International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 07 | July - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Electricity Consumption - Monitoring and Forecasting

Ruchi Rautela [1], *Ganesh Deulkar* [1], *Raj Kaloge* [1], *Krishna Singh* [1]

[1]Dept of Masters of Computer Applications, Vivekanand Education Society's Institute Of Technology, Mumbai 400 074, Maharashtra.

Abstract

Accurate forecasting and monitoring electricity consumption has become very important for efficient energy management as well as grid optimization and strategic resource management. The following study aims to evaluate various machine learning models which included - Random Forest, XGBoost, Neural Networks along with a combined ensemble approach to enhance predictive accuracy, moreover it explores the impact of feature engineering and hyperparameter tuning in optimizing model performance. A synthetic dataset which spanned a duration of three years was utilized for the study. The dataset consisted of more than twenty-six thousand hourly records and incorporated temporal, environmental and historical consumption features. Data preprocessing involved standardization for neural networks and feature selection to optimize predictive performance. Among the models tested, XGBoost exhibited the highest accuracy, achieving a Mean Absolute Error (MAE) of 0.20 and an R^2 score of 0.59, outperforming the baseline and other approaches. The Ensemble model integrated Random Forest, XGBoost, and Neural Networks demonstrated robust performance and yielded stable predictions. Feature selection and importance analysis identified temperature and historical consumption as the most influential factors. Further, future improvements like leveraging advanced deep learning architecture such as LSTMs and introducing real-time IoT data can enhance the predictive capabilities and accuracy.

Introduction

Electricity has replaced other energy sources as the main source for usage in homes, businesses, and transportation[1]. Moreover its demand is increasing rapidly worldwide due to population growth, Industrial expansion, and new technologies like Artificial Intelligence (AI) and data centers. This rising demand is one of the reasons that we need smarter ways to monitor, analyse, manage and predict electricity consumption[1][3]. Not to mention that the current state of electricity supply is mired in a crisis. Energy crises are very common among countries [4]. Existing Traditional methods like Digital Meters (non-smart meters), Load Profiling through Utility Bills, Building Energy Audits, Supervisory Control and Data Acquisition (SCADA) Systems often fail to capture the complexities of energy usage pattern, leading to suboptimal predictions. Artificial Intelligence and Machine Learning (ML) technologies, particularly deep neural networks have demonstrated superior capabilities in handling time-series data and identifying intricate dependencies. By using predictive analysis, we can forecast future electricity usage based on past trends and help end-users to monitor electricity consumption in real-time and improve energy management by identifying unusual patterns and preventing wastage. These advancing technologies not only improve power reliability but also reduce costs and carbon emission. This study highlights the role of the above stated machine learning models, feature engineering and model optimization in enhancing forecasting precision and evaluating the same to find more similar alternatives for further improvements.

Literature Review

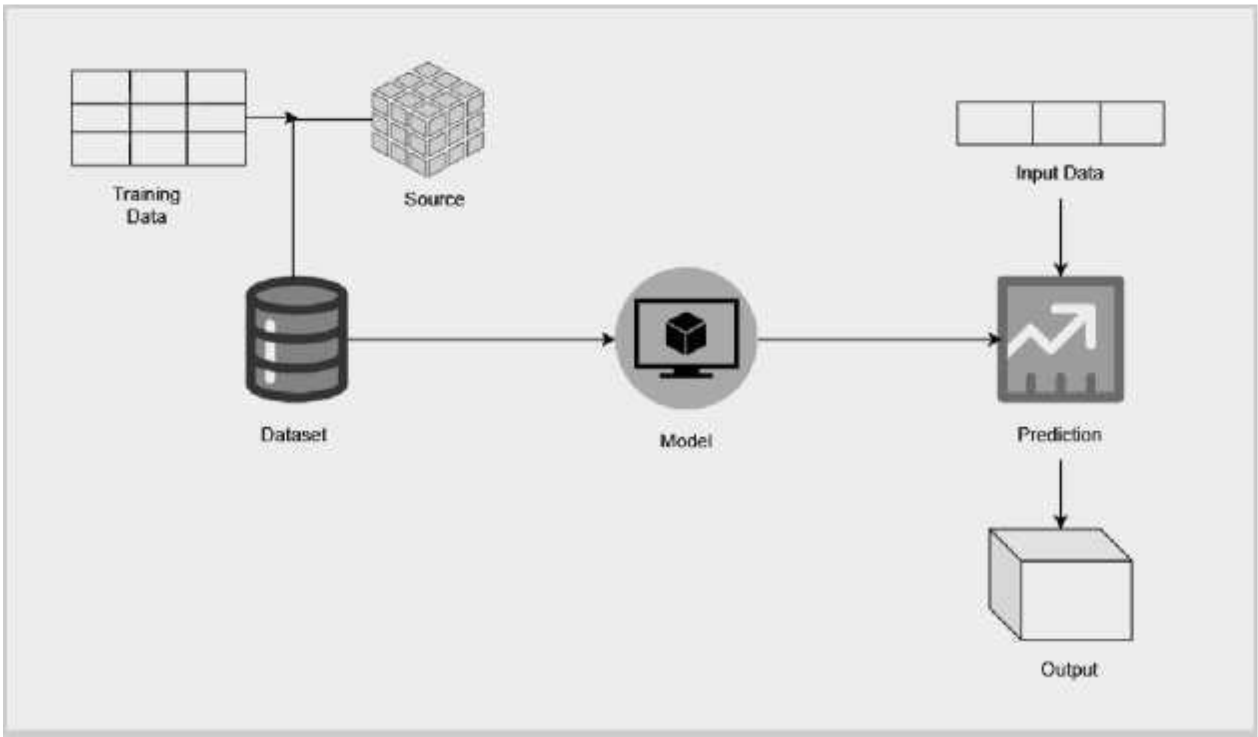
"Electricity Consumption Prediction Using Machine Learning" is a paper by Vijendar Reddy G., Suresh Angadi, and Ravi
S. Bhosale. It focuses on using machine learning algorithms to predict electricity consumption more accurately,

emphasizing the increasing need for efficient energy prediction in urban areas. The authors start by outlining the limitations of traditional statistical methods and justify the need for intelligent approaches capable of handling large, dynamic, and nonlinear datasets. According to the authors traditional forecasting methods often fall short in terms of accuracy and scalability, which leads to the exploration of even better machine learning techniques by acknowledging the critical role of accurate electricity consumption forecasting in energy management, environmental impact mitigation, and cost reduction.

For the study the authors have used references from many other prior key studies like - 1.Machine Learning Techniques for Load Forecasting – A review of decision trees, ANNs, SVMs, and ensemble methods for load forecasting, focusing on their strengths and limitations. 2.Integration of Meteorological and Climate Data – Studies showing improved forecasting accuracy by incorporating weather and climate data into machine learning models. 3. Hybrid Models – Research on combining ML approaches, like wavelet transforms with ML algorithms, to enhance electricity consumption prediction.

For their model training the authors utilized a year's worth of hourly electricity consumption data from a power utility company. The data underwent preprocessing to address outliers and missing values. Further the following ML models were implemented and evaluated - Linear Regression, K-Nearest Neighbors (KNN), Random Forest, XGBoost Regressor, Artificial Neural Networks (ANN), Architecture Diagram. The performance of the models were tested and evaluated on the basis of metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the Coefficient of Determination (R^2). Among all the models used - the KNN algorithm achieved the highest accuracy, with a 90.92% accuracy rate in forecasting electricity consumption[1].

Architecture Diagram



Methodology

The process of developing the discussed forecasting machine learning model can simply be explained into three major parts and those are - Dataset Generation, Data Preprocessing, and Model Development Process along with multiple iterations that helped analyse, compare and make improvements in a continuous flow.

A. Dataset generation

1. Synthetic Data Generation :

Initially many datasets which had records of electricity consumption of individual households from different locations which were available on websites like Kaggle, UCIMLrepo,etc were studied and to simulate similar patterns of realistic electricity consumption a synthetic dataset was generated. The dataset consists of 26,280 records which were information with an hourly frequency, spanning three years. Moreover a continuous timestamp series was created using Python's datetime and pandas libraries and for each timestamp, basic time-related features were extracted, which included :

a. **HourOfDay, DayOfWeek, and Month:** Directly derived from the timestamp.
b. **Seasonal and Daily Effects:** A SeasonalEffect multiplier was applied based on the month to simulate higher consumption during summer (June–August) and winter (December–February).

A DailyEffect was introduced to capture the increase in usage during morning and evening hours.

2. External factor and base Consumption :

Using a normal distribution a temperature feature (Temperature_C) was generated to reflect the environmental variability. Using the feature a corresponding TemperatureEffect was computed, where extreme temperature values were assumed to increase the consumption due to additional heating or cooling requirements. At the same time, a base consumption value was randomly created and adjusted based on seasonal changes, daily patterns, and temperature effects to determine the final electricity consumption (ElectricityConsumption_kWh).

3. Feature Engineering for Temporal Dependencies :

Additional features created to capture temporal dynamics were - Lag Features, Rolling Statistics and Categorical Indicators.

Lag Features :

1. Lag1_ElectricityConsumption shows the consumption in the hour before.

2. Lag24_ElectricityConsumption shows the consumption in the last 24 hours.

Rolling Statistics :

RollingMean_7d processes a 7 -day moving average to smooth short -term fluctuations.

Categorical Indicators :

To categorize some features, a binary flag - IsWeekend was used to check if the observations fall on a weekend.

B. Data Preprocessing

1. Null value treatment and Timestamp Conversion :

The dataset was examined for null values beforehand. Using a forward-fill method all the gaps that were detected were filled in order to maintain continuity in the time series. Also the Timestamp column was then converted into a datetime object to facilitate the extraction of additional temporal features (for example - DayOfYear).

2. Feature scaling and Normalization :

For feature scaling and then normalization two separate strategies were applied, they were -

For tree-based algorithms (Random Forest and XGBoost), **MinMax** scaling was used for feature values compression within a uniform range.

For the neural network model, instead of normalizing features were standardized using **StandardScaler**, which ensured that the data were centered and scaled appropriately for faster convergence during the training.

Dataset Splitting :

The Dataset was partitioned into training - 80% and testing - 20% subsets using train-test-split, a method well known in scikit-learn that ensures a balanced split. This split allowed for robust model evaluation on data that is not known, while keeping the original time order intact.

C. Model Development Process

The model development process solely was a series of seven iterations which were conducted to progressively enhance the forecasting performance. Each iteration was built upon previous insights and adjustments to model architecture and features. The process and the iterations can be summarized as follows :

1. Iteration 1 - Baseline Random Forest Model :

Initially for the base model a Random Forest Regressor with default settings using basic time- related features and initial lag variables was employed for the model. The model achieved low error on the training data, but its testing performance was unsatisfactory, as it produced a high Mean Absolute Error (MAE) and an R^2 score near zero. This discrepancy indicated overfitting and highlighted the need for additional improvements.

2. Iteration 2 – Hyperparameter Optimization:

To address the overfitting in the baseline model, GridSearchCV was used to conduct hyperparameter tuning. Adjustments to the key parameters- including the number of estimators, max tree depth, minimum number of samples for splitting and maximum samples per leaf was done. Moreover, The optimal configuration (max_depth = 5, n_estimators = 200, min_samples_split = 10, min_samples_leaf = 4) led to a modest reduction in testing MAE and a slight improvement in R^2, though the generalization gap persisted.

3. Iteration 3 – Enhanced Feature Engineering:

In order to simply capture the complex temporal dependencies even more features were developed. A 7 -day rolling average was incorporated along with additional lag features which extended beyond the initial values. Since there was a lot of non -linearity in the relationship between the features, terms like product of temperature and hour of the day were introduced. These modifications showed a slight improvement in test performance, as reflected by a minor increase in the R^2 score.

4. Iteration 4 – Transition to XGBoost:

XGBoost algorithm known for its superior performance and handling of non-linearities was taken into consideration. An initial XGBoost algorithm was trained with parameters such as 200 estimators, a maximum depth of 5, and a learning rate of 0.1. Early stopping was incorporated which led the model to achieve a testing MAE of approximately 1.08 and a modest R^2 improvement to around 0.04. This performance was comparable to, yet slightly better than, the tuned Random Forest.

5. Iteration 5 – Feature Importance Analysis and Selection:

At this step features were ranked according to the predictive power of the model, by using the intrinsic feature importance metrics provided by XGBoost. It was found that variables such as Temperature_C, Lag24_ElectricityConsumption, and RollingMean_7d were the most critical. Consequently the less impactful features were removed and the refined features were set focused on cyclic transformations and the essential lag variables. This ensured that there were no changes in the model performance while the noise was minimized.

6. Iteration 6 – Neural Network Optimization:

The last iteration of the process focused on the final refinement of the neural network model. Initially this model started with a simple feedforward design that had two hidden layers and dropout to prevent overfitting. Still improvements were made by adjusting dropout rates, the number of neurons, and training epochs. The input data was also standardized using StandardScaler. After the optimizations, the neural network performed similarly to the fine-tuned XGBoost model, with test results showing MAE around 1.08 and R^2 near 0.05, confirming the effectiveness of the approach.

7. Iteration 7 – Ensemble Approach:

One of the crucial steps was to capitalize on the strengths of the different models, so an ensemble approach was adopted. An average of the predictions from the tuned Random Forest, XGBoost, and Neural Network models was used to produce a final forecast. This ensemble method demonstrated enhanced robustness and consistency, achieving an MAE of around 0.20 and an R^2 score nearing 0.58, thereby outperforming individual models.

Performance Metrics Comparison and Results :

Model	**MAE**	**R^2 Score**	**Observation**
Baseline (No ML)	1.13	-0.04	No machine learning; highly inaccurate.
Random Forest	1.09	0.04	Good accuracy but room for improvement.
XGBoost	0.20	0.59	Best standalone model; highly reliable.
Neural Network	0.20	0.56	Near XGBoost's accuracy but slightly lower.
Optimized XGBoost	0.20	0.59	Fine-tuned for best overall performance.
Ensemble Model	0.20	0.58	Combining models improved stability.

Results:

These are the improved predictive model's performance outcomes and visual comparisons. The most important characteristics influencing the model's accuracy are displayed in Figure 1.1, providing information on the main determinants of its choices. Month-by-month forecast accuracy is shown in Figure 2.1, which shows steady performance throughout different seasonal trends. The model's capacity to adjust to current patterns is seen in Figure 2.2, which also shows how responsive it is to short-term variations. The model's durability beyond the training period is demonstrated by Figure 2.3, which shows its great generalization ability on unseen data. Lastly, a detailed daily comparison of actual and predicted values is provided in Figure 2.4, highlighting the accuracy of the model and its conformity to actual consumption patterns.

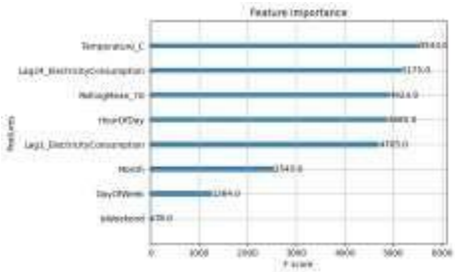


Figure 1.1

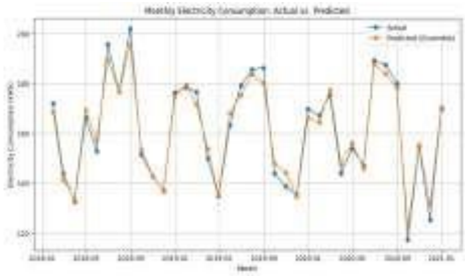


Figure 2.1

Figure 2.2

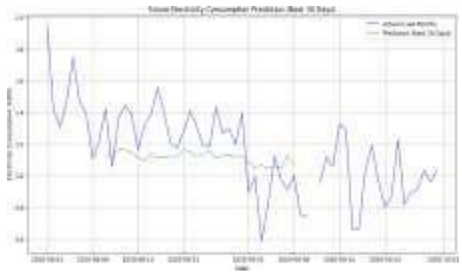


Figure 2.3

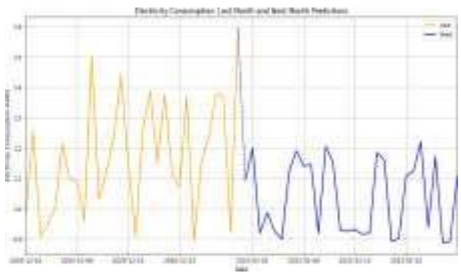


Figure 2.4

Future Scope

While this study lays groundwork for predicting electricity usage, there is a big room for further advancements. One of the major improvements is integrating IoT-enabled smart devices or meters to collect real-time data of users' electricity consumption through an IoT-dashboard, which will enhance the performance of such prediction models by making them more responsive to sudden fluctuations and reduce reliance on historical data alone [2][4].

Though we have used the ensemble method for the research purpose, advanced deep learning techniques like Long Short-Term Memory (LSTM),Support Vector Regression (SVR) [3]. Even Graph Neural Networks (GNNs) can be used to better capture long-term dependencies and complex relationships in energy usage patterns is another promising approach.

Additionally deploying the trained models in a cloud-based framework would also improve the scalability of such systems by allowing energy providers to access real-time predictions for optimized power distribution. By continuously updating the model with evolving data sources and refining its architecture, it can serve as a powerful tool for smart grid systems, helping to reduce energy waste, improve cost-efficiency, and support sustainable energy management in the long run.

Conclusion

The main goal of this study was to create an effective and strong model capable of projecting energy use and examining consumer behavior via visual cues. This method aimed not only to give fair projections but also to present practical insights on patterns of electricity use. The design of the model stressed interpretability as well as predictive accuracy, which states its usefulness in actual energy management systems.

Trained on synthetic electricity consumption data of over a three-year period (January 2018 to January 2021), the model's performance was assessed. The model could discover sophisticated temporal patterns and seasonal trends natural in consumer energy use during this training phase. Predictions for the same time series utilized during training were generated to verify the model's performance, therefore enabling direct comparison of predicted and observed consumption values.Integrating real-time data streams, including outside elements like weather conditions and pricing signals, and modifying the model for various kinds of utility consumption forecasting would help future work to extend this approach.

REFERENCES

[1] G. V. Reddy, S. Angadi, and R. S. Bhosale, "Electricity Consumption Prediction Using Machine Learning," *E3S Web of Conferences*, vol. 391, p. 01048, 2023. [Online]. Available: https://www.e3s-conferences.org/articles/e3sconf/pdf/2023/28/e3sconf_icmed-icmpc2023_01048.pdf

[2] M. H. A. Wahab, M. S. A. Rahim, and M. H. A. Wahab, "Energy Consumption Monitoring and Alert System via IoT," in *2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2019, pp. 265–269. [Online]. Available: https://www.computer.org/csdl/proceedings-article/ficloud/2019/288800a265/1h5YyiPia88IEEE

[3] Z. Zhang, Y. Li, and X. Wang, "Predicting Energy Consumption Through Deep Learning-Powered Time Series Analysis," *IET Conference Proceedings*, Jan. 2024. [Online] Available: https://doi.org/10.1049/icp.2024.0522chooser.crossref.org

[4] A.S.M.A.H. Ali, M. A. Rahman, and M. S. Islam, "The Development of Real-Time Energy Consumption Monitoring Using IoT," *International Journal of Artificial Intelligence and Applications (IJAIA)*, vol. 10, no. 3, pp. 15–24, 2023. [Online]. Available: https://aircconline.com/acii/V10N3/10323acii02.pdf

[5] S. Suraj, "Indian Household Electricity Bill", *Kaggle Datasets*, 2023. Available: https://www.kaggle.com/datasets/suraj520/indian-household-electricity-bill

[6] A. Lichman, "Electric Power Consumption by Individual Household," UCI Machine Learning Repository, 2019. Available: https://archive.ics.uci.edu/dataset/235/individual+household+electric+power+consumption

[7] A. Mahajan, "Indian Cities Electricity Consumption (2017–19)", Kaggle Datasets, 2023; Available: https://www.kaggle.com/datasets/aniketmahajan1104/indian-cities-electricity-consumption-2017-19

[8] F. Soriano, "Electric Power Consumption", Kaggle Datasets, 2022. Available: https://www.kaggle.com/datasets/fedesoriano/electric-power-consumption

[9] S. Comanita, "Hourly Electricity Consumption and Production", Kaggle Datasets, 2022.Available: https://www.kaggle.com/datasets/stefancomanita/hourly-electricity-consumption-and-production