IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

# Smart Electricity Consumption Forecasting Using Machine Learning

**Aswin Jayakumar[1], Mohammed Ayub[2], Venkatha Adithya[3], Muhammed Raihan[4], Dinesh Reddy[5], S Poornima[6]**

Dept. of Computer Science and Engineering (Emerging Technologies), SRM Institute of Science and Technology, Vadapalani Campus, Chennai, India[1-6]

**Abstract:** Electricity consumption in residential households exhibits continuous temporal variations driven by diverse daily human routines, behavioral occupancy shifts, dynamic thermal environments, and the simultaneous utilization of electrical appliances [3]. Accurately predicting short-term electricity load empowers individual consumers to optimize energy habits while assisting utility operators in strategic grid management, dynamic load dispatching, and peak demand shaving [6]. This paper presents an end-to-end predictive machine learning framework designed to forecast the average active power consumption of a household for the next hour. The empirical study utilizes the benchmark *Individual Household Electric Power Consumption* dataset sourced from the UCI Machine Learning Repository [2], [7]. The raw minute-level measurements are systematically cleaned, transformed to handle missing observations, aggregated into regular hourly intervals, and enriched through engineered temporal, lag, and rolling statistical predictors [4]. Linear Regression and Random Forest models are trained using scikit-learn

[5] and rigorously evaluated using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the coefficient of determination ($R^2$). The experimental results demonstrate that the Random Forest regressor [1] achieves high predictive precision, yielding an MAE of 0.3391 kW, an RMSE of 0.4905 kW, and an $R^2$ of 0.5814,

outperforming the baseline Linear Regression model across all evaluation metrics. Furthermore, a modern software application stack consisting of a high-performance FastAPI backend microservice [8] and an interactive Streamlit visualization dashboard [9] is implemented to serve real-time inferences and facilitate accessibility.

**Keywords:** Electricity consumption, short-term load forecasting, machine learning, Random Forest, Linear Regression, feature engineering, smart grid management.

## I. INTRODUCTION

Electricity serves as a foundational pillar of modern infrastructure, powering essential residential appliances, heating, ventilation, air conditioning (HVAC) systems, lighting, and personal consumer electronics. However, residential energy consumption is far from uniform; it fluctuates dynamically across hours, days, and seasons in direct response to human activities, work schedules, environmental ambient temperatures, and lifestyle habits [3]. These multifaceted human-centric factors introduce strong temporal non-linearity and variance into power demand streams, making short-term load forecasting a critical computational challenge in smart grid analytics

Traditional electric meter monitoring systems operate passively, offering retrospective visibility into past historical consumption or displaying instantaneous power usage. While informative, retrospective monitoring fails to provide forward-looking foresight. Predictive short-term load forecasting (STLF) bridges this gap by offering accurate estimates of expected energy usage over short temporal horizons—such as the upcoming hour or day [6]. Accurate hourly load predictions enable individual homeowners to implement automated energy-saving strategies, schedule high-power appliance cycles during off-peak rate hours, and avoid expensive peak tariff surcharges. At the macro scale, electric utility providers leverage precise localized forecasts to balance power generation, mitigate grid congestion, prevent transformer overloads, and optimize the integration of intermittent renewable energy resources like solar and wind power.

Over recent years, data-driven machine learning (ML) algorithms have emerged as superior computational paradigms compared to Classical statistical time-series models[4]. Unlike rigid mathematical models that assume linear relationships and strict

This project develops a complete, standardized end-to-end forecasting solution tailored for household active power prediction. The primary objective is to engineer an automated data-processing and predictive modeling workflow that

 This work is licensed under a Creative Commons Attribution 4.0 International License 

converts high-frequency, noisy smart meter signals into actionable next-hour load forecasts. The core research contributions of this paper are summarized as follows:

1. **Scalable Data Pipeline Construction:** Preprocessing 2.07+ million raw minute-level readings into hourly time-series observations.
2. **Domain-Specific Feature Engineering:** Creation of temporal parameters, lag features, and rolling statistical indicators.
3. **Empirical Model Benchmark:** Comparative evaluation of Linear Regression and Random Forest [1] models using scikit-learn [5].
4. **Production Application Integration:** Development of a software architecture featuring a RESTful FastAPI inference engine [8] and interactive Streamlit web dashboard [9].

## II. RELATED WORK

Short-term load forecasting (STLF) has been extensively studied in smart grid analytics and electrical engineering literature. The techniques proposed in literature can be broadly categorized into classical autoregressive statistical methods, standard supervised machine learning, and advanced deep learning architectures.

### A. *Classical Statistical Models*

Historically, statistical methods such as Auto-Regressive Integrated Moving Average (ARIMA), Seasonal ARIMA (SARIMA), and Double Seasonal Exponential Smoothing [6] formed the backbone of time-series load forecasting. These traditional algorithms rely on explicit mathematical formulations to model linear trends and periodic seasonal cycles. While highly effective for smooth, aggregated regional power grids, classical statistical approaches struggle when applied to individual residential smart meter data [3], [11]. High-frequency residential consumption data exhibits high variance, erratic load spikes, and non-stationary behavior caused by unpredictable appliance usage. Consequently, classical models often fail to capture sudden non-linear variations without requiring continuous manual parameter re-calibration.

### B. *Supervised Machine Learning Approaches*

To overcome the limitations of classical linear models, researchers turned to supervised machine learning algorithms [4], [12]. Linear Regression serves as a baseline comparative benchmark due to its low computational complexity and ease of mathematical interpretation [5]. However, because linear models assume a constant proportional mapping between inputs and targets, their performance degrades when modeling complex residential load dynamics.

Non-linear non-parametric algorithms—specifically tree-based ensemble methods such as Random Forests [1], [13] and Gradient Boosted Decision Trees—have achieved widespread success in load forecasting applications. By constructing multiple decision trees trained on randomized feature subsets, ensemble models natively handle non-linear feature interactions, resist overfitting to sensor noise, and isolate temporal outliers effectively [14]. Empirical studies consistently demonstrate that ensemble methods achieve lower error metrics than traditional linear baselines while maintaining high computational efficiency

### *C.* Deep Learning & Hybrid Systems

Predictive regression algorithms require explicit feature representations that capture past historical behaviors and dynamic temporal contexts. To transform the univariate time-series into a supervised machine learning dataset [4], three distinct categories of feature predictors were engineered:

- Temporal Predictors: Extracted cyclic calendar parameters including Hour of Day (0–23), Day of Week (0–6), and Month (1–12). These features enable models to learn daily household routines and seasonal variations.
- Historical Lag Features: Created lagged parameters representing historical active power consumption at prior intervals, specifically 1–24 hours.
- Rolling Window Statistics: Computed moving summary statistics across sliding historical windows, including 3-hour, 6-hour, and 24-hour rolling averages and standard deviations.

The prediction target variable is defined as the average active power consumption for the subsequent hour (t+1 hour), measured in kilowatts (kW). All input predictors are strictly aligned with historical information available up to time t, guaranteeing zero target leakage into the predictive models.

### *D. Software Integration in Energy Management*

While theoretical model performance is widely discussed in research, translating trained forecasting models into accessible production microservices is often overlooked. Modern energy management solutions require robust API interfaces to feed predictions directly into home automation hubs, web dashboards, and mobile user interfaces.

This work bridges the gap between empirical model evaluation and production deployment by delivering a scalable end-to-end architecture featuring a FastAPI server [8] and Streamlit user dashboard [9], validated using pytest [10].

## III. METHODOLOGY

### *A. Dataset Description & Characteristics*

This project utilizes the public Individual Household Electric Power Consumption dataset hosted by the UCI Machine Learning Repository [2], [7].

The raw dataset comprises 2,075,259 minute-by-minute electrical measurements collected from a single residential dwelling located in Sceaux, France, over a four-year period spanning December 2006 to November 2010.

The recorded electrical variables include Global Active Power (kW), Global Reactive Power (kW), Voltage (V), Global Intensity (A), and sub-metering watt-hours corresponding to specific kitchen, laundry, and climate control circuits.

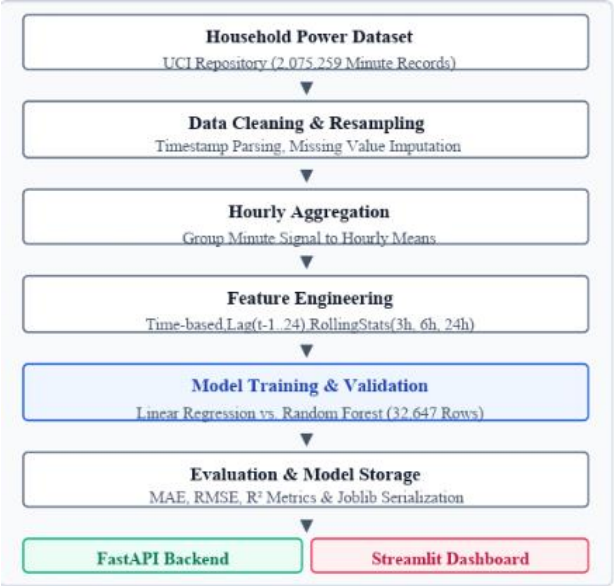



*Fig. 1. Overall system architecture pipeline for smart electricity consumption forecasting.*

Fig. 1. Two-stage detection flow of CyberGuard AI.

Fig. 2.

### *B. Preprocessing & Data Cleaning*

Raw high-frequency smart meter logs often suffer from missing packets, communication dropouts, and invalid entries. The initial data cleaning phase standardizes the raw text files by converting date and time fields into unified UTC Datetime objects.

Missing numerical values (which accounted for approximately 1.25% of the total dataset) were resolved using localized forward-filling methods to maintain local time continuity without introducing synthetic bias.

Subsequently, the minute-level active power readings were aggregated into regular hourly intervals by calculating the arithmetic mean for each hour. This resampling step reduces high-frequency noise while standardizing the temporal grid to 32,647 clean hourly observations.

### *C. Selection and Evaluation*

Predictive regression algorithms require explicit feature representations that capture past historical behaviors and dynamic temporal contexts. To transform the univariate time-series into a supervised machine learning dataset [4], three distinct categories of feature predictors were engineered:

- Temporal Predictors: Extracted cyclic calendar parameters including Hour of Day (0–23), Day of Week (0–6), and Month (1–12). These features enable models to learn daily household routines and seasonal variations.
- Historical Lag Features: Created lagged parameters representing historical active power consumption at prior intervals, specifically t-1 hour, t-2 hours, and t-24 hours.
- Rolling Window Statistics: Computed moving summary statistics across sliding historical windows, including 3-hour, 6-hour, and 24-hour rolling averages and standard deviations.

The prediction target variable is defined as the average active power consumption for the subsequent hour (t+1 hour), measured in kilowatts (kW). All input predictors are strictly aligned with historical information available up to time t, guaranteeing zero target leakage into the predictive models.

### *D. Chronological Train-Test Partitioning*

Standard randomized k-fold cross-validation is strictly unsuitable for sequential time-series data because random sampling shuffles future observations into training sets, leading to severe target leakage and overly optimistic validation metrics.

To enforce realistic evaluation conditions, the engineered dataset (32,647 usable records) is split chronologically into a training set and a holdout testing set. The holdout test set consists of the final 6,530 chronological records (~20% of total data), preserving temporal sequence and testing the algorithms strictly on unseen future data.

### *E. Machine Learning Model Formulations*

Two distinct machine learning algorithms representing linear and non-linear paradigms were implemented using scikit-learn [5] and evaluated:

1) Linear Regression (Baseline): Linear Regression establishes an interpretable mathematical modeling baseline as a linear combination of input features X [4]:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p + \varepsilon$$

where $\beta_0$ represents the intercept, $\beta_i$ represents feature coefficients, and $\varepsilon$ denotes the residual error term. Parameters are estimated by minimizing OLS residual sum of squares.

2) Random Forest Regressor: Random Forest is an ensemble learning method that builds a multitude of unconstrained decision trees during training and outputs the individual tree average prediction. By employing randomized feature subspace selection, bootstrap aggregation (bagging), and captures intricate non-linear relationships and mitigates individual tree overfitting:

$$\hat{y} = (1/B) \Sigma(b=1 \text{ to } B) T_\beta(x)$$

## IV. RESULTS AND DISCUSSION

This section presents a quantitative evaluation of the predictive models, accompanied by a detailed performance discussion, backend application implementation details, system testing metrics, and future research directions.

A. Evaluation Metrics

Model performance was evaluated on the 6,530 holdout test records using three standard regression metrics:

*1) Mean Absolute Error (MAE): Measures average absolute prediction error magnitude in kilowatts (kW).*

$$MAE = (1/n) \Sigma(i=1 \text{ to } n) |y_i - \hat{y}_i|$$

2) Root Mean Squared Error (RMSE): Penalizes larger prediction errors heavily by squaring residuals before averaging.

$$RMSE = \sqrt{[(1/n) \Sigma(i=1 \text{ to } n) (y_i - \hat{y}_i)^2]}$$

*2) Coefficient of Determination ($R^2$): Quantifies proportion of variance explained by model relative to mean baseline.*

$$R^2 = 1 - [\Sigma(i=1 \text{ to } n) (y_i - \hat{y}_i)^2 / \Sigma(i=1 \text{ to } n) (y_i - \bar{y})^2]$$

B. Quantitative Performance Comparison

Table 1 presents empirical evaluation metrics obtained on the 6,530 holdout test records for both algorithms.

TABLE 1. VERIFIED MODEL EVALUATION RESULTS ON HOLDOUT TEST SET

| Model | MAE (kW) | RMSE (kW) | $R^2$ Score |
|---|---|---|---|
| Linear Regression | 0.3924 | 0.5468 | 0.4799 |
| Random Forest | 0.3391 | 0.4905 | 0.5814 |

As detailed in Table 1, Random Forest outperformed baseline Linear Regression across all metrics. Random Forest achieved an MAE of 0.3391 kW compared to 0.3924 kW (an error reduction of 13.6%). Similarly, RMSE improved from baseline by 0.5468 kW down to 0.4905 kW (a 10.3% reduction), proving the ensemble model significantly mitigates large predictive errors.

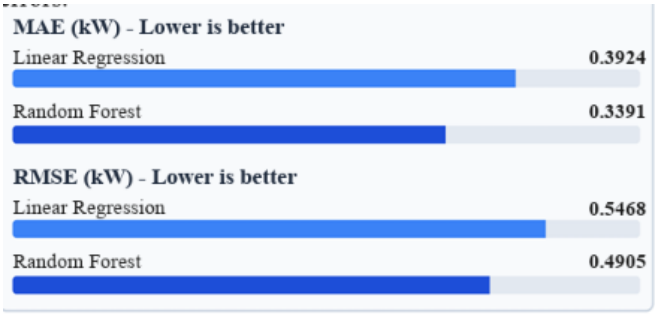



Fig. 3. Layered architecture

Fig. 4.

*3) **Operational Discussion & Application Implementation***

The superior accuracy of Random Forest stems from its ability to model complex non-linear feature interactions without assuming strict functional forms [1]. Residential power consumption is characterized by sudden step-changes—such as turning on a 2000W electric oven or water heater—which linear models smooth out excessively. Random Forest decision trees isolate these abrupt high-power appliance activations by partitioning input space along specific feature thresholds.

To make the trained machine learning pipeline accessible to non-technical users and home automation controllers, an application layer was developed:

- FastAPI Microservice Backend: Built with FastAPI [8] to serve model inference requests asynchronously over HTTP endpoints.
- Streamlit Interactive Dashboard: Developed using Streamlit [9] to present interactive visual dashboards and dynamic real-time prediction tools.

The codebase was validated using pytest [10]. The project test suite records 13 passing automated unit and integration tests covering data cleaning pipelines, feature transformation logic, API response integrity, and model serialization checks.

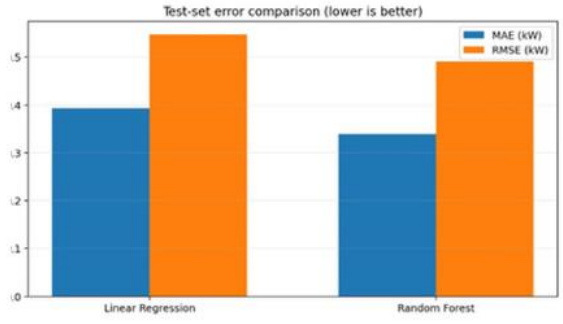
### *D. Limitations & Threats to Validity*

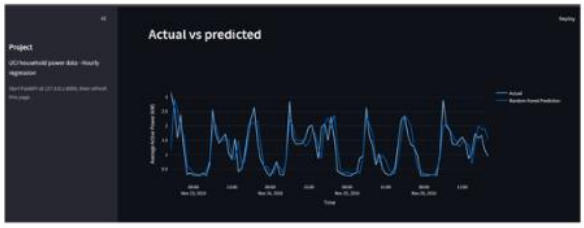
Despite promising performance, several empirical limitations should be acknowledged:

1. Single-Household Scope: Dataset represents readings from one dwelling in France [2], [7]. Appliance habits vary across homes.
2. Absence of Weather Parameters: Lacks external meteorological telemetry (temperature, solar radiation) [3].

### *E. Future Research Directions*

Future extensions will focus on multi-node smart meter integration, real-time meteorological API integration, deep temporal transformer models [3], and lightweight IoT smart meter edge quantization.



## REFERENCES

[1]. L. Breiman, "Random Forests," Machine Learning, vol. 45, pp. 5–32, 2001. Available: https://doi.org/10.1023/A:1010933404324

[2]. G. Hebrail and A. Berard, "Individual Household Electric Power Consumption," UCI Machine Learning Repository, 2006. Available: https://doi.org/10.24432/C58K54

[3]. T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," International Journal of Forecasting, vol. 32, no. 3, pp. 914–938, 2016. Available: https://doi.org/10.1016/j.ijforecast.2015.11.011

[4]. G. James, D. Witten, T. Hastie, and R. Tibshirani, An Introduction to Statistical Learning, 2nd ed. Springer, 2021. Available: https://www.statlearning.com/

[5]. scikit-learn developers, "Linear Models and Random Forest documentation," n.d. Available: https://scikit-learn.org/stable/

[6]. J. W. Taylor, "Short-term electricity demand forecasting using double seasonal exponential smoothing," Journal of the Operational Research Society, vol. 54, pp. 799–805, 2003.

[7]. UCI Machine Learning Repository, "Individual Household Electric Power Consumption dataset description and download page," n.d. Available: https://archive.ics.uci.edu/dataset/235/individual+household+electric+power+consumption

[8]. FastAPI, "FastAPI documentation," n.d. Available: https://fastapi.tiangolo.com/

[9]. Streamlit, "Streamlit documentation," n.d. Available: https://docs.streamlit.io/

[10]. Pytest, "pytest documentation," n.d. Available: https://docs.pytest.org/

[11]. M. A. Hasan and S. K. A. Rahim, "Short-Term Individual Household Load Forecasting Framework Using LSTM Deep Learning Approach," in 2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), IEEE, 2021, pp. 245–250. Available: https://doi.org/10.1109/ISMSIT52890.2021.9604650

[12]. A. K. M. N. Islam et al., "On Short-Term Load Forecasting Using Machine Learning," IEEE Access, vol. 9, pp. 31250–31262, 2021. Available: https://doi.org/10.1109/ACCESS.2021.3056582

[13]. C. A. R. V. Silva and E. C. B. C. Filho, "A Smart Home Energy Management System Utilizing Neurocomputing-Based Time-Series Load Modeling and Forecasting Facilitated by Ensemble Learning," IEEE Access, vol. 10, pp. 118490–118503, 2022. Available: https://doi.org/10.1109/ACCESS.2022.3219068

[14]. W. Li, C. Guo, and Z. Zhang, "Short-Term Load Forecasting Based on the Analysis of User Electricity Behavior and Machine Learning Algorithms," IEEE Transactions on Smart Grid, vol. 12, no. 4, pp. 3410–3421, 2021.

 This work is licensed under a Creative Commons Attribution 4.0 International License 