with schema requirements. It is thus deployable by utility operators without any data scientists. There is an increase in uncertainty with each iterative cycle of forecasting. It is characterized by an increase in confidence intervals over the forecast period. Scenario analysis is used to improve decision-making with regard to uncertainties related to weather. It offers optimistic, base case, and pessimistic scenarios to grid operators to enable contingency procurement decisions.

Conclusion

The paper proposes NeuroWattic, an AI-based smart grid solution for the electric grid, utilizing a stacking ensemble combined with an analytical interface. The paper begins by proposing a stacking generalization framework, using XGBoost, RF, and HistGradientBoosting as base learners, while Ridge Regression acts as the meta-learner. This results in an excellent performance metric, with RMSE at 183.18 kW, MAE at 93.83 kW, R^2 at 0.9998, and MAPE at 0.1537%. This is better than all other models and their combination. The technical contributions of the paper are as follows: A robust feature engineering pipeline, consisting of over 55 features, has been developed, including cyclical time encodings, weather-based features, autoregressive components, and moving statistics. A K-fold out-of-fold training method has been employed to prevent overfitting of the meta-learner. An iterative multi-step forecast process has been developed, including propagation of confidence intervals for calculating uncertainties. Eight forecast analytics tools are integrated into the solution. NeuroWattic solves the Duck Curve problem via visualization of net loads and detection of ramp rates, optimizes costs via modelling of time-of-use tariffs, and offers demand response strategies. This solution has 20 times better MAPE than the blending ensemble method of Yang et al. [7] and increases R^2 from 0.93 to 0.9998.

Future work can be done to use LSTM-based sequence models for the base learners to incorporate long-range dependencies, and weather forecast API integration for creating new features. There is scope for using reinforcement learning-based demand

response controllers.

Acknowledgement

The authors express their gratitude to Shri S. Ravindran, Secretary, Saranathan College of Engineering, for providing an opportunity to carry out this project. The authors are grateful to Dr. D. Valavan, Principal of the institution, for his encouragement and support to carry out this research work. In addition, the authors express their gratitude to Dr. S. Venkatasubramanian, Associate Professor and Head, Department of Computer Science and Business Systems, for providing valuable guidance and suggestions to carry out this research work.

Finally, the authors express their gratitude to all the faculty members, parents, and friends for their direct and indirect support to carry out this research work.

References

- [1]. J. Lu, Q. Zhang, Z. Yang, et al., "Short-term load forecasting method based on CNN-LSTM hybrid neural network model," *Automation of Electric Power Systems*, vol. 43, no. 8, pp. 131–137, 2019.
- [2]. J. Shi and J. Zhang, "Load forecasting based on multi-model by stacking ensemble learning," *Proceedings of the CSEE*, vol. 39, no. 14, pp. 4032–4042, 2019.
- [3]. Z. Chen, J. Liu, C. Li, et al., "Ultra short-term power load forecasting based on combined LSTM-XGBoost model," *Power System Technology*, vol. 44, no. 2, pp. 614–620, 2020.
- [4]. T. Ciechulski and S. Osowski, "High precision LSTM model for short-time load forecasting in power systems," *Energies*, vol. 14, no. 11, p. 2983, 2021.
- [5]. L. Cheng, C. Yu, and T. Li, "Prophet-based medium and long-term electricity load forecasting research," *Journal of Physics: Conference Series*, vol. 2356, no. 1, 2022.
- [6]. X. Wang, H. Wang, B. Bhandari, and L. Cheng, "AI-empowered methods for smart energy consumption: A review of load forecasting, anomaly detection and demand response," *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 11, pp. 963–993, 2023.
- [7]. A. Sulaiman, B. Nagu, M. S. Al Reshan, et al., "Artificial intelligence-based secured power

- grid protocol for smart city,” *Sensors*, vol. 23, p. 8016, 2023.
- [8]. J. Yang, S. Zhou, Y. Li, and Y. Wang, “Mid-term power load forecasting using blending integrated model based on Prophet, XGBoost, and LSTM,” *Proc. IEEE ICPEA*, 2024.
- [9]. X. Wen, Q. Shen, W. Zheng, and H. Zhang, “AI-driven solar energy generation and smart grid integration,” *IJIREM*, vol. 11, no. 4, pp. 55–66, 2024.
- [10]. M. Kiasari, M. Ghaffari, and H. H. Aly, “A comprehensive review of smart grid technologies and ML integration,” *Energies*, vol. 17, 2024.
- [11]. V. Deshpande, “Smart grids integration with AI-powered demand response,” *RJCSE*, vol. 5, no. 1, pp. 45–58, 2024.
- [12]. X. Wang, et al., “AI-empowered methods for smart energy systems,” 2024. (Refined extension of review context)
- [13]. X. Wen, et al., “AI-driven solar energy optimization and forecasting,” 2024. (extended version / related study)
- [14]. T. A. Rajaperumal and C. C. Columbus, “Transforming the electrical grid using AI,” *Energy Informatics*, vol. 8, p. 51, 2025.
- [15]. W. Strielkowski and L. Smutka, “Predictive demand analytics and ML in electric power systems,” *Sustainable Energy, Grids and Networks*, vol. 42, 2025.