Advances in Artificial Intelligence and Machine Learning; Research (Ahead of Print) Received 15-12-2025; Accepted 03-03-2026; Published online 10-03-2026

Joint Forecasting of Residential Energy Consumption and Solar Generation Using Advanced AI Architectures

Milia Habib milia.habib@liu.edu.lb
Department of Computer & Communications Engineering
Lebanese International University
Beirut, Lebanon

Majd Al Ayoubi majdayoubi4@gmail.com
Department of Computer & Communications Engineering
Lebanese International University
Beirut, Lebanon

Mohamad Kanaan mohamad.kanaan01@liu.edu.lb
Department of Electrical Engineering
Lebanese International University
Beirut, Lebanon

Zaher Merhi zaher.merhi@liu.edu.lb
Department of Computer & Communications Engineering
Lebanese International University
Beirut, Lebanon

Corresponding Author: Milia Habib

Abstract

This paper presents a machine learning–based Smart Home Energy Management System designed to predict both household electricity consumption and rooftop solar power generation. It incorporates a structured pipeline for data preprocessing and feature extraction, enriched with contextual variables such as weather conditions and calendar information. sequence-to-sequence learning using convolutional neural networks (CNNs) and long short-term memory (LSTM) architectures is implemented. For comparative evaluation, a robust tree-based model serves as a baseline. Model performance is evaluated using standard metrics, including MAE, RMSE, and R^2 along with peak oriented metrics to assess ramp rate and peak fidelity. Additionally, a lightweight web front-end based is developed to provide real-time inference and interactive visualization for decision support. The results show that LSTM achieved the highest accuracy in global metrics and was therefore adopted as the system's forecasting backbone.

Keywords: Smart home, Energy forecasting, Solar generation, Machine learning, LSTM, CNN.

Citation: Milia Habib, et al. Joint Forecasting of Residential Energy Consumption and Solar Generation Using Advanced AI Architectures. **Advances in Artificial Intelligence and Machine Learning**. 2026. (Ahead of Print). https://dx.doi.org/10.54364/AAIML.2026.62286

1. INTRODUCTION

Smart homes are no longer convenient, but are becoming intelligent systems that integrate sensing, Internet of Things (IoT) devices, and distributed energy resources (DER) to balance comfort, cost, and sustainability at the same time. Residential electricity use accounts for most of the total use, yet it fluctuates considerably due to human activity, climate conditions, and appliance cycles [1, 2]. This variability creates uncertainty for households in planning their electricity bills and for operators in maintaining grid stability. Therefore, accurate short- to medium-term forecasts of both household consumption and rooftop solar generation are essential to enable demand response, support dynamic pricing, optimize storage scheduling, and automate smart home energy management in a carbon-aware manner [3, 4]. Even with recent progress, many energy forecasting methods still have clear limitations. Most prior studies focus only on past energy consumption, assuming that the usage of yesterday is the best predictor of tomorrow. Although this can capture basic trends, it ignores critical external factors such as temperature, humidity, solar irradiation, or even day of the week, all of which strongly influence household demand. As a result, such models often fail when conditions change. Moreover, residential data are noisy, irregular, and multi-scale, making robust preprocessing and feature construction essential [5]. Another deficit is that models calibrated for smooth averages typically fail to capture sharp peaks and ramps, just the occurrences that cause higher expenses and strain on the grid. Lastly, most previous work studies consumption and PV generation separately; however, actual residential energy management systems require considering both collectively to schedule loads and balance storage efficiently [6].

In this study, a broader spectrum is considered by incorporating multiple sources of information beyond historical consumption. Specifically, it integrates weather data, calendar-based features, and engineered variables designed to capture consumption peaks and underlying trends. With a fuller view, the model learns not only how much was used but also why usage shifts with changing conditions. The main contributions of this paper are:

1. *Data preparation pipeline with advanced feature engineering*: a unified preprocessing framework that cleans and aligns data, enriches it with weather and calendar information, scales features, and constructs supervised sequences suitable for residential forecasting.
2. *Dual forecasting tasks:* simultaneous prediction of household consumption and solar PV generation, allowing a unified view of net demand.
3. *Modeling suite and comparison:* evaluation of deep sequence models (LSTM and CNN) alongside a tree-based baseline (gradient-boosted trees), under consistent training and validation protocols [7–9].
4. *Peak-aware evaluation:* assessment that goes beyond global error (MAE, RMSE, R^2) by highlighting performance in peaks, with observations in sequence lengths, forecast horizons, and feature sets.
5. *Operational dashboard:* a lightweight web interface that delivers real time predictions and visualizations to support household decisions and potential demand response programs.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 describes the data and preprocessing. Section 4 presents the models. Section 5 explains the experiments and evaluation. Section 6 presents the results. Section 7 concludes the paper and discusses future work.

2. RELATED WORK

Significant research has focused on forecasting energy consumption, by using various machine learning and deep learning algorithms to improve prediction accuracy.

2.1 Traditional Machine Learning Approaches

Traditional machine learning methods have been used for building load prediction. The authors in [10], used regression models like linear and polynomial regression for short-term energy forecasting, which demonstrated their usability for basic demand patterns. Bourhnane et al. (2020) [11], used Artificial Neural Networks (ANN) in smart building scheduling, demonstrating the potential of neural methods for capturing complex consumption behaviors. Elhabyb et al. (2024) [12], compared Random Forest (RF), Gradient Boosting Regressor (GBR), and Long Short-Term Memory (LSTM) on educational buildings, and reported that ensemble and sequence models both provide strong results.

Paudel et al. (2015) [13], applied Support Vector Regression (SVR) with relevant-day selection to real-world load prediction data. Shapi et al. (2021) [14], presented a Microsoft Azure–based framework combining SVR, ANN, and k-Nearest Neighbors (KNN) for industrial energy demand prediction. Olu-Ajayi et al. (2022) [15], also studied annual residential energy consumption using several supervised learning models, providing observations on residential consumption patterns. Boopathi et al. (2021) [16], used Grey Models to forecast electricity and water generation in Bahrain, showing that grey-box methods can be effective for long-term utility planning. Sivakoti and Mozumdar (2015) [17], proposed a hybrid load forecasting approach combining artificial neural networks, linear regression, and curve-fitting methods to improve short-term prediction accuracy in smart grid networks.

2.2 Deep Learning and Hybrid Approaches

Deep learning methods have recently been used to better capture time-based patterns in energy consumption data. Abumohsen (2023) [18], constructed LSTM-based models for energy forecasting and evaluated the impact of operational patterns on prediction performance. Ndife et al. (2022) [19], introduced a ConvLSTM encoder–decoder model to predict power consumption in low-power devices, demonstrating the effectiveness of convolutional recurrent networks for sequence data. Kong et al. (2019) [20], developed an LSTM-based method for short-term residential load forecasting, showing significant improvements over traditional time-series approaches. Shi et al. (2018) [21], proposed a pooling deep recurrent neural network tailored for household load forecasting, improving temporal modeling under varying load behaviors. Kim and Cho (2019) [22], introduced a CNN–LSTM hybrid architecture that extracts spatial–temporal consumption patterns to improve residential energy prediction. Kim and Cho (2019) [23], further extended their work by proposing an explainable autoencoder-based model to provide interpretable deep-learning predictions for residential energy consumption.

Neshat et al. (2025) [24], proposed an adaptive evolutionary ensemble framework that combines different learners to improve forecast accuracy in smart building applications. Liu et al. (2025) [25], presented BuildSTG, a spatio-temporal graph neural network that leverages correlations across multiple buildings for improved load forecasting. Liang et al. (2025) [26], proposed the CCNN-QL model for load prediction in urban areas. The model combines convolutional neural networks with reinforcement learning. Duong and Nam (2022) [27], developed a machine learning system to monitor energy use at the appliance level. Alghamdi (2024) [28], studied stacked LSTM models for energy consumption prediction when data are limited.

Many energy prediction studies mainly use past load values. In many cases, energy consumption and renewable generation are treated separately. This does not reflect how energy is used in real homes. In this work, weather data, solar irradiance, and time-related information are also included. Household consumption and solar generation are predicted at the same time, so demand and on-site production can be examined together in a smart home setting.

2.3 Advantages of the Proposed Approach Over Existing Studies

Energy forecasting has been studied for many years. Still, many models do not work well in real homes. They often rely on limited data, such as past energy use or basic weather information. Household energy use is irregular and can change quickly. Because of this, many models fail to reflect what happens in practice. Some approaches focus only on electricity consumption. Others focus only on solar generation. Treating these separately limits their use in real applications.

In this work, more data are used, including weather conditions, solar irradiance, time-related features, and past energy values. This helps the model follow daily changes in energy use. Consumption and solar generation are predicted together, which gives a more realistic view of energy behavior in homes with solar panels. Using different types of data also helps the model deal with sudden changes and irregular energy use. The model keeps working when energy use changes because of people at home, the weather, or the season. The approach also includes a simple dashboard that shows energy use, solar production, and battery level. This helps users see what is going on and make practical decisions.

3. DATA AND PREPROCESSING

Several datasets are used in this work. The first dataset contains household electricity consumption data. It has 26,303 hourly records from one home. Each record includes a time value and the electricity used during that hour. The second dataset contains weather data for the same time period. It includes temperature(T), dew point(TD), solar irradiance(Q), humidity(HH), and basic calendar values. Weather affects electricity use, especially heating, cooling, and lighting. The datasets were joined using the timestamp. This allows electricity use and weather data to be analyzed together.

A third dataset was used to describe solar power generation from a solar energy system equipped with multiple inverters and dedicated weather sensors. This dataset provides inverter-level AC power readings, while a second dataset contains the corresponding weather sensor measurements, including ambient temperature, module temperature, and solar irradiation. As with the consumption dataset, these two sources were not originally stored together. The AC power readings and the environmental conditions were therefore merged using their shared timestamp so that each power value is directly linked to the exact weather conditions under which it was produced. Since all measurements follow the same hourly time structure, the combined dataset was accurately aligned after removing redundant identifiers and rows with missing values. The final dataset combines electricity data and weather data. It includes consumption values and environmental information related to solar production. The dataset can be used for multi-step forecasting. Before modeling, the data was explored. The consumption data shows daily and weekly patterns. Electricity use increases in the morning and again in the evening. Some sudden spikes appear. These are often linked to high-power appliances or heating and cooling systems reacting to temperature. The solar data follows daylight conditions. Production starts after sunrise. It reaches a peak around midday on clear days. It drops to zero at night. On cloudy days or during seasonal changes, the curves become irregular. These patterns are visible in FIGURE 1.

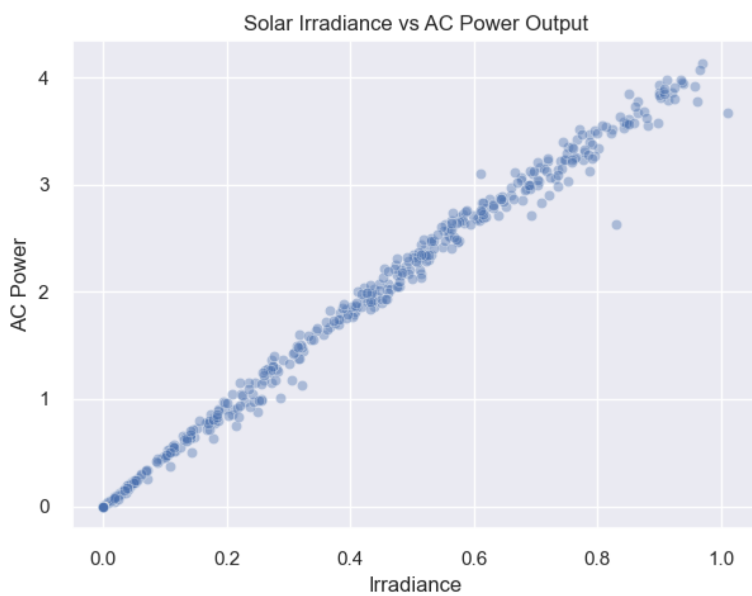

Figure 1: Effect of solar irradiance on AC power generation.

The energy, weather, and solar datasets were all resampled to an hourly frequency, sorted chronologically, and aligned by timestamp to keep consistency. This produced a unified, gapfree time series that served as the basis for subsequent preprocessing, feature engineering, and the sequence-to-sequence forecasting model described later in this work.

3.1 Data Cleaning and Formatting

The raw data were preprocessed to ensure both quality and temporal consistency. The timestamps were standardized, chronologically order and set as indices to facilitate time-series computations. Data points with missing values for the major attributes, such as energy, temperature, irradiance, or humidity were removed. The inverter level measurements for the solar generation data were aggregated and resampled to an hourly level, thereby resolving issues related to irregular sampling rates.

3.2 Feature Engineering

Several temporal and statistical features were utilized to capture environmental drivers as well as consumption dynamics. Temporal indicators were the day-of-week, and a binary indicator of for the weekend, in addition to a series of rescaled hour-of-day indices representing daily periodicity and weekly cycles. Autoregressive effects were accounted for by including lagged consumption values, hourly differences, and rolling statistics such as a 24 h moving average. A maximum flag was included to indicate consumption levels greater than the rolling average plus one standard deviation. For solar panel forecasting, extra features involved standard hour normalized one hot encoded time of day and weekday, modeling both the daily and weekly patterns in generation.

3.3 Feature Selection

Correlation analysis was also performed to find variables highly predictive, as seen in FIGURE 2. Temperature, dew point, and irradiance were all correlated with electricity consumption and solar generation. Temporal characteristics like months and hours satisfactorily caught the seasonality and daily pattern of the data, while autoregressive features improved the temporal closeness between target values. To overcome possible feature redundancy, Principal Component Analysis (PCA) was performed and the original eleven descriptive features were transformed into five principal components explaining 96.6% of the total variations. This reduction created a dense independent space that accounted for almost all of the essential information required to make intrusive predictions.

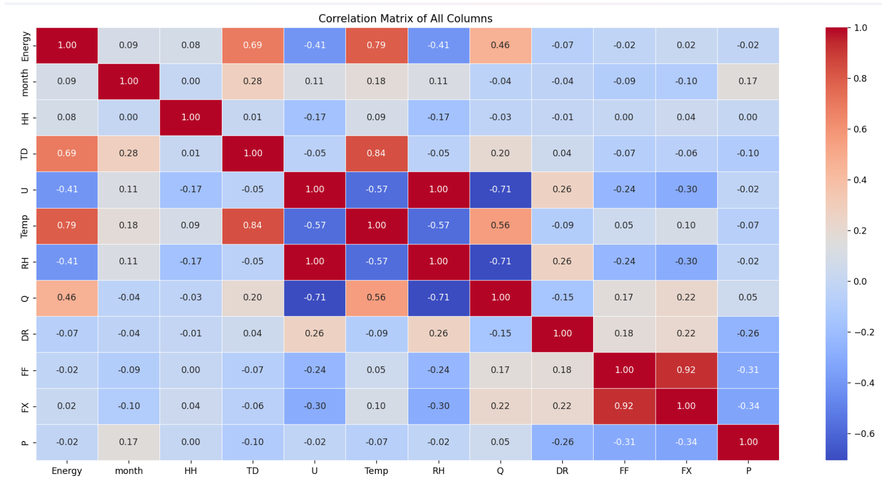

Figure 2: Correlation Matrix between the main consumption, weather, and engineered features.

3.4 Normalization and Scaling

All features were Min-Max scaled to $[0, 1]$ range of values in order to eliminate potential bias due to magnitude differences and ensure proper behavior of the neural network training. The energy consumption and PV generation outputs were scaled back separately, so that the model's output can be inverted back to original kilowatt-hour (kWh) for evaluation.

3.5 Sequence Construction

The forecasting task was defined as an n-step sequence-to-sequence problem. To give you a better idea of how the sequence-to-sequence problem was structured: a sliding window approach was used to create the inputs. Each input sequence was made up of 336 hourly data points two weeks worth of historical information. The model was then tasked with predicting the next 168 hours of data a full week ahead of the last point in the input sequence.

A sequence-to-sequence problem was chosen instead of using a standard LSTM which can only make predictions based on the previous time-step to work-around the typical one-time-step limitation found in most LSTMs. Since the model will now be able to look at the entire input sequence to make its prediction of the entire output sequence, it allows the model to better capture multi-step relationships between data points.

Another approach to solving this type of problem is known as recursive forecasting, where the model makes a prediction for the first time-step, uses that prediction as the input to the second time-step, and continues to do so until the desired number of time-steps have been predicted. Recursive forecasting does allow the model to predict further out than a single time-step but also has some major drawbacks. For example, if the model makes an error in its prediction of the first few time-steps, that error will continue to propagate through the rest of the predictions, causing the accuracy of the final predictions to decrease rapidly. By contrast, the sequence-to-sequence approach does not suffer from this same issue because the model never leaves the historical data during the prediction phase; instead the model generates the entire future sequence of values in one pass. This way, the model has a better understanding of both rapid changes in the data and long term trends in the data.

3.6 Train/Test Partitioning

The last dataset was split into training and test sets considering temporal order, to prevent information leaking over time. In case of energy consumption, 90% of the data was used for training and the other 10% for testing. An 80/20 division for solar panel generation, following the greater size of its dataset. This splitting method allowed us to cross validate the models on time segments not seen, hence giving an accurate approximation of out-of-sample predictive performance.

4. METHODS

Three prediction methodologies were studied, representing three categories of learning algorithms: recurrent neural networks, convolutional neural networks and ensemble based on gradient boosting. This comparative framework enabled systematic evaluation of sequence-oriented models alongside tree-based approaches under identical data conditions.

4.1 Long Short-Term Memory (LSTM)

Recurrent Neural Networks (RNNs) provide a natural sequence modeling framework due to their recurrent connections that allow the passage of information over time. Nevertheless, standard RNNs usually fail in long-term dependencies due to the vanishing and exploding gradient problem. Long-Short-Term Memory (LSTM) networks were introduced in an effort to overcome this weakness through the application of gating units that regulate the passage of information from one time step to the next. An LSTM cell consists of three gates—in, forget, and out—that adaptively tune how much past information is retained, how much new information is added, and how much of the internal memory states are passed out to the next layer. The gate structure lets the network learn short-term fluctuations (e.g., the daily peaks in energy) as well as longer-term seasonal patterns (e.g., daily or weekly patterns) and is therefore well-suited for applications in energy forecasting [7]. For home consumption and solar generation prediction, a bidirectional LSTM was used to learn dependencies in both forward and reverse directions of the input window. This architecture utilizes contextual information from future and past time steps at the same time, enhancing the modeling of seasonal cycles as weekday–weekend changes and daily consumption patterns.

To improve generalization, dropout was applied after recurrent layers at rate of 0.2. The dropout sets some percentage of the hidden units at zero during training in a random manner, effectively sampling from an ensemble of sub-networks. This prevents the model from co-adapting too strongly to specific neurons and helps mitigate overfitting, which is particularly desirable in sequential forecasting where strong temporal autocorrelation could otherwise lead to memorization rather than abstraction.

For both consumption and solar generation,the architecture was implemented as a sequential network combining bidirectional and unidirectional LSTM layers. The structure consisted of an initial bidirectional LSTM layer with 128 hidden units and enabled return sequences, followed by a ReLU activation function and a dropout layer with a rate of 0.2. This was followed by a unidirectional LSTM layer with 64 units, also paired with the ReLU activation function and dropout layer. The final dense layer projected the learned temporal representation into a 168-dimensional output vector, corresponding to the forecast horizon of one week. Training process used Adam optimizer with gradient clipping to ensure stable updates. Early stopping was applied based on validation loss, with patience of ten epochs, ensuring that the final model represented the optimal balance between fit and generalization. Each model was trained for up to 100 epochs with a batch size of 32, allowing sufficient iterations for convergence while avoiding overfitting.

4.2 Convolutional Neural Networks (CNN)

CNNs are widely used in the analysis of structured data, particularly in situations where local patterns and hierarchical extraction of the features are critical. In the context of energy forecasting, CNNs provide the ability to capture short-term temporal dependencies and abrupt fluctuations that may not be efficiently modeled by purely recurrent structures. The convolution layers are essentially sliding filters that learn local patterns in energy consumption, such as short peaks, while the pooling layers reduce dimensionality and enhance robustness to noise.

The CNN model employed a one-dimensional structure tailored to sequential input. Two convolutional layers with 64 filters each and a kernel size of 3 were used. Each convolutional layer was followed by max-pooling layers to progressively downsample the temporal dimension. This design allowed the network to learn increasingly abstract temporal features while retaining essential consumption and irradiance dynamics. After the convolution–pooling stack, the representation was flattened and passed through a fully connected dense layer of 128 units with ReLU activation. Dropout regularization was applied before the output layer, which produced a 168-dimensional forecast vector representing the prediction horizon of one week.

CNN was trained and optimized by Adam Optimizer with learning rate 0.001 and mean squared error (MSE) loss function. To avoid overfitting the early stopping with a patience of ten epochs was applied. All models were trained up to 100 epochs with a batch size of 32 as in LSTM configuration, in order to guarantee the comparability among architectures.

4.3 EXtreme Gradient Boosting (XGBoost)

Gradient Boosting Models are ensemble methods where weak learners are trained sequentially in order to minimize prediction errors. In the process of doing so, the algorithm will train a new tree at each step as an approximation of the residuals of the previous ensemble in order to refine the predictive model. XGBoost is an extension of this framework that has additional features including regularization, shrinkage and sparse optimization. Due to these features, XGBoost is one of the most popular gradient boosting implementations available today. XGBoost was used in the one shot, multi-step prediction configuration for the prediction task. Each input sequence of 336 hourly observations were flattened into a fixed length feature vector which was then mapped to a 168 dimensional output representing the forecast horizon of one week. Unlike recursive feedback loop strategies used by autoregressive techniques, this approach prevented the propagation of cumulative error and improved stability over large intervals of time for extended prediction horizons. Both ℓ_1 and ℓ_2 penalties were incorporated into the training objective in order to constrain the depth of the trees and weights of leaves. By doing so, the risk of overfitting was reduced. The contribution of each successive tree was scaled by a shrinkage parameter (learning rate), ensuring gradual optimization and maintaining stability. Finally, the algorithm's ability to be parallelized further accelerated the construction of the trees in the model while preserving predictive performance making it highly efficient for large scale datasets for forecasting.

5. EXPERIMENTS AND EVALUATION

5.1 Evaluation Metrics

This subsection presents a comprehensive evaluation of the proposed approach using four performance metrics. The mean absolute error (MAE) evaluates the average magnitude of deviations between predictions and ground truth (Equation 1).

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{1}$$

where y_i represents the actual observation, $\hat{y}_i$ the predicted value, and n the samples number.

The second metric is the root mean squared error (RMSE) that penalizes larger gaps more strongly by squaring the residuals before averaging (Equation 2):

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2}$$

where higher deviations $(y_i - \hat{y}_i)$ contribute disproportionately to the error measurement.

The third metric is the mean absolute percentage error (MAPE) that defines the predictive error where each sample is normalized by the true value (Equation 3):

$$\mathrm{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{3}$$

Finally, the coefficient of determination (R^2) measures the power by comparing residual variance with total variance (Equation 4):

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4}$$

where $\bar{y}$ defines the mean of the observed values. A larger R^2 shows that the model presents a larger variance in the data.

5.2 Visualization and Comparative Analysis

Visual inspections of forecasted time series were carried out using both visual graphics and summary statistics against the actual time series values. The time series graphics (plots) were created using the `Matplotlib` Python package and styled with `Seaborn` to improve readability. These plots were used to assess whether the modeled outputs reflected the correct timing of the daily patterns as well as to assess how well the models captured sudden spikes in demand and/or generation.

A comparative evaluation of the performance of each family of models (LSTM, CNN, XGBoost) was performed under the same conditions regarding sequence-to-sequence processing so that the performance of each model would only be due to the different capacities of each type of model and not to differences in the way the data was prepared. This framework enabled the assessment of recurrent architectures for their ability to model long-range dependencies, convolutional networks for their capacity to detect local temporal fluctuations, and boosted tree ensembles for their effectiveness in exploiting non-linear feature interactions.

6. RESULTS AND DISCUSSION

6.1 Consumption Model Comparison

TABLE 1 summarizes the performance of the three candidate models for forecasting household energy consumption over a 168-hour horizon. Evaluation is based on Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), where Accuracy is defined as $(1 - \text{MAPE}) \times 100$, and the coefficient of determination (R^2).

Model	MAE (kWh)	RMSE (kWh)	MAPE (kwh)	Accuracy (%)	R^2
LSTM	0.0316	0.0428	0.0365	96.35	0.959
CNN	0.0380	0.0512	0.0436	95.64	0.9409
XGBoost	0.0404	0.0582	0.0615	93.85	0.9237

Table 1: Comparison of prediction models on household energy consumption (168-hour horizon).

6.2 Consumption Forecasts: Model-by-Model Evidence

FIGURE 3-5, present 168-hour overlays of predicted (dashed) versus observed (solid) consumption for a representative test week. The three plots highlight distinct error modes across the models. The LSTM (TABLE 4) aligns most closely with the diurnal cycle, reproducing the timing of the morning ramps and the evening peaks with minimal phase error; residuals are primarily due to amplitude attenuation at the highest peaks. CNN (FIGURE 3) responds well to short-lived fluctuations and local oscillations, but exhibited systematic peak underestimation and slightly broadened maxima during high-load periods. XGBoost (FIGURE 5) matches the overall level across the week but trend to smooth abrupt transitions, producing lag on sharp up-ramps and attenuating peak magnitudes.// Based on the overall evaluation across the entire test set, the LSTM achieved the best overall performance achieving the lowest MAE, RMSE, MAPE, and the highest R^2 among the candidates (TABLE 1). Accordingly, the LSTM architecture was selected as the forecasting backbone and was subsequently employed for the solar generation experiment. The related results are illustrated in FIGURE 6.

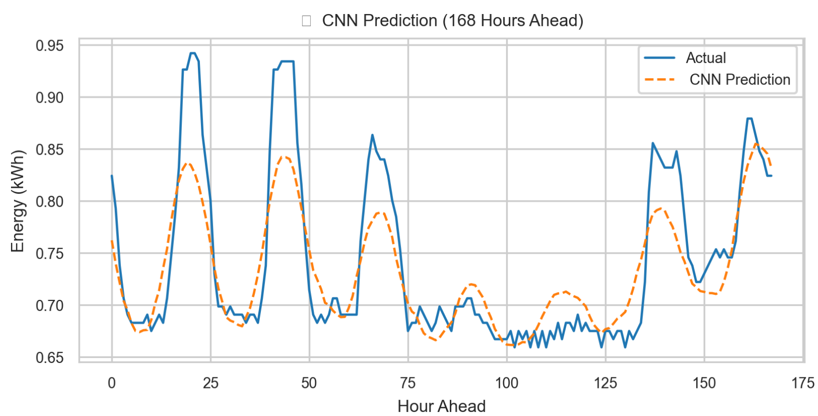

Figure 3: CNN consumption forecast vs. actual (168-hour horizon).

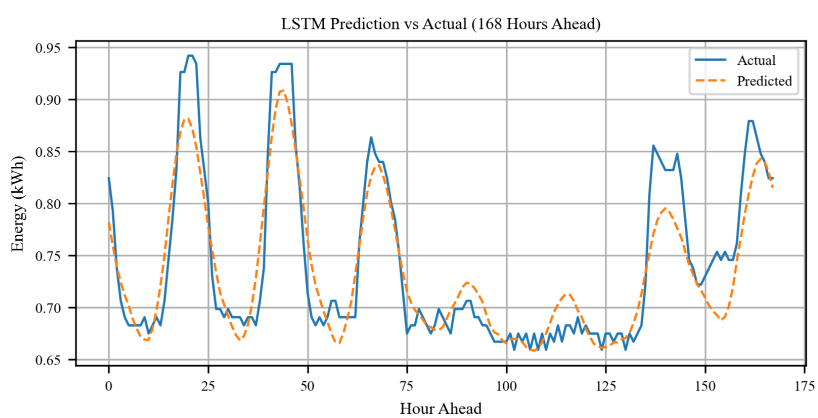

Figure 4: LSTM consumption forecast vs. actual (168-hour horizon).

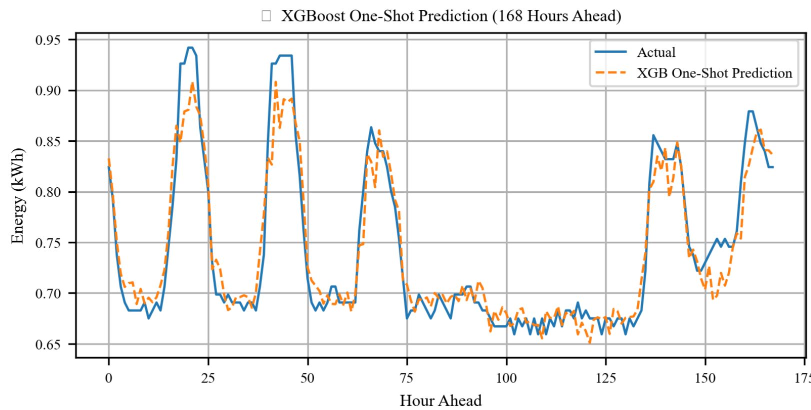

Figure 5: XGBoost consumption forecast vs. actual (168-hour horizon).

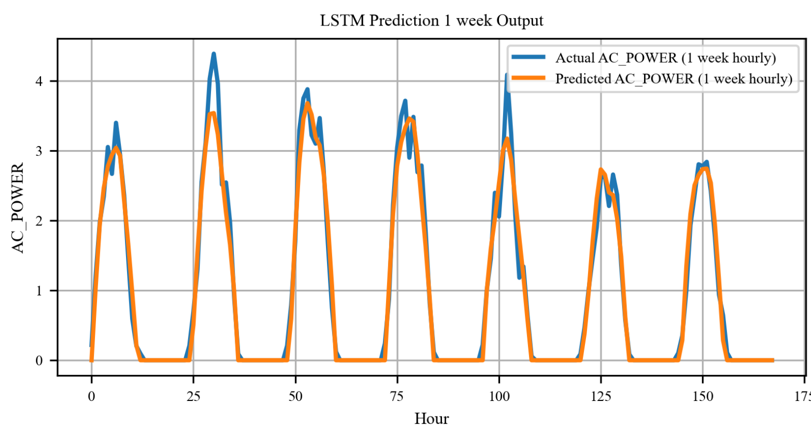

Figure 6: LSTM solar panel generation prediction vs. actual (168-hour horizon).

6.3 Interface Alerts and Operational Use

A warning panel summarized forecasted low state-of-charge (SoC) intervals over the 168 h horizon. Contiguous hours were aggregated into intervals and labeled by severity: advisory for 20–50% (amber) and critical for < 20% (red). Alerts were computed from the net-load forecast and the simulated SoC trajectory under power and capacity limits, and were used to inform the downstream scheduler. Following each alert, a grid-support module proposed secondary-source actions to sustain service and protect battery health as illustrated in FIGURE 7.

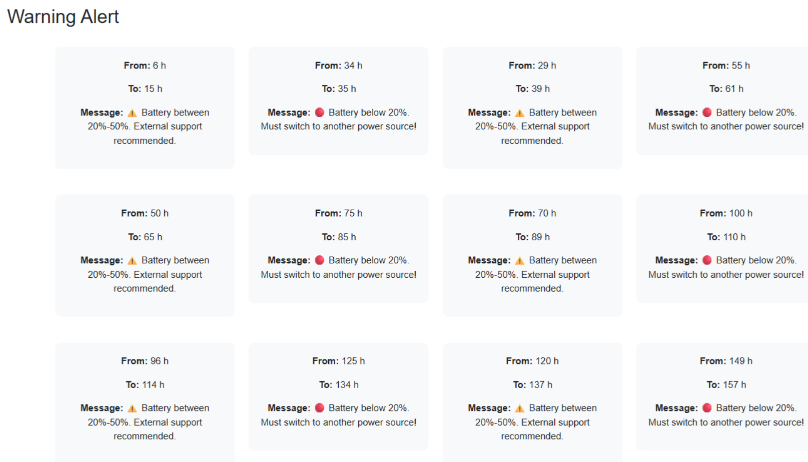

Figure 7: Low-SoC warning panel over the next 168 h.

7. CONCLUSION

Accurate forecasting of household energy demand and solar power generation is the key to further advancing the smart home energy management system and supporting grid stability. The proposed approach integrates deep learning and ensemble learning—LSTM (Long Short-Term Memory) networks, Convolutional Neural Networks (CNN), and eXtreme Gradient Boosting (XGBoost)—for the first time into one multi-step forecasting framework. Experiments indicate the fact that while CNN capture sharp local variations and XGBoost handles stable patterns effectively, LSTMs consistently achieve the strongest overall performance by balancing short-term fluctuations with long-term temporal dependencies. Future research could focus on real-time integration of IoT and smart meter data, scaling the framework to larger communities for collective energy management. In addition, developing mobile interfaces with built-in optimization for appliance scheduling and battery could boost real-world adoption.

References

[1] Siano P. Demand Response and Smart Grids – A Survey. Renew Sustain Energy Rev. 2014;30:461-478.

[2] https://www.iea.org/reports/global-energy-review-2021?edition=2022.

[3] Albadi MH, El-Saadany EF. Demand Response and Smart Grids – A Surveya Summary of Demand Response in Electricity Markets. Electr Power Syst Res. 2008;78:1989-1996.

[4] Ehsan A, Yang Q. Optimal Integration and Planning of Renewable Distributed Generation in the Power Distribution Networks: A Review of Analytical Techniques. Applied Energy. 2018;210:44–59.

[5] Rahm E, Do HH. Data Cleaning: Problems and Current Approaches. IEEE Data Eng Bull. 2000;23:3-13.

[6] Pipattanasomporn M, Kuzlu M, Rahman S. An Algorithm for Intelligent Home Energy Man Agement and Demand Response Analysis. IEEE Trans Smart Grid. 2012;3:2166-2173.

[7] Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Comput. 1997;9:1735-1780.

[8] Bai S, Kolter JZ, Koltun V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. arXiv preprint arXiv: https://arxiv.org/pdf/1803.01271

[9] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM. 2016:785-794.

[10] Brown T. Short-Term Load Forecasting Using Regression Techniques. Energy Rep. 2020;6:1-10.

[11] Bourhnane S, Abid MR, Lghoul R, Elkamoun N, Zine-Dine K, Benhaddou D. Machine Learning for Energy Consumption Prediction and Scheduling in Smart Buildings. SN Appl Sci. 2020;2:297.

[12] Elhabyb, K., Baina, A., Bellafkih, M., & Deifalla, A. F. Machine Learning Algorithms for Predicting Energy Consumption in Educational Buildings. International Journal of Energy Research. 2024(1), 6812425.

[13] Paudel S, Nguyen PH, Kling WL, Elmitri M, Lacarrière B, Le Corre O. Support Vector Machine in Prediction of Building Energy Demand Using Pseudo Dynamic Approach. 2015. ArXiv preprint: https://arxiv.org/pdf/1507.05019

[14] Shapi MK, Ramli NA, Awalin LJ. Energy Consumption Prediction by Using Machine Learning for Smart Building: Case Study in Malaysia. Dev Built Environ. 2021;5:100037.

[15] Olu-Ajayi R, Alaka H, Sulaimon I, Sunmola F, Ajayi S. Building Energy Consumption Prediction for Residential Buildings Using Deep Learning and Other Machine Learning Techniques. J Build Eng. 2022;45:103406.

[16] Boopathi AM, Ali ME, Velappan S, Abudhahir A. Forecasting the Generation and Consumption of Electricity and Water in Kingdom of Bahrain Using Grey Models. Int J Comput Digit Syst. 2021;10:1-8.

[17] Sivakoti K, Mozumdar M. Load Prediction in Smart Grid Networks. Int J Comput Digit Syst. 2015;4:245-250.

[18] Abumohsen M, Owda AY, Owda M. Electrical Load Forecasting Using LSTM, GRU, and RNN Algorithms. Energies. 2023;16:2283.

[19] Ndife AN, Rakwichian W, Muneesawang P, Mensin Y. Smart Power Consumption Forecast Model With Optimized Weighted Average Ensemble. Int J Artif Intell. 2022;11(3):1004–1018.

[20] Kong W, Dong ZY, Jia Y, Hill DJ, Xu Y, et al. Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network. IEEE Trans Smart Grid. 2019;10:841-851.

[21] Shi H, Xu M, Li R.Deep Learning for Household Load Forecasting – A Novel Pooling Deep RNN. IEEE Trans Smart Grid. 2018;9:5271-5280.

[22] Kim TY, Cho SB. Predicting Residential Energy Consumption Using CNN-LSTM Neural Networks. Energy. 2019;182:72-81.

[23] Kim JY, Cho SB. Electric Energy Consumption Prediction by Deep Learning With State Explainable Autoencoder. Energies. 2019;12:739.

[24] Neshat M, Thilakaratne M, El-Abd M, Mirjalili S, Gandomi AH, Boland J. Smart Buildings Energy Consumption Forecasting Using Adaptive Evolutionary Ensemble Learning Models. 2025. ArXiv preprint: https://arxiv.org/pdf/2506.11864

[25] Liu Y, Wang Y, Xu P, Xu Y, Chen Y, Zhang D. BuildSTG: A Multi-Building Energy Load Forecasting Method Using Spatio-Temporal Graph Neural Network. 2025. ArXiv preprint: https://arxiv.org/pdf/2507.20838

[26] Liang Z, Chen J. Research on Building Energy Consumption Prediction Algorithm Based on Customized Deep Learning Model. Energy Inform. 2025;8:25.

[27] Duong VH, Nguyen NH. Machine Learning Algorithms for Electrical Appliances Monitoring System Using Open-Source Systems. Int J Artif Intell (IJAI). 2022;11:300-309.

[28] Alghamdi MA, Al-Malaise Al-Ghamdi AS, Ragab M. Predicting Energy Consumption Using Stacked LSTM Snapshot Ensemble. Big Data Mining and Analytics. 2024;7:247–270.