

AI-Driven Energy Management Platform Using IoT and Machine Learning

¹Balakarthish T, ²Sushma A

Department of Computer Science, Rathinam College of Arts and Science (Autonomous), Coimbatore,
Tamil Nadu, India

Corresponding Author: balakarthish2003@gmail.com

Electricity demand is escalating rapidly owing to urbanization, industrial growth, and the pervasive adoption of electrical devices, making efficient power management a critical societal challenge. This paper presents an AI-driven Power Management System that leverages machine learning techniques to forecast future electricity demand and identify peak load periods. The system employs the Gradient Boosting algorithm, which analyzes historical power consumption data alongside time-based and environmental features—including hour of day, day of month, month of year, and ambient temperature—to generate accurate predictions. Advanced preprocessing pipelines and feature engineering strategies, including lag features and rolling averages, are applied to enhance model accuracy. The system produces monthly electricity demand forecasts and provides proactive peak load identification. An interactive Streamlit-based dashboard allows users to upload datasets, apply temporal filters, and visualize results through charts and summary statistics, thereby supporting data-driven decision-making. The proposed platform contributes to reducing electricity wastage, optimizing energy distribution, improving grid stability, and facilitating the efficient utilization of renewable energy sources during peak demand periods. Experimental results demonstrate the viability of the approach for intelligent, scalable, and sustainable modern energy management.

I. INTRODUCTION

Electricity is among the most indispensable resources underpinning modern civilization. Its consumption is growing at an unprecedented rate due to rapid urbanization, industrial expansion, and the proliferation of consumer electronic devices [1]. Efficient power management has consequently emerged as a pressing challenge for both energy providers and end consumers. Traditional energy management systems predominantly depend on manual monitoring, fixed scheduling, and historical-average-based estimation—approaches that are fundamentally insufficient to accommodate dynamic and unpredictable consumption patterns [2].

The last decade has witnessed the emergence of Artificial Intelligence (AI) and Machine Learning (ML) as transformative tools for large-scale data analysis and high-fidelity prediction [3]. These paradigms empower intelligent systems to learn from historical observations, extract latent patterns, and generate forecasts with high accuracy. In the specific context of energy management, AI can be applied to predict electricity demand, detect impending peak load conditions, and dynamically optimize power usage to avert wastage and grid instability [4].

The Internet of Things (IoT) further amplifies this capability by enabling continuous, real-time data collection from distributed sensors deployed in residential, commercial, and industrial environments [5]. Integrating IoT-sourced streaming data with ML-based forecasting models yields a coherent platform for proactive energy management.

This paper presents a comprehensive AI-driven Energy Management Platform that combines the Gradient Boosting algorithm with an interactive web-based dashboard to deliver accurate electricity demand forecasting and peak load detection. The principal contributions of this work are:

- A feature-rich preprocessing pipeline incorporating temporal decomposition, lag features, and rolling statistical

aggregates.

- Implementation of a Gradient Boosting Regressor for non-linear electricity demand forecasting.

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the system architecture and methodology. Section IV covers implementation details. Section V presents experimental results and discussion. Section VI concludes the paper and outlines future directions.

II. RELATED WORK

Electricity demand forecasting has attracted considerable research attention over the past two decades. Early approaches relied on statistical methods such as ARIMA (Auto-Regressive Integrated Moving Average) [6] and exponential smoothing. While these models perform adequately under stationary conditions, they struggle with the inherently non-linear and multivariate nature of modern electricity consumption patterns.

The advent of machine learning introduced regression-based approaches including Support Vector Regression (SVR) [7] and Random Forests [8], which demonstrated improved accuracy by capturing feature interactions. Ensemble learning methods, particularly boosting-based algorithms, have subsequently shown superior performance on tabular energy datasets. Chen and Guestrin [9] demonstrated that gradient boosting frameworks achieve state-of-the-art results across diverse regression tasks, including energy prediction.

Deep learning architectures such as Long Short-Term Memory (LSTM) networks [10] have been applied to time-series energy forecasting, exploiting sequential dependencies across extended temporal horizons. However, such models require large training datasets and substantial computational resources, limiting their applicability in resource-constrained environments.

IoT-integrated energy management platforms have been proposed by several researchers. Ref. [11] introduced a smart

grid architecture leveraging IoT sensor networks for real-time load balancing. Ref. [12] applied reinforcement learning to residential energy management systems, achieving meaningful reductions in peak demand. Despite these advances, a practical and accessible open-source platform combining ML-based

forecasting with an interactive visualization layer remains underexplored, motivating the present work.

III. SYSTEM ARCHITECTURE AND METHODOLOGY

A. Overall Architecture

The proposed platform follows a three-tier architecture comprising a *Data Layer*, a *Processing and Intelligence Layer*, and a *Presentation Layer*, as described below.

1) *Data Layer*: The data layer is responsible for ingesting, storing, and organizing raw power consumption data. Historical consumption records are stored in CSV format, a lightweight and universally supported medium that integrates seamlessly with Python data processing libraries. Each record captures the following primary attributes: (i) *Date* (date of the observation), (ii) *Hour* (hour of day, 0–23), (iii) *Temperature* (ambient temperature in °C), and (iv) *Power_Consumption* (electricity usage in kW/kWh).

2) *Processing and Intelligence Layer*: This layer encapsulates the preprocessing pipeline, feature engineering module, and machine learning model. Input data flows through validation, null-value treatment, temporal decomposition, and feature augmentation before being presented to the Gradient Boosting model for training or inference.

3) *Presentation Layer*: The Streamlit-based dashboard con-

stitutes the presentation layer. It provides an intuitive interface for dataset upload, temporal filtering, and the display of both

raw trends and model predictions through interactive charts and summary tables.

B. Data Preprocessing

Raw electricity consumption datasets frequently contain missing entries, outliers, and inconsistent formatting. The preprocessing pipeline addresses these issues through the following sequential steps:

- 1) **Data Loading**: The CSV dataset is loaded using `pandas.read_csv()`, and the *Date* column is parsed into datetime objects.
- 2) **Null Value Handling**: Rows containing null or NaN values are identified and removed. In production deployments, forward-fill or interpolation strategies may be applied.
- 3) **Outlier Detection**: Extreme consumption values beyond three standard deviations from the rolling mean are flagged and optionally capped.
- 4) **Data Type Validation**: Numeric columns are cast to float64 to prevent type-mismatch errors during model training.

C. Feature Engineering

Raw temporal and environmental attributes alone provide

limited discriminative signal for a machine learning model. The feature engineering module constructs an enriched feature

- **Weekend Indicator**: A binary flag *Is_Weekend* is assigned (1 if weekday ≥ 5 , else 0) to capture the weekend consumption drop typical in commercial and industrial loads.
- **Lag Feature (Prev_Load)**: The power consumption value of the immediately preceding hour is appended. This feature encodes short-term temporal autocorrelation in the data.
- **Rolling Average (Rolling_Avg)**: A 3-hour backward rolling mean of power consumption is computed to capture smoothed local trends and reduce noise sensitivity.

The full feature vector \mathbf{x} used for prediction is therefore:

$$\mathbf{x} = [\text{Hour}, \text{Day}, \text{Month}, \text{Weekday}, \text{Temp}, \text{Is_Weekend}, \text{PrevLoad}, \text{Rol}]$$

Rol

D. Machine Learning Model: Gradient Boosting

Gradient Boosting is an ensemble learning technique that constructs an additive model in a forward stage-wise fashion [13]. At each stage m , a weak learner (typically a shallow decision tree) $h_m(\mathbf{x})$ is fitted to the negative gradient of the loss function L with respect to the current ensemble prediction $F_{m-1}(\mathbf{x})$:

$$\frac{\partial L(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}$$

$$r_{im} = - \frac{\partial L(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)} \quad F = F_{m-1}$$

The ensemble is updated as:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \eta \cdot h_m(\mathbf{x})$$

where $\eta \in (0, 1]$ is the learning rate that controls the contribution of each tree and serves as a regularization mechanism against overfitting. The final prediction after M iterations is:

$$\hat{y} = F_M(\mathbf{x}) = F_0(\mathbf{x}) + \sum_{m=1}^M \eta \cdot h_m(\mathbf{x})$$

For the mean squared error (MSE) loss—appropriate for regression—the negative gradient in Eq. (2) reduces to the residual $r_{im} = y_i - F_{m-1}(\mathbf{x}_i)$.

In the proposed system, the Scikit-learn `GradientBoostingRegressor` is configured with $M = 200$ estimators, $\eta = 0.05$, and a maximum tree depth of 4, striking a balance between model expressiveness and overfitting risk.

E. Peak Load Detection

Following monthly demand forecasting, the system identifies the peak load interval by locating the maximum predicted value within the forecast horizon:

$$t^* = \arg \max_{t \in \{1, \dots, T\}} \hat{y}_t$$

set as follows:

- **Temporal Decomposition:** The *Day*, *Month*, and *Week-day* are extracted from the *Date* field using pandas dt acces

(e.g., all-weekend periods, extreme temperatures).

V. RESULTS AND DISCUSSION

A. Model Performance

The trained Gradient Boosting model was evaluated on the held-out 20% test partition. Table III summarizes the performance metrics obtained.

An R^2 score of 0.9312 indicates that the model explains approximately 93% of the variance in electricity consumption, demonstrating strong predictive capability. The inclusion of lag

TABLE III
 MODEL EVALUATION METRICS ON TEST SET

Metric	Value
Mean Absolute Error (MAE)	18.43 kW
Root Mean Squared Error (RMSE)	27.61 kW
Coefficient of Determination (R^2)	0.9312

features and rolling averages contributed substantially to this accuracy, as the model could leverage short-term autocorrelation in the consumption series.

B. Feature Importance

Analysis of the Gradient Boosting model’s feature importances revealed that *Prev_Load* (previous-hour consumption) and *Rolling_Avg* were the two most influential predictors, accounting for approximately 54% of total feature importance. Temporal features *Hour* and *Month* ranked third and fourth, reflecting the strong diurnal and seasonal periodicities inherent in electricity demand. *Temperature* exhibited a moderate importance score, consistent with its known correlation with cooling and heating loads. *Is_Weekend* showed the lowest importance, suggesting that the lag features already implicitly encode consumption regime differences between weekdays and weekends.

C. Monthly Forecast and Peak Detection

The system generated a 30-day (720-hour) forward forecast for April. The predicted peak load was identified at Day 14, Hour 14:00, with a forecasted consumption of 487.23 kW. This peak coincided with a simulated high-temperature afternoon scenario, validating the model’s temperature-load relationship capture.

The monthly average predicted consumption was 374.6 kW, with a standard deviation of 51.2 kW, reflecting realistic diurnal fluctuation patterns.

D. Comparison with Baseline Methods

Table IV compares the Gradient Boosting model against two baselines—a simple historical mean estimator and a linear regression model—evaluated on the same test set.

TABLE IV
 COMPARISON OF FORECASTING APPROACHES

Method	MAE (kW)	RMSE (kW)	R^2
Historical Mean	61.7	84.2	0.421
Linear Regression	34.1	46.8	0.763
Gradient Boosting	18.4	27.6	0.931

The Gradient Boosting model consistently outperforms both baselines across all metrics, confirming the benefit of ensemble-based non-linear modeling for electricity demand

E. Dashboard Usability

The interactive Streamlit dashboard was evaluated in informal usability sessions with five non-technical users. All participants successfully uploaded a dataset, applied a date range filter, and interpreted the resulting visualizations without any external guidance. The average task completion time was under 3 minutes, affirming the dashboard’s accessibility.

VI. CONCLUSION AND FUTURE WORK

This paper presented an AI-driven Energy Management Platform that integrates Gradient Boosting-based electricity demand forecasting with an interactive Streamlit dashboard. The system demonstrated strong predictive performance ($R^2 = 0.931$), accurate monthly peak load identification, and intuitive usability, making it a viable candidate for deployment across residential, commercial, and industrial settings.

The key contributions include a robust feature engineering pipeline exploiting temporal decomposition and autoregressive features, a quantitative comparison establishing the superiority of Gradient Boosting over baseline estimators, and a user-centric dashboard design validated through usability evaluation.

Future Work

Several avenues merit exploration in future research:

- **Deep Learning Integration:** Incorporating LSTM or Transformer-based sequence models [14] for extended multi-step forecasting horizons.
- **Real-Time IoT Data Integration:** Connecting the platform to live sensor streams via MQTT or HTTP APIs to enable online model updating.
- **Renewable Energy Scheduling:** Extending the system with an optimization module that schedules solar/wind dispatch against the predicted demand curve.
- **Cloud Deployment:** Deploying the platform on scalable cloud infrastructure (e.g., AWS, Google Cloud) to support large-scale, multi-site energy management.
- **Federated Learning:** Adopting federated learning protocols to train models across distributed energy consumers while preserving data privacy.

ACKNOWLEDGMENT

The author sincerely thanks Ms. Sushma A. (Guide, Department of Computer Science), Dr. T. Velumani (Head, Department of Computer Science), and the management of Rathinam College of Arts and Science (Autonomous), Coimbatore, for their invaluable guidance and support throughout the duration of this project.

forecasting.

REFERENCES

- [1] International Energy Agency, "World Energy Outlook 2023," IEA, Paris, 2023. [Online]. Available: <https://www.iea.org/reports/world-energy-outlook-2023>
- [2] M. M. Rahman, M. A. Hossain, and M. A. Islam, "Limitations of Traditional Energy Management Systems in Smart Grid Environments," *Energy Reports*, vol. 6, pp. 1122–1131, 2020.
- [3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
- [4] T. Hong and S. Fan, "Probabilistic Electric Load Forecasting: A Tutorial Review," *International Journal of Forecasting*, vol. 32, no. 3, pp. 914–938, Jul. 2016.
- [5] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
- [6] 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.
- [7] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer, 1995.
- [8] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
- [9] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 785–794.
- [10] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
- [11] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart Grid – The New and Improved Power Grid: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 944–980, 2012.
- [12] E. Mocanu *et al.*, "On-Policy Deep Reinforcement Learning for Electrical Energy Management," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 1960–1970, Mar. 2018.
- [13] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.
- [14] A. Vaswani *et al.*, "Attention Is All You Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, Long Beach, CA, USA, Dec. 2017, vol. 30, pp. 5998–6008.
- [15] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.