Household Electricity Peak Forecasting

Final Report

Kushani C P De Silva

1.Introduction and Problem Statement

Electricity demand fluctuates throughout the day, with peak periods placing heavy pressure on the power grid. Meeting these peaks often requires costly backup generation, leading to higher operational costs, carbon emissions, and potential supply shortages.
Households significantly contribute to this issue by operating high-power appliances during peak hours, often without awareness of grid demand. As a result, energy use becomes inefficient and unsustainable.
Accurately predicting household electricity peaks can help shift consumption to off-peak times, reducing grid strain and promoting sustainable energy use.

Objectives

- Forecast household electricity consumption using time-series models.
- Identify peak demand periods from historical data.
- Recommend optimal times for operating energy-intensive appliances.

Problem Statement
Uncoordinated household energy usage during peak hours increases costs, emissions, and grid stress. A predictive model is needed to forecast electricity peaks and guide optimal appliance usage for efficient and sustainable energy management.

2. Dataset Description

Dataset Source: UCI Machine Learning Repository
Dataset Name: Individual Household Electric Power Consumption
Direct Link:
https://archive.ics.uci.edu/dataset/235/individual+household+electric+power+consumption

Size: ~2 million rows

This dataset contains measurements of electricity consumption from a single household over nearly four years (December 2006 to November 2010) with a

one-minute sampling rate, resulting in over 2 million records. It provides a detailed view of household energy usage, making it suitable for time-series forecasting of electricity peaks.

Features:

1. Date – Day of measurement
2. Time – Time of measurement
3. Global Active Power (kW) – Total active power consumption
4. Global Reactive Power (kW) – Reactive power consumption
5. Voltage (V) – Household voltage
6. Global Intensity (A) – Overall current
7. Sub-Metering 1 (Wh) – Kitchen appliances energy consumption
8. Sub-Metering 2 (Wh) – Laundry room appliances energy consumption
9. Sub-Metering 3 (Wh) – Water heater and air-conditioner energy consumption

Relevance:

The dataset captures both total household consumption and specific appliance usage, allowing analysis of peak demand periods and the development of strategies for optimal appliance scheduling.

Limitations and Bias:

- Data comes from a single household, limiting generalizability to other households or regions.
- Some entries contain missing values, which require preprocessing.
- The dataset does not include external factors such as weather, occupancy, or seasonal variations, which may influence electricity consumption.

3. Preprocessing & EDA

The household electricity consumption dataset was preprocessed using an integrated workflow to prepare it for machine learning tasks. The main steps include:

1. Handling Missing Values – Missing readings were replaced using a forward fill method and time-based interpolation to maintain temporal continuity.

2. Outlier Detection & Removal – Extreme values were identified using the Interquartile Range (IQR) method and capped to reduce their impact on model performance.

3. Feature Engineering – Additional predictive features were generated:
    - Lag features (previous consumption values)
    - Rolling statistics (mean, standard deviation)
    - Time-based features (hour, day of week, month)
    - Feature importance ranking using Random Forest to select relevant variables

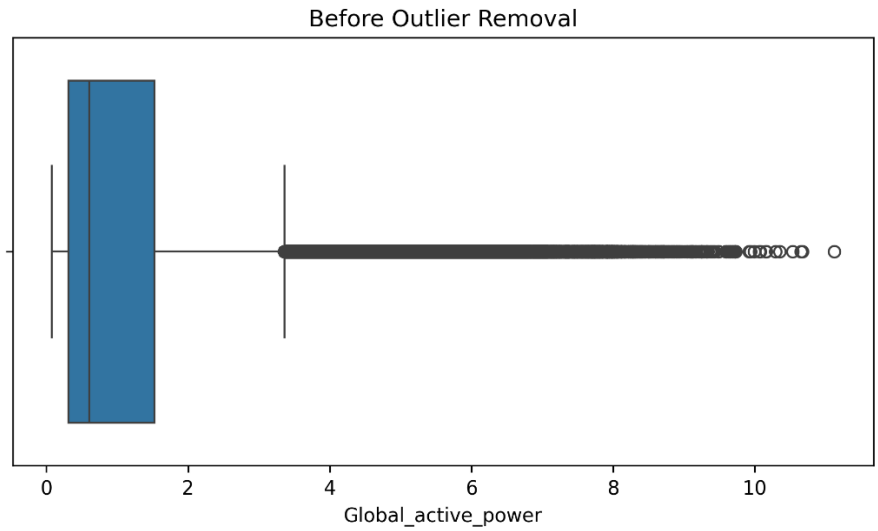
4. Dimensionality Reduction – PCA was applied to reduce redundancy and highlight principal consumption patterns.

5. Log Transformation – Skewed distributions (e.g., energy consumption) were log-transformed for normality.

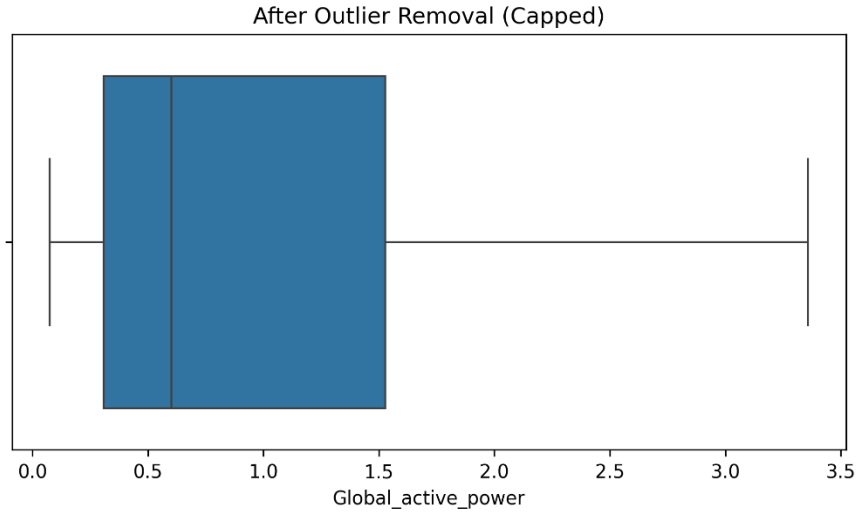
6. Scaling – All numerical features were scaled using Min–Max normalization to ensure uniform range for modeling.

The pipeline produces a clean, feature-rich dataset ready for time-series forecasting.

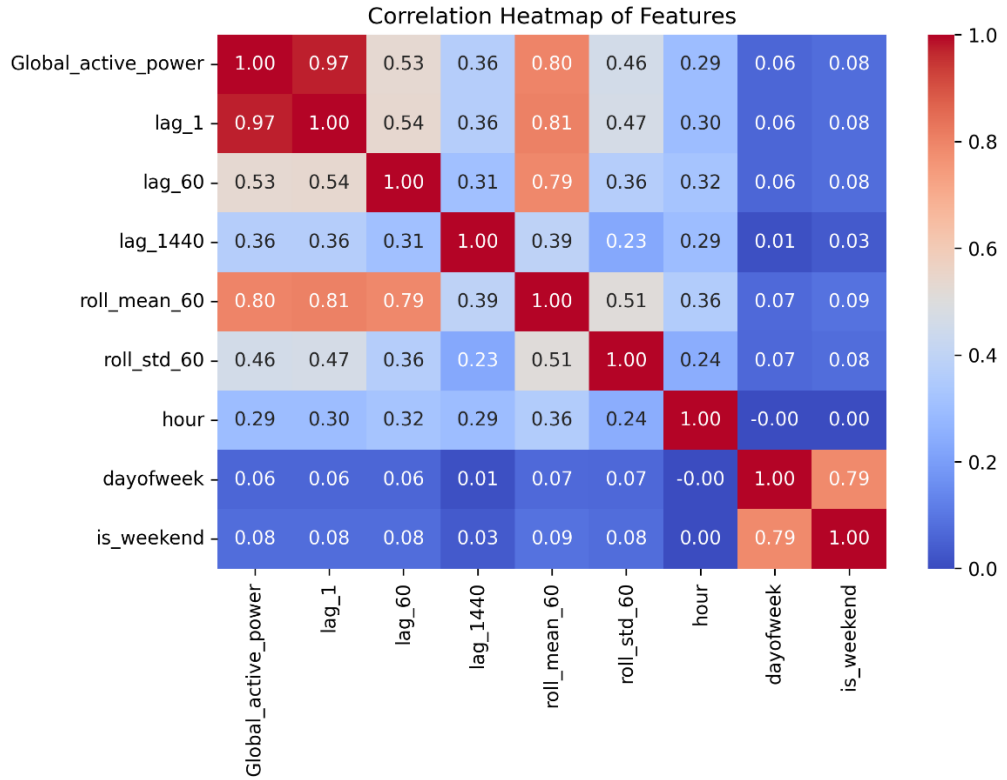
EDA Visualizations:



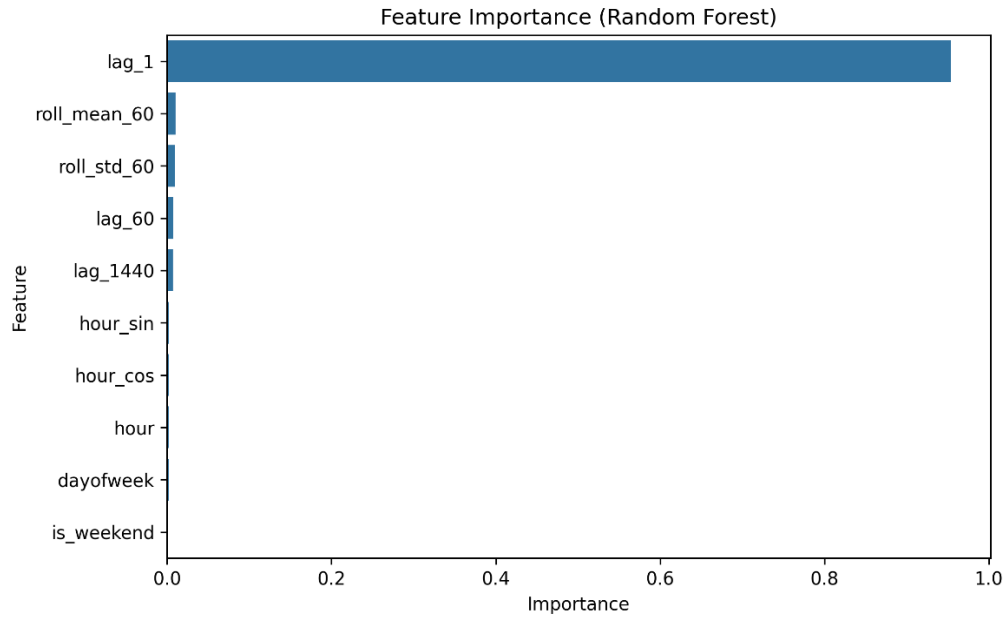
The boxplot showed a highly right-skewed distribution with many extreme outliers. Most values were between 0 and 2, while a few extended beyond 10, indicating irregular high energy readings.



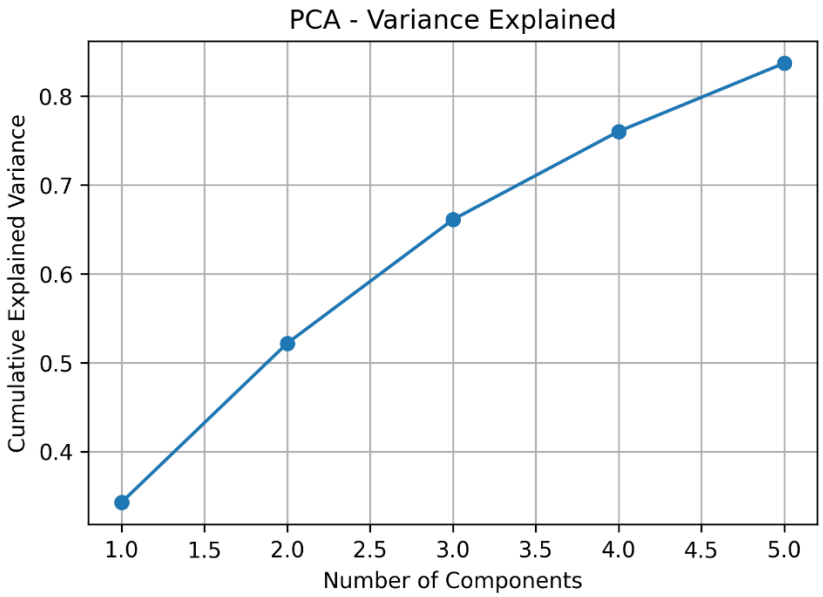
After capping, the data became more compact and balanced, with values limited to around 0–3.5. This reduced the influence of extreme values and improved data consistency.



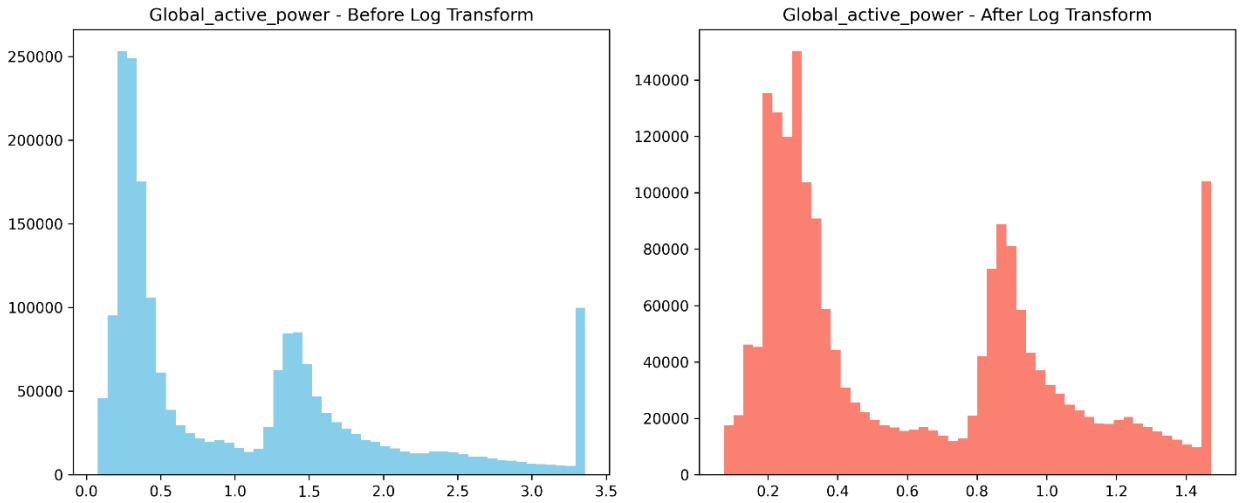
The correlation heatmap reveals strong positive relationships among time-lagged features, particularly lag_1 and roll_mean_60. This confirms the presence of temporal dependency in the dataset, validating the suitability of time-series forecasting approaches.



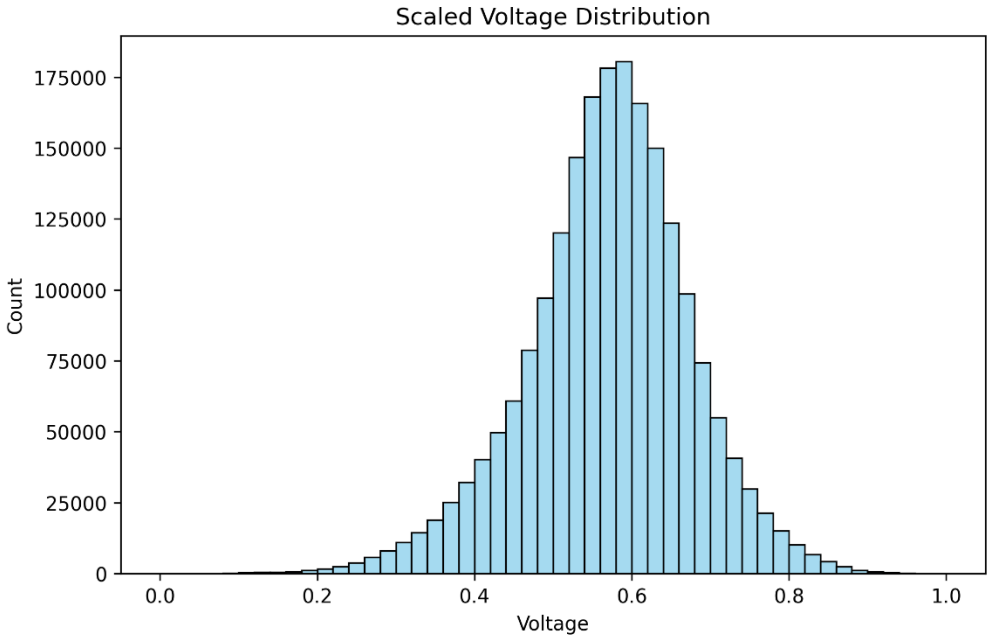
The Random Forest feature importance analysis indicates that lag_1 is the most dominant predictor, significantly outweighing other features.



The PCA variance explained plot shows that the first 3–4 principal components account for over 75% of the total variance. This suggests the presence of feature redundancy and confirms that dimensionality reduction can be applied without significant information loss.



Before transformation, the data was strongly right-skewed. After applying a log transform, the distribution became smoother and more symmetrical, making it suitable for modeling.



The voltage data was cleaned and scaled between 0 and 1. The histogram shows a smooth, bell-shaped distribution centered around 0.6, indicating consistent readings and well-prepared data suitable for further analysis.

4. Model Design and Implementation

The regression task involves predicting 'Global_active_power_log' from a time-series dataset. A diverse set of models was chosen to capture different types of relationships (linear, non-linear, temporal) and to benchmark performance, particularly focusing on their ability to handle the sequential nature of the data.

1).Support Vector Regression (SVR)

SVR is a robust non-linear model that maps data to a higher-dimensional space for linear regression, using an epsilon margin to ignore small errors, making it effective for noisy data like power consumption.

Variation	Architectural Choice / Justification	Key Parameters Explained
Variation 1: Basic SVR	Baseline performance using the standard RBF kernel before any tuning.	kernel= ‘rbf’ : Standard non-linear mapping. C=1.0 : Default regularization. ε =0.1: Default error tolerance (epsilon-tube).

Variation 2: Tuned SVR	Hyperparameter Optimization using RandomizedSearchCV and TimeSeriesSplit to efficiently find optimal settings while respecting temporal order.	C: Controls error penalty (regularization strength). ε: Defines the acceptable error margin. gamma: Governs kernel influence/model complexity.
Variation 3: Reduced Features + Poly Kernel	Evaluates a Polynomial kernel to model specific polynomial trends and tests performance with a reduced feature set (top 8 correlated features).	kernel='poly', degree=2: Implements a quadratic mapping. C=10, ε =0.05: Manually adjusted for a stronger fit on the simplified input space.

2). Decision Tree Regressor (DTR)

DTR is a simple, non-parametric model that predicts by recursively partitioning the feature space, offering interpretability and capturing complex non-linear interactions without needing feature scaling.

Variation	Architectural Choice / Justification	Key Parameters Explained
Variation 1: Basic DTR	High-variance baseline; minimal constraints (max depth allowed).	Default: max_depth=None (prone to overfitting).
Variation 2: Tuned DTR	RandomizedSearchCV+ TSCV to find optimal pruning constraints.	Ranges: max_depth, min_samples_split/leaf (complexity limits).
Variation 3: Regularized DTR	Manually constrained (max_depth=10) for a generalized, robust tree structure.reduced feature set (top 8 correlated features).	max_depth=10, min_samples_split=10, min_samples_leaf=5

3). K-Nearest Neighbors Regressor (KNN)

KNN is a non-parametric, instance-based model that predicts values using the average of the K nearest neighbors, relying on local structure—especially effective for time-series data—with feature scaling being essential.

Variation	Architectural Choice / Justification	Key Parameters Explained
Variation 1: Basic KNN	Baseline for instance-based local structure; standard neighborhood size.	n_neighbors=5, weights='uniform', p=2 (Euclidean).
Variation 2: Tuned KNN	RandomizedSearchCV+ TSCV optimize neighborhood size (K) and distance metric.	Ranges: n_neighbors, weights(uniform/distance), p(L1/L2 norm).
Variation 3: Regularized KNN	Manual configuration (K=9) prioritizing a distance-weighted, stable local average.	n_neighbors=9, weights='distance'.

4). Random Forest Regressor (RFR)

RFR is an ensemble of decision trees, offering strong non-linear modeling and reduced overfitting risk, making it a reliable out-of-the-box regression method.

Variation	Architectural Choice / Justification	Key Parameters Explained
Variation 1: Basic RFR	Baseline for ensemble method; low variance and high non-linear capacity.	n_estimators=100 (default ensemble size).
Variation 2: Tuned RFR	RandomizedSearchCV+ TSCV to balance ensemble size and individual tree complexity.	Ranges: n_estimators,max_depth, max_features.
Variation 3: Regularized RFR	Manual configuration (n_estimators=150,max_depth=15) for a balanced, generalized model.	n_estimators=15, max_depth=15

5). Long Short-Term Memory (LSTM)

LSTM, a type of RNN, models sequential data and long-range dependencies, making it well-suited for time-series forecasting; input data is reshaped into 3D format (samples, timesteps, features) for compatibility.

Variation	Architectural Choice / Justification	Key Parameters Explained
Variation 1: Simple LSTM	Feasibility Baseline: Shallow network to confirm the suitability of the architecture.	Single LSTM (16) unit layer. Adam optimizer, mse loss.
Variation 2: Tuned RFR	Wider Network with regularization to increase learning capacity and reduce overfitting.	Two LSTM layers (100, 50 units). Dropout(0.2) for regularization. Learning_Rate=0.001.
Variation 3: Stacked LSTM	Deeper Network to extract hierarchical temporal features.	Three Stacked LSTMs (128, 64, 32). return_sequences=True. Dropout(0.3, 0.2)

5. Evaluation and Comparison

Model evaluation was conducted using standard regression metrics — Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Coefficient of Determination (R^2) — to assess prediction accuracy and generalization performance. Training time was also analyzed to understand computational efficiency. The results for each team member's optimal model are summarized in Table 1.

Member	Model	RMSE (Test)	MAE (Test)	R^2 (Test)	Training Time(s)
1	LSTM	0.014818	0.010035	0.996950	27.21
2	SVR	0.012696	0.008211	0.997745	38.87
3	Random Forest	0.010209	0.005457	0.998542	30.41
4	KNN	0.051599	0.032965	0.962756	0.16

5	Decision Tree	0.012301	0.006804	0.997883	0.53

Performance Comparison

Most models achieved strong results, with R^2 scores above 0.99, except for KNN. The Random Forest Regressor (RFR) delivered the best overall performance, achieving the lowest RMSE (0.0102) and MAE (0.0055), indicating highly accurate and stable forecasts. SVR and Decision Tree models performed well with slightly higher errors. LSTM captured temporal dependencies effectively but required longer training (27.2 s). KNN trained fastest but yielded the least accurate predictions due to its simplicity and sensitivity to data variations.

Insights and Trade-offs

The results highlight a trade-off between accuracy and computational efficiency. Advanced models like RFR and LSTM offer high precision at the cost of longer processing times, whereas simpler models provide speed with reduced accuracy. Overall, RFR achieves the best balance, making it the most suitable choice for household electricity forecasting.

6. Ethical Considerations and Bias Mitigation

This project leverages AI and ML to forecast household electricity consumption and recommend shifting usage to off-peak hours, promoting energy efficiency and sustainability. However, it introduces ethical responsibilities around fairness, transparency, privacy, and accountability to ensure equitable benefits and system integrity.

Understanding Bias in Energy Forecasting

Bias arises when models produce systematically skewed outcomes due to data collection, representation, or algorithmic design. In electricity forecasting, potential biases include:

- Data bias: Overrepresentation of specific regions, household types, or income levels.

- Sampling bias: Disproportionate focus on certain time periods (e.g., weekdays, peak seasons).
- Algorithmic bias: Optimization favoring high-consumption households, reducing accuracy for low-usage households.

Unchecked, these biases can lead to unequal forecast accuracy and unfair recommendations for underrepresented groups.

Ethical Design Principles

To ensure responsible operation, the project incorporates:

- Fairness: Models are evaluated across household types and usage patterns using balanced sampling and fairness-aware validation.
- Transparency & Explainability: Feature importance analysis and SHAP/LIME methods clarify how inputs influence predictions, building trust.
- Accountability: All development stages—including preprocessing, feature selection, and parameter tuning—are documented for reproducibility.
- Privacy & Data Protection: Household data is anonymized, securely stored, and processed only for analysis; personally identifiable information is excluded.
- Human Oversight: Recommendations are decision-support tools; users retain full control.
- Beneficence & Non-Maleficence: The system maximizes environmental and financial benefits while minimizing potential harm.

Bias Mitigation Strategies

Bias is mitigated through a combination of technical and governance measures:

- Balanced and Representative Data: Sampling ensures diverse consumption patterns across time, seasons, and households.
- Fairness-Aware Modeling: Multiple regression models (SVR, Decision Tree, Random Forest, KNN, LSTM) are compared to identify consistent, unbiased performers.

- Explainable AI (XAI): Feature interpretation detects and corrects variables with disproportionate influence.
- Continuous Monitoring: Model performance is periodically reviewed to detect drift or emerging bias.
- Ethical Governance: Principles of fairness, transparency, accountability, and safety guide the project lifecycle.

Societal and Environmental Responsibility

Beyond fairness, the project promotes social and environmental benefits. Encouraging off-peak electricity usage improves energy efficiency, supports grid stability, and reduces carbon emissions. Recommendations are voluntary, personalized, and inclusive, ensuring all households can benefit equitably from sustainable energy insights.

7. Reflections and Lessons Learned

This project strengthened our understanding of applying machine learning to real-world energy forecasting. We learned that achieving accurate results requires not only model tuning but also careful data handling and fairness awareness. Managing large time-series data and balancing model accuracy with computation were key challenges. Comparing models like LSTM, SVR, and Random Forest helped us appreciate their different strengths and limitations. Team collaboration was essential for troubleshooting and integrating results effectively. Overall, this project strengthened our technical, analytical, and collaborative skills, deepening our understanding of responsible AI development in sustainable energy applications.

8. References

[1] Bias and Ethics in Artificial Intelligence (AI) and Machine Learning, Lecture notes, Dept. of Computing, 2024.

[2] OECD, OECD Principles on Artificial Intelligence, Organisation for Economic Co-operation and Development, 2019.

[3] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA: MIT Press, 2016.

[4] F. Chollet, Deep Learning with Python. Shelter Island, NY: Manning Publications, 2017.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, ... and É. Duchesnay, "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, no. 85, pp. 2825–2830, 2011.

[6] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.