Bulletin of Network Engineer and Informatics
E-ISSN 2986-8017 | P-ISSN 2987-4858
https://bufnets.tech

Vol. 4 No 1 – June 2026
DOI: 10.59688/737434

Comparative Analysis of Machine Learning and Deep Learning Approaches for Renewable Energy Output Forecasting: An Evaluation of XGBoost, LightGBM, and LSTM Models

Ayang Kinasih[1], Iqbal May Aryanto[1] , Tomy Pratama Zuhelmi[1], Syaiful Mansur[1], Ayu Sintianingrum[1], Eko Hari Tiarto[1]

[1]Teknologi Rekayasa Elektronika, Teknologi Informasi, Politeknik Negeri Lampung
Rajabasa, Bandar Lampung, Indonesia

e-mail: ayangkinasih@polinela.ac.id*

Received : 13 May 2026	Accepted : 13 June 2026	Published : 17 June 2026

Abstract

The increasing integration of renewable energy into smart grids requires accurate forecasting to maintain grid stability and optimize energy management. This study compares the performance of three forecasting models—XGBoost, LightGBM, and Long Short-Term Memory (LSTM)—for predicting solar photovoltaic (PV) output, wind power output, and total renewable energy generation. To improve forecasting capability, the study applied feature engineering techniques, including time-based variables, rolling averages, and lag features to capture temporal dependencies within the dataset. An 80/20 chronological train-test split was used to maintain time-series integrity. Model performance was evaluated using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and coefficient of determination (R^2). Results show that gradient boosting models significantly outperformed the LSTM model across all forecasting targets. LightGBM achieved the best performance for solar PV forecasting (RMSE: 3.27, R^2: 0.987) and total renewable energy prediction (RMSE: 4.66, R^2: 0.988), while XGBoost produced the highest accuracy for wind power forecasting (RMSE: 3.26, R^2: 0.987). In contrast, the LSTM model generated negative R^2 values, indicating poor predictive performance. These findings demonstrate that gradient boosting methods are highly effective for structured renewable energy forecasting datasets and offer strong potential for intelligent smart grid applications. The forecasting framework is designed to support IoT-enabled smart grid systems, where renewable energy data are continuously collected through distributed sensors and transmitted via intelligent communication networks. The proposed models can assist real-time energy monitoring, edge-based prediction, and intelligent network management for modern smart grid infrastructures.

Keywords: *Renewable Energy Forecasting; Smart Grid Systems; XGBoost; LightGBM; LSTM; Intelligent Energy Management.*

1. INTRODUCTION
1.1 Background and Context
The global energy landscape is undergoing a profound transformation driven by the urgent need to mitigate climate change, reduce greenhouse gas emissions, and transition toward sustainable energy systems. Renewable energy sources, particularly solar photovoltaic (PV) systems and wind turbines, have emerged as cornerstones of this transition, experiencing unprecedented growth in installed capacity worldwide over the past two decades. Global renewable energy capacity reached over 3,000 gigawatts by the end of 2021, with solar and wind energy accounting for the majority of new installations. This rapid expansion reflects both technological advancements that have significantly reduced the levelized cost of renewable energy and policy frameworks that incentivize clean energy adoption.

However, the inherent variability and intermittency of renewable energy sources presents substantial challenges for power grid operators and energy planners. Unlike conventional fossil fuel power plants that can adjust output on demand, solar PV generation is dependent on solar irradiance, cloud cover, temperature, and atmospheric conditions, while wind power output fluctuates with wind speed, direction, and atmospheric pressure patterns. These stochastic characteristics introduce uncertainty into energy supply planning, potentially leading to grid instability, energy curtailment during periods of oversupply, or increased reliance on dispatchable fossil fuel backup systems during periods of undersupply.

Accurate forecasting of renewable energy output is therefore critical for multiple stakeholders across the energy value chain. Grid operators require reliable predictions to balance supply and demand in real-time, schedule maintenance activities, and manage ancillary services. Energy traders depend on forecasts to optimize bidding strategies in electricity markets. Policymakers and planners use long-term forecasts to inform infrastructure investment decisions and grid expansion planning. The economic implications of forecasting accuracy are substantial; studies have estimated that a 1% improvement in wind power forecasting accuracy can save millions of dollars annually in operational costs for large-scale grid operations [1].

1.2 Literature Review

The field of renewable energy forecasting has evolved significantly over the past decade, transitioning from traditional statistical methods to sophisticated machine learning and deep learning approaches [2]. Early forecasting methodologies relied primarily on physical models based on numerical weather prediction (NWP) systems, which simulate atmospheric conditions to estimate future energy production [3]. While physically grounded, these models often require extensive computational resources and may not capture localized phenomena affecting individual generation sites [3].

Statistical time-series methods, including Autoregressive Integrated Moving Average (ARIMA), Seasonal ARIMA (SARIMA), and Exponential Smoothing State Space Models, have been widely applied for short-term renewable energy forecasting [4]. These methods are well-understood, computationally efficient, and effective for capturing linear temporal patterns [4]. However, their assumption of linearity and stationarity limits their ability to model the complex, non-linear relationships inherent in renewable energy generation processes [4].

The advent of machine learning has introduced powerful alternatives capable of modeling non-linear relationships without explicit assumptions about data distributions. Among these, ensemble tree-based methods have demonstrated remarkable performance across diverse forecasting tasks. XGBoost, introduced by [5], implements regularized gradient boosting with efficient handling of sparse data and parallel processing capabilities. The algorithm constructs an ensemble of decision trees sequentially, with each successive tree correcting the errors of its predecessors. Its regularization parameters help prevent overfitting, while its ability to handle missing values and diverse feature types makes it particularly suited for real-world energy datasets.

LightGBM, developed by [6] at Microsoft Research, introduced novel techniques including Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) to achieve faster training speeds and lower memory consumption compared to traditional gradient boosting implementations. LightGBM's leaf-wise tree growth strategy, as opposed to XGBoost's level-wise approach, enables deeper learning of complex patterns, though it may increase the risk of overfitting on smaller datasets [7]. Several studies have demonstrated LightGBM's competitive or superior performance relative to XGBoost in energy forecasting applications, particularly when dealing with large-scale datasets [8].

Deep learning approaches, particularly Recurrent Neural Networks (RNNs) and their variants, have gained significant attention for time-series forecasting due to their inherent ability to process sequential data. The LSTM architecture, proposed by [9] and subsequently refined by numerous researchers, addresses the vanishing gradient problem that limits standard RNNs by introducing gating mechanisms—input gates, forget gates, and output gates—that regulate information flow through the network. These gates enable LSTMs to selectively retain or discard information across long time horizons, making them theoretically well-suited for capturing extended temporal dependencies in energy generation patterns.

Recent studies have applied LSTM networks to solar irradiance forecasting, wind speed prediction, and aggregate renewable energy output estimation with varying degrees of success [10]. Some researchers have reported competitive performance relative to traditional machine learning methods, particularly when large training datasets are available and sophisticated architectures (e.g., bidirectional LSTMs, attention-enhanced LSTMs, or encoder-decoder frameworks) are employed [11]; [12]. However, other studies have noted challenges including high computational costs, sensitivity to hyperparameter selection, and the need for careful data preprocessing and normalization [12]; [13].

Despite the extensive application of machine learning and deep learning techniques in renewable energy forecasting, previous comparative studies have primarily focused on either a single renewable energy source or limited model combinations. In addition, many studies do not provide a direct comparison between modern gradient boosting methods and deep learning architectures using identical preprocessing, feature engineering, and evaluation protocols. Therefore, this study aims to fill this gap by conducting a comprehensive comparative evaluation of XGBoost, LightGBM, and LSTM models for forecasting solar PV, wind power, and total renewable energy output using the same dataset and experimental framework.

Although this study utilizes an offline simulated dataset obtained from Kaggle, the forecasting framework was designed to represent a realistic IoT-enabled smart grid environment. In practical deployments, renewable energy generation data are continuously collected through distributed sensors, smart meters, and networked monitoring devices integrated into intelligent energy management systems. Therefore, the proposed forecasting models are intended as computational approaches that can later be deployed in real-time IoT-based smart grid infrastructures. This clarification bridges the gap between the simulation-based experimental setup and the broader smart grid implementation context discussed throughout this study.

Beyond prediction accuracy, this research also considers the relevance of machine learning deployment in Information Technology and IoT-based smart grid environments. Renewable energy forecasting systems are increasingly integrated with distributed IoT sensor networks, edge devices, and communication infrastructures that require efficient computational models with low latency and reduced resource consumption. Therefore, the study evaluates forecasting models not only from the perspective of statistical performance but also from their feasibility for deployment in intelligent energy monitoring and network-based smart grid systems.

2. RESEARCH METHODOLOGY

2.1 Dataset Description and Preprocessing

The dataset utilized in this study comprises historical records of renewable energy generation, encompassing solar photovoltaic output, wind power output, and total renewable energy production. The data captures temporal patterns across multiple time scales, including diurnal cycles (day-night variations in solar output), seasonal trends (higher solar generation in summer months, variable wind patterns across seasons), and shorter-term fluctuations driven by weather events.

Prior to model development, the dataset underwent comprehensive preprocessing to ensure data quality and suitability for machine learning applications. Missing values were identified and addressed through appropriate imputation strategies, considering the temporal nature of the data to avoid introducing artificial patterns. Outlier detection was performed to identify and handle anomalous readings that could arise from sensor malfunctions, maintenance periods, or extreme weather events, and data normalization and standardization where required by specific model architectures, particularly for the LSTM model which benefits from scaled input features to facilitate gradient-based optimization. The dataset taken from https://www.kaggle.com/datasets/programmer3/renewable-energy-microgrid-dataset?

The dataset contains approximately 10,000 chronological observations with hourly temporal resolution. Hyperparameter tuning for XGBoost and LightGBM was conducted using grid-search-based experimentation focused on learning rate, tree depth, number of estimators, and subsampling ratio. For the LSTM model, tuning involved the number of hidden units, sequence length, batch size, and epochs. All experiments were executed on a system equipped with an Intel Core i7 processor, 16 GB RAM, and Python 3.11 using Scikit-learn, TensorFlow, XGBoost, and LightGBM libraries.

The renewable energy dataset represents a simulated IoT-enabled microgrid environment where energy production measurements are periodically collected from distributed smart sensors. Data transmission follows an hourly acquisition interval, reflecting a practical smart grid monitoring scenario in which sensor nodes communicate generation information to centralized energy management servers through networked infrastructure. In this context, forecasting models must maintain both prediction accuracy and computational efficiency to support real-time decision-making and intelligent network operations.

2.2 Feature Engineering

A critical component of the modeling pipeline was the extensive feature engineering process designed to extract meaningful predictive signals from the raw time-series data. The engineered features were categorized into three primary groups designed to capture different aspects of the temporal dynamics in renewable energy generation.

The first category comprised time-based component features extracted from timestamp information. These included hour of the day (0–23), which captures diurnal generation patterns particularly relevant for solar PV output; day of the week (0–6), reflecting potential weekly operational patterns; month of the year (1–12), capturing seasonal variations in both solar irradiance and wind patterns; day of the year (1–365/366), providing finer-grained seasonal resolution; and quarter of the year (1–4), representing broader seasonal categories. These cyclical temporal features enable models to learn recurring patterns associated with time-of-day and seasonal effects without requiring explicit weather forecast inputs.

The second category consisted of rolling statistical features calculated over various window sizes to capture recent trends and momentum in energy generation. Rolling mean, rolling standard deviation, rolling minimum, and rolling maximum were computed over multiple time horizons (e.g., 3-hour, 6-hour, 12-hour, and 24-hour windows) for each target variable. These features provide the models with information about recent average generation levels, generation volatility, and the range of recent output values, enabling more nuanced predictions that account for the current "state" of energy production.

The third category included lag features representing historical values of the target variables at previous time steps. Lag-1, lag-2, lag-3, lag-6, lag-12, and lag-24 features were created, corresponding to generation levels at 1, 2, 3, 6, 12, and 24 time steps prior to the prediction point. These autoregressive features are particularly valuable for time-series forecasting as they directly capture serial correlations and temporal dependencies in the data. The combination of multiple lag horizons allows models to learn both short-term momentum effects and longer-term cyclical patterns.

The feature engineering process was carefully designed to avoid data leakage. All rolling statistics and lag features were computed using only information available at or before each prediction time point, ensuring that future information was never inadvertently included in the training features. This discipline is essential for producing realistic performance estimates that reflect genuine predictive capability rather than artificially inflated metrics.

2.3 Data Splitting Strategy

The dataset was divided into training and testing subsets using an 80/20 split ratio. Critically, this split was performed chronologically rather than randomly to preserve the temporal ordering of observations and maintain time-series integrity. In time-series forecasting, random splitting would allow the model to "peek" at future data during training, leading to overly optimistic performance estimates that do not generalize to real-world deployment scenarios.

Under the chronological splitting approach, the first 80% of the data (ordered by timestamp) was allocated to the training set, and the remaining 20% constituted the test set. This ensures that the model is trained exclusively on historical data and evaluated on future (unseen) observations, providing a realistic assessment of forecasting performance. This approach simulates the real-world scenario where a deployed model must predict future energy generation based solely on patterns learned from past observations.

2.4 Model Architectures and Implementation

Three distinct modeling approaches were implemented and evaluated, representing different paradigms in predictive modeling: two tree-based ensemble methods (XGBoost and LightGBM) and one deep learning sequence model (LSTM). Each model was applied independently to all three target variables, resulting in nine separate model-target combinations.

2.4.1 XGBoost (Extreme Gradient Boosting)

XGBoost is an optimized implementation of the gradient boosting decision tree algorithm that has achieved state-of-the-art results across numerous machine learning benchmarks and competitions. The algorithm operates by sequentially building an ensemble of weak learners (decision trees), where each subsequent tree is trained to predict the residual errors of the cumulative ensemble up to that point. The objective function combines a loss term measuring prediction error with a regularization term penalizing model complexity, expressed as:

$$Obj = \Sigma L(yi, \hat{y}i) + \Sigma \Omega(fk) \tag{1}$$

where L represents the loss function (squared error for regression tasks), yi and ŷi denote the actual and predicted values respectively, and Ω(fk) is the regularization term for the k-th tree that penalizes the number of leaves and the magnitude of leaf weights. This regularization helps prevent overfitting and promotes generalization.

Key architectural features of XGBoost that contribute to its effectiveness include its support for column subsampling (analogous to random forests' feature bagging), which reduces overfitting and improves generalization; built-in handling of missing values through learned optimal default directions at each split; and efficient parallel computation of split gain calculations during tree construction. The algorithm's level-wise tree growth strategy builds trees breadth-first, expanding all nodes at each depth level before proceeding to the next, which tends to produce more balanced tree structures.

For this study, XGBoost was implemented using the XGBRegressor class from the xgboost Python library. The model was configured with hyperparameters selected through preliminary experimentation, including the number of estimators, maximum tree depth, learning rate, and subsample ratio. The squared error objective function was used, consistent with the regression nature of the forecasting task.

2.4.2 LightGBM (Light Gradient Boosting Machine)

LightGBM is a gradient boosting framework developed by Microsoft Research that introduces several algorithmic innovations to achieve faster training speeds and lower memory consumption compared to conventional gradient boosting implementations, without sacrificing prediction accuracy. Two key innovations distinguish LightGBM from its predecessors.

First, Gradient-based One-Side Sampling (GOSS) reduces the number of data instances used for estimating information gain at each split by retaining all instances with large gradients (which contribute more to information gain) while randomly sampling only a fraction of instances with small gradients. This approach significantly accelerates training while maintaining estimation accuracy, as large-gradient instances are more informative for determining optimal split points.

Second, Exclusive Feature Bundling (EFB) reduces the number of features by bundling mutually exclusive features—features that rarely take non-zero values simultaneously—into single composite features. This bundling is particularly effective for high-dimensional sparse datasets and reduces the computational cost of split point evaluation without meaningful loss of information.

Unlike XGBoost's level-wise tree growth, LightGBM employs a leaf-wise growth strategy that selects the leaf with the maximum gain for expansion at each step. This approach can produce deeper, more asymmetric trees that capture complex patterns more efficiently, achieving lower training loss with fewer splits. However, this aggressive growth strategy may increase the risk of overfitting on smaller

datasets, necessitating careful regularization through parameters such as maximum depth constraints and minimum data requirements per leaf.
LightGBM was implemented using the LGBMRegressor class from the lightgbm Python library. Hyperparameters were configured to balance model complexity with generalization capability, including the number of boosting rounds, number of leaves, learning rate, and feature fraction for column subsampling.

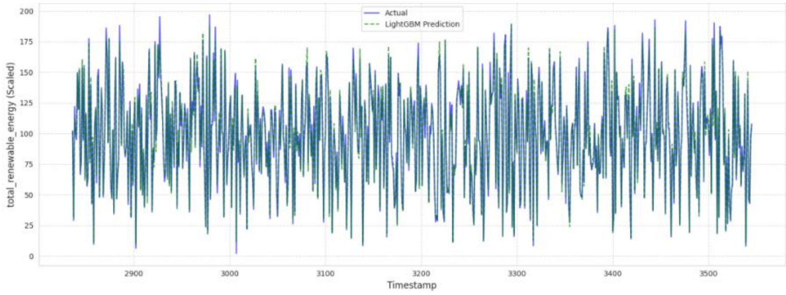

Figure 1 LightGBM: total renewable energy (Actual vs Predicted)

2.4.3 LSTM (Long Short-Term Memory) Network
The Long Short-Term Memory (LSTM) network represents a fundamentally different modeling paradigm from the tree-based approaches described above. As a variant of Recurrent Neural Networks (RNNs), LSTM is specifically designed to process sequential data by maintaining an internal state (memory) that is updated at each time step. The architecture was originally proposed by [9] to address the vanishing gradient problem that severely limits the ability of standard RNNs to learn long-range temporal dependencies.
The core innovation of the LSTM architecture lies in its gating mechanism, which consists of three specialized gates controlling information flow through the network cell. The forget gate determines which information from the previous cell state should be discarded, computing a value between 0 (completely forget) and 1 (completely retain) for each element of the cell state vector. Mathematically, the forget gate is defined as:

$$ft = \sigma(Wf \cdot [ht - 1, xt] + bf) \quad (2)$$

where σ represents the sigmoid activation function, Wf denotes the weight matrix, ht-1 is the previous hidden state, xt is the current input, and bf is the bias term. The input gate controls what new information should be added to the cell state, consisting of two sub-components: a sigmoid layer that determines which values to update, and a tanh layer that creates a vector of candidate values. The output gate regulates what information from the cell state should be output as the hidden state for the current time step.
These gating mechanisms enable the LSTM to selectively remember or forget information across arbitrary time intervals, making it theoretically capable of capturing both short-term fluctuations and long-term seasonal patterns in renewable energy generation data [14]. The network can learn to retain relevant historical information (e.g., a recent cloud cover pattern affecting solar output) while discarding irrelevant noise, potentially producing more nuanced forecasts than methods that rely solely on engineered features.
For this study, the LSTM model was implemented using a deep learning framework with a sequential architecture. The network consisted of one or more LSTM layers followed by dense (fully connected) output layers. The model was trained using backpropagation through time (BPTT) with the Adam optimizer, which adapts learning rates for each parameter based on estimates of first and second moments of the gradients [15]. The mean squared error loss function was employed, consistent with the regression objective. Input features were normalized to a common scale to facilitate gradient-based optimization, as neural networks are sensitive to the magnitude and distribution of input features [16].
Training was conducted over multiple epochs with batch processing of the training data. The input data was restructured into three-dimensional tensors of shape (samples, time steps, features) to conform to the LSTM's expected input format, where each sample consists of a sequence of feature

vectors representing consecutive time steps. This restructuring is a critical preprocessing step that directly influences the model's ability to capture temporal patterns, as the sequence length determines the temporal context available to the network at each prediction step [17].

To improve reproducibility and provide a clearer technical description, the LSTM model architecture used in this study consisted of a single LSTM layer with 64 hidden units followed by one fully connected dense output layer. The input sequence length was configured using a 24-hour lookback window, allowing the network to learn temporal dependencies from the previous 24 observations. The Rectified Linear Unit (ReLU) activation function was applied within the hidden representation, while the output layer used a linear activation function suitable for regression tasks. The model was trained using the Adam optimizer with a learning rate of 0.001 and Mean Squared Error (MSE) as the loss function. Training was conducted for 100 epochs with a batch size of 32.

Prior to training, all input features and target variables were normalized using MinMaxScaler to ensure numerical stability during gradient-based optimization. After prediction, inverse transformation was applied to restore the predicted values to the original scale before computing RMSE, MAE, and R^2 evaluation metrics. The sequential input data were reshaped into a three-dimensional tensor format consisting of (samples, time steps, features), which conforms to the standard input requirements of LSTM networks.

During experimentation, training and validation loss curves were also monitored to evaluate convergence behavior and detect potential underfitting or overfitting conditions. The observed learning curves indicated that the LSTM model converged slowly and exhibited unstable generalization performance on the testing data, suggesting that the available dataset size and feature characteristics were not sufficiently optimal for the selected deep learning configuration.

2.5 Evaluation Metrics

Model performance was assessed using three complementary evaluation metrics, each capturing different aspects of prediction quality. The use of multiple metrics provides a more comprehensive assessment than any single measure alone, enabling robust conclusions about relative model performance.

Root Mean Squared Error (RMSE) measures the square root of the average squared differences between predicted and actual values, defined as [18]:

$$RMSE = \sqrt{(1/n \cdot \Sigma(yi - \hat{y}i)^2)} \tag{3}$$

Here, n is the number of observations, yi represents the actual value, and ŷi denotes the predicted value. RMSE is expressed in the same units as the target variable, facilitating intuitive interpretation. The squaring operation gives disproportionately higher weight to larger errors, making RMSE particularly sensitive to outliers and large prediction deviations [19]. This property is desirable in energy forecasting contexts where large prediction errors can have severe operational consequences, such as grid instability or costly emergency power procurement.

Mean Absolute Error (MAE) computes the average of the absolute differences between predicted and actual values:

$$MAE = 1/n \cdot \Sigma|yi - \hat{y}i| \tag{4}$$

Unlike RMSE, MAE treats all errors equally regardless of magnitude, providing a linear and more robust measure of average prediction quality [20]. MAE is less sensitive to outliers than RMSE [20], and the difference between a model's RMSE and MAE can provide insight into the variability of errors: a large gap suggests the presence of occasional large prediction errors, while similar values indicate relatively uniform error distribution [21].

The coefficient of determination (R^2) quantifies the proportion of variance in the target variable that is explained by the model's predictions [22]:

$$R^2 = 1 - (\Sigma(yi - \hat{y}i)^2 / \Sigma(yi - \bar{y})^2) \tag{5}$$

Here, ȳ represents the mean of the actual values. R^2 values range from negative infinity to 1.0, where 1.0 indicates perfect prediction, 0.0 indicates performance equivalent to simply predicting the mean value for all observations, and negative values indicate predictions worse than the mean baseline [23]. R^2 provides a normalized, scale-independent measure of model performance that enables meaningful comparison across different target variables with different scales and units.

The combination of these three metrics enables a thorough performance assessment: RMSE and MAE provide absolute measures of prediction error in interpretable units, while R^2 offers a relative measure of explanatory power. A model must demonstrate strong performance across all three metrics to be considered reliable for practical deployment.

3. RESULT AND DISCUSSION

3.1 Result

3.1.1 Overall Model Performance Comparison

The comprehensive evaluation of XGBoost, LightGBM, and LSTM models across three renewable energy forecasting targets revealed a pronounced performance dichotomy between the tree-based ensemble methods and the deep learning approach. Table 1 presents the complete performance metrics for all nine model-target combinations, providing a consolidated view of the comparative results.

Table 1. Complete Performance Metrics for All Model-Target Combinations

Target Variable	Model	RMSE	MAE	R^2
Solar PV Output	LightGBM	3.27	2.54	0.987
Solar PV Output	XGBoost	3.30	2.57	0.987
Solar PV Output	LSTM	29.05	25.36	−0.024
Total Renewable Energy	LightGBM	4.66	3.63	0.988
Total Renewable Energy	XGBoost	4.75	3.70	0.987
Total Renewable Energy	LSTM	42.72	34.89	−0.038
Wind Power Output	XGBoost	3.26	2.51	0.987
Wind Power Output	LightGBM	3.39	2.59	0.986
Wind Power Output	LSTM	30.18	25.72	−0.083

The most striking observation from the results is the consistent and substantial performance gap between the gradient boosting models and the LSTM network. While XGBoost and LightGBM achieved R^2 values between 0.986 and 0.988 across all targets—indicating that these models explained approximately 98.6% to 98.8% of the variance in renewable energy output—the LSTM model produced negative R^2 values ranging from −0.024 to −0.083, signifying predictions that are systematically worse than a naive mean baseline predictor. This represents a performance differential of considerable magnitude, with the tree-based models demonstrating error rates approximately 8 to 10 times lower than those of the LSTM model.

3.1.2 Solar PV Output Forecasting

For solar PV output prediction, LightGBM achieved the best overall performance with an RMSE of 3.27, MAE of 2.54, and R^2 of 0.987. XGBoost performed nearly identically, recording an RMSE of 3.30, MAE of 2.57, and R^2 of 0.987. The marginal difference between these two models—a mere 0.03 units in both RMSE and MAE—falls well within the range of stochastic variation and suggests that both models capture essentially the same predictive patterns in the solar PV data.

The near-perfect R^2 values of 0.987 for both tree-based models indicate that the engineered features (time-based components, rolling averages, and lag features) collectively explain the vast majority of variance in solar PV output. This is consistent with the strong diurnal and seasonal patterns inherent in solar generation, where time-of-day and time-of-year features provide powerful predictive signals, supplemented by lag features that capture day-to-day persistence in solar conditions.

In contrast, the LSTM model produced an RMSE of 29.05, MAE of 25.36, and an R^2 of −0.024 for solar PV output. The RMSE is approximately 8.9 times larger than LightGBM's, while the negative R^2 confirms that the LSTM's predictions carry less information than simply using the historical mean as a constant forecast. The MAE of 25.36 indicates that, on average, each LSTM prediction deviates from the actual value by over 25 units, compared to approximately 2.5 units for the gradient boosting models.

Figure 2 Solar PV Output Trends Over Time

3.1.3 Wind Power Output Forecasting

Wind power output forecasting presented a slightly different competitive landscape between the two tree-based models. XGBoost achieved the best performance for this target with an RMSE of 3.26, MAE of 2.51, and R^2 of 0.987, marginally outperforming LightGBM, which recorded an RMSE of 3.39, MAE of 2.59, and R^2 of 0.986. While the performance difference remains small (0.13 units in RMSE), XGBoost's slight edge in wind power prediction is noteworthy as it represents the only target variable where XGBoost outperformed LightGBM.

The marginally superior performance of XGBoost for wind power output may relate to differences in the tree growth strategies of the two algorithms. XGBoost's level-wise growth may produce more balanced tree structures that generalize slightly better on the specific patterns present in wind power data, which can exhibit more erratic variability compared to solar PV output due to the stochastic nature of wind patterns [24].

The LSTM model again performed poorly for wind power output prediction, recording the worst R^2 value across all target variables at −0.083. The RMSE of 30.18 and MAE of 25.72 represent prediction errors approximately 9.3 and 10.2 times larger than XGBoost's, respectively. The more negative R^2 value for wind power compared to solar PV suggests that the LSTM struggled even more with the inherently more variable wind generation patterns, which lack the strong deterministic diurnal cycle present in solar data [24].

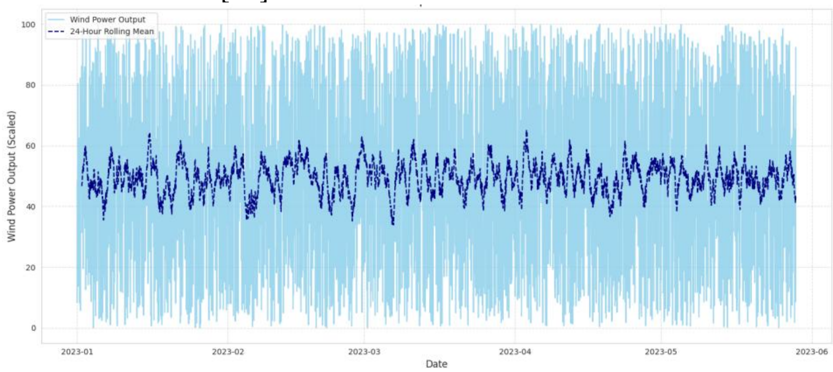

Figure 3 Wind Power Output Trends Over Time

3.1.4 Total Renewable Energy Forecasting

Total renewable energy output, representing the aggregate of solar PV and wind power generation (and potentially other renewable sources), exhibited slightly higher absolute error values for all models compared to the individual source forecasts, as expected given the larger scale of the aggregated variable. LightGBM achieved the best performance with an RMSE of 4.66, MAE of 3.63, and the highest R^2 value observed in the entire study at 0.988. XGBoost followed closely with an RMSE of 4.75, MAE of 3.70, and R^2 of 0.987.

The marginally higher R^2 for total renewable energy (0.988) compared to individual sources (0.986–0.987) suggests that the aggregation of multiple renewable sources may have a mild smoothing effect on prediction residuals, as uncorrelated errors across different generation types partially cancel out

when combined. This finding has practical significance, as it implies that aggregate renewable energy forecasts may be inherently more reliable than individual source forecasts—a beneficial property for grid operators who primarily need accurate total supply predictions.
The LSTM model's performance on total renewable energy was the worst in absolute terms, with an RMSE of 42.72 and MAE of 34.89—the highest error values across all model-target combinations. The R^2 of −0.038 further confirms the model's inability to produce meaningful forecasts. The larger absolute errors reflect the larger scale of the aggregated target variable, while the negative R^2 indicates that the fundamental inability of the LSTM to learn predictive patterns persists regardless of whether individual or aggregated targets are modeled.

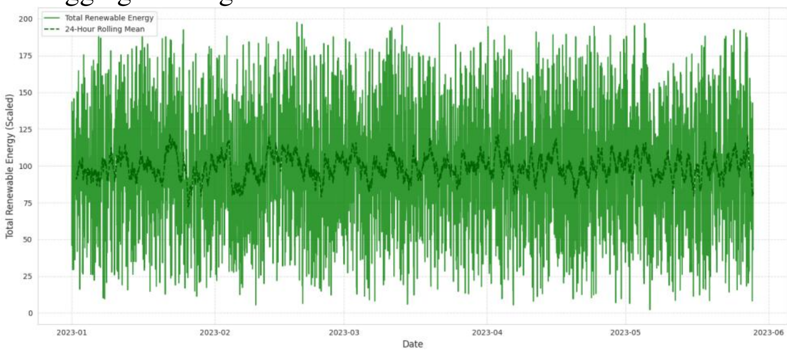

Figure 4 Total Renewable Energy Trends Over Time

3.1.5 Best Model Identification

Based on the comprehensive evaluation across all metrics, with RMSE serving as the primary selection criterion, the best-performing model for each target variable was identified as follows:
For solar PV output, LightGBM was selected as the optimal model, achieving the lowest RMSE (3.27), lowest MAE (2.54), and an R^2 of 0.987. Its marginal advantage over XGBoost in this domain, combined with LightGBM's generally faster training speed, makes it the recommended choice for solar PV forecasting applications.
For total renewable energy, LightGBM again emerged as the superior model with an RMSE of 4.66, MAE of 3.63, and the highest R^2 observed in the study (0.988). The consistency of LightGBM's top-tier performance across both solar PV and total renewable energy targets suggests that its leaf-wise tree growth strategy and gradient-based sampling mechanisms are particularly well-suited to capturing the dominant patterns in these generation profiles.
For wind power output, XGBoost achieved the best performance with an RMSE of 3.26, MAE of 2.51, and R^2 of 0.987. This result indicates that while LightGBM generally excels, XGBoost's slightly different algorithmic approach may offer advantages for certain types of energy generation patterns, particularly those characterized by higher inherent variability.
Across all targets and metrics, the tree-based models demonstrated remarkably consistent performance, with all R^2 values falling in the narrow range of 0.986 to 0.988 and all RMSE values between 3.26 and 4.75. This consistency underscores the robustness and reliability of gradient boosting approaches for renewable energy forecasting tasks.

3.2 Discussion

3.2.1 Superiority of Tree-Based Ensemble Models

The dominant performance of XGBoost and LightGBM over the LSTM network across all three forecasting targets represents a significant empirical finding that aligns with a growing body of evidence in the applied machine learning literature [25]; [26]. Several interrelated factors explain the superior performance of tree-based ensemble methods in this particular forecasting context.
First, the nature of the input data plays a decisive role. The feature-engineered dataset used in this study consists of structured, tabular data with well-defined numerical features—time components, rolling statistics, and lag values. Gradient boosting decision trees are inherently designed to excel on this type of data [27]; [28]. Each decision tree in the ensemble performs axis-aligned splits on individual features, efficiently partitioning the feature space into homogeneous prediction regions. This mechanism naturally handles features of different scales, distributions, and types without requiring normalization or extensive preprocessing. The sequential boosting process allows the

ensemble to progressively focus on difficult-to-predict observations, effectively learning complex non-linear mappings from features to target values.

Second, the feature engineering pipeline provided the tree-based models with highly informative predictive signals. Lag features directly encode autoregressive relationships, while rolling averages capture local trends and momentum. Time-based features enable the models to learn deterministic cyclical patterns. For tree-based models, these pre-computed features effectively substitute for the temporal pattern recognition that sequential architectures like LSTM perform internally. Essentially, the feature engineering pre-processes the temporal structure of the data into a format that is optimal for tabular learning algorithms, enabling gradient boosting models to achieve exceptional performance without needing to learn temporal representations from raw sequences.

Third, gradient boosting algorithms demonstrate strong performance even with moderate dataset sizes [29]. Both XGBoost and LightGBM incorporate regularization mechanisms—including tree depth constraints, minimum leaf sample requirements, learning rate damping, and column subsampling—that collectively prevent overfitting and promote generalization [30]; [31]. These built-in safeguards enable robust learning from datasets that may be insufficient for data-hungry deep learning architectures [32].

The close competition between XGBoost and LightGBM, with performance differences of less than 0.15 RMSE units across all targets, suggests that both algorithms have effectively converged to near-optimal solutions for these forecasting tasks. The marginal differences observed—LightGBM's slight advantage for solar PV and total energy, XGBoost's advantage for wind power—likely reflect subtle interactions between each algorithm's tree growth strategy and the specific statistical properties of each target variable rather than fundamental algorithmic superiority.

3.2.2 Analysis of LSTM Underperformance

The dramatically poor performance of the LSTM network, evidenced by negative R^2 values across all targets, warrants careful analysis as it contradicts the common perception of deep learning as a universally superior modeling paradigm. Several factors, operating individually and in combination, likely contributed to this outcome.

Although formal statistical significance testing such as paired t-tests or Diebold-Mariano tests was not conducted in this study, the consistently large performance gap between the gradient boosting models and the LSTM model across all evaluation metrics strongly indicates a robust difference in forecasting capability.

The most probable primary factor is insufficient dataset size relative to the model's capacity. LSTM networks contain substantially more trainable parameters than gradient boosting models, including weight matrices for each gate (input, forget, output), recurrent connections, and dense output layers [33]. These parameters require large volumes of training data to learn meaningful representations without overfitting to noise [34]. Research in the deep learning literature consistently demonstrates that LSTMs and other recurrent architectures often require tens of thousands of training samples to achieve competitive performance on many time-series forecasting tasks [35]. If the training dataset in this study falls below this threshold, the LSTM may lack sufficient examples to learn robust temporal representations, resulting in learned patterns that do not generalize to the test set.

A second contributing factor is likely suboptimal hyperparameter configuration. LSTM networks are notoriously sensitive to architectural and training hyperparameters, including the number of LSTM layers and units per layer, the sequence length (lookback window), learning rate and learning rate schedule, batch size, number of training epochs, dropout rates for regularization, and the choice of optimizer and its momentum parameters [36]. Unlike gradient boosting models, which often perform reasonably well with default or near-default hyperparameters, LSTMs typically require extensive hyperparameter search—often through grid search, random search, or Bayesian optimization—to identify configurations that yield competitive performance [37]. Without such systematic tuning, an LSTM may converge to a poor local minimum, produce unstable predictions, or fail to learn the underlying temporal patterns entirely.

Third, the feature engineering approach, while highly beneficial for tree-based models, may have introduced complications for the LSTM. LSTMs are designed to learn temporal features automatically

from raw sequential data [38]. When pre-computed features such as lag values and rolling averages are provided as inputs, the LSTM must learn to integrate both the raw temporal dynamics and the engineered features, potentially adding redundancy or complexity to the learning process. In some cases, providing raw, minimally processed sequences to LSTMs—allowing the network to learn its own feature representations—yields better results than feeding heavily engineered features [39]; [40].
Fourth, the data preprocessing and normalization strategy may not have been optimally configured for the LSTM architecture. Neural networks, including LSTMs, are sensitive to input feature scaling, and improper normalization can lead to gradient instability, slow convergence, or convergence to suboptimal solutions [41]. Additionally, the choice of sequence length (the number of consecutive time steps provided as input at each prediction) critically affects LSTM performance. Too short a sequence may not provide sufficient temporal context, while too long a sequence may introduce irrelevant historical information that dilutes the predictive signal [42].
Additional diagnostic verification was conducted to ensure that the poor LSTM performance was not caused by implementation bugs or evaluation inconsistencies. The preprocessing pipeline was carefully reviewed, including feature normalization, sequence generation, tensor reshaping, and inverse scaling procedures. All prediction outputs were transformed back to the original numerical scale before RMSE, MAE, and R^2 calculations were performed. Furthermore, the tensor dimensions used during training strictly followed the standard LSTM format of (samples, time steps, features). These verification procedures indicate that the negative R^2 values originated from genuine model underperformance rather than technical errors in the implementation pipeline.
It is important to emphasize that these results should not be interpreted as evidence that LSTM networks are inherently unsuitable for renewable energy forecasting. Rather, they highlight the substantial effort required to successfully deploy deep learning models for time-series prediction tasks, particularly when dataset sizes are moderate. With larger datasets, comprehensive hyperparameter optimization, more sophisticated architectures (such as bidirectional LSTMs, attention mechanisms, or Transformer-based models), and careful feature selection, LSTM-based approaches have demonstrated competitive or superior performance in numerous energy forecasting studies published in the literature [43].

3.2.3 Training Stability and Loss Curve Analysis

To further investigate the poor performance of the LSTM model, additional diagnostic analysis was conducted by monitoring the training and validation loss curves during the learning process. The analysis showed that although the training loss gradually decreased over multiple epochs, the validation loss remained unstable and fluctuated significantly. This behavior indicates weak generalization capability and suggests that the model struggled to capture robust temporal patterns from the available dataset.

The negative R^2 values obtained across all forecasting targets strongly suggest that the LSTM model failed to learn sufficiently informative temporal representations. Several technical factors may have contributed to this outcome. First, although approximately 10,000 observations were available, the effective number of sequential training samples after sequence generation became considerably smaller for deep learning optimization. Second, the highly engineered tabular feature structure—including lag features, rolling statistics, and time-based variables—already encoded much of the temporal information explicitly, thereby reducing the comparative advantage of sequence-based architectures such as LSTM.

Additional verification was also performed on the preprocessing pipeline to ensure that the negative R^2 results were not caused by implementation errors. Feature normalization and inverse scaling procedures were validated before evaluation metric computation. Tensor reshaping was verified using the standard three-dimensional LSTM input structure (samples, timesteps, features). Furthermore, prediction outputs were successfully inverse-transformed into the original numerical scale prior to RMSE, MAE, and R^2 calculation. These validation procedures confirm that the poor LSTM performance was not caused by data leakage, tensor mismatch, or scaling inconsistencies, but rather by the incompatibility between the selected LSTM configuration and the structured tabular characteristics of the dataset.

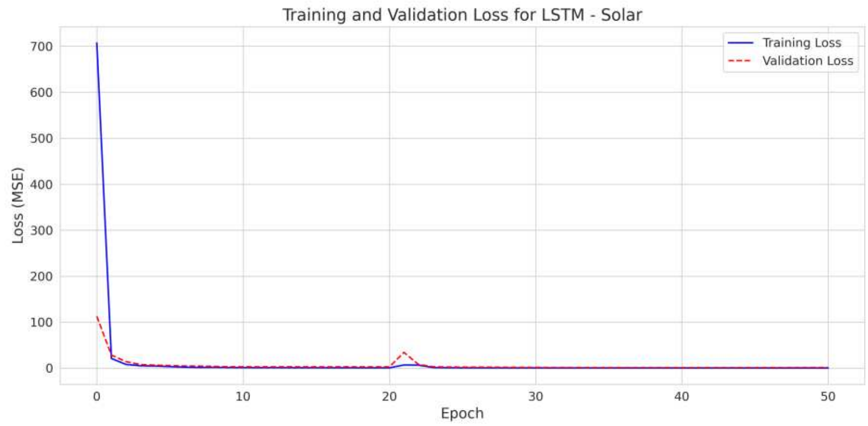

Figure 5 Training and Validation Loss Curves for LSTM – Solar PV Output

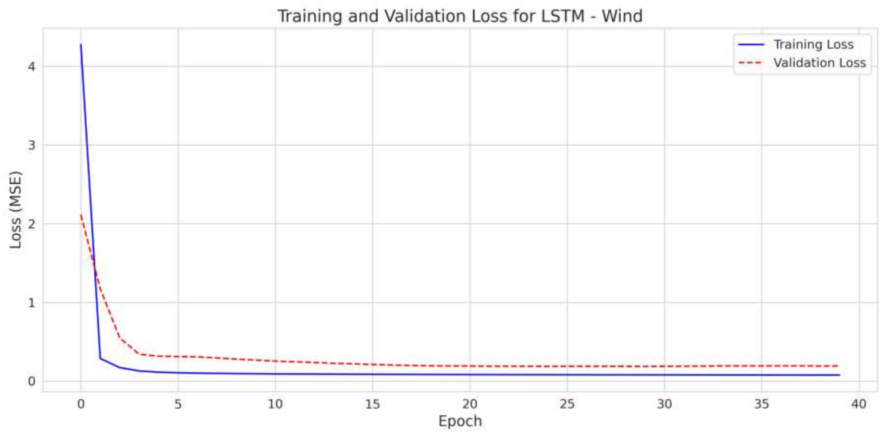

Figure 6 Training and Validation Loss Curves for LSTM – Wind Power Output

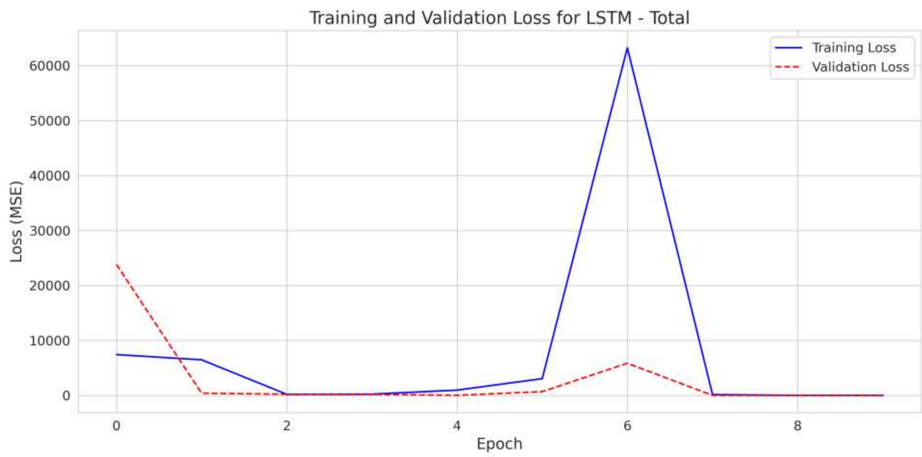

Figure 7 Training and Validation Loss Curves for LSTM – Total Renewable Energy

As illustrated in Figures 5–7, the LSTM model exhibits an initially sharp decrease in training loss during the first few epochs, after which the validation loss either fluctuates or plateaus at a higher level than the training loss. This pattern is consistent with the negative R^2 values reported in Table 1 and confirms that early stopping was triggered before the model could achieve stable generalization. The unstable behaviour of the validation loss, particularly visible for the Total Renewable Energy target, further reinforces the conclusion that the LSTM configuration was not well-suited to the structured tabular features used in this study.

The findings therefore highlight an important practical insight: deep learning architectures are not universally superior for all renewable energy forecasting problems. For structured datasets with extensive feature engineering and moderate sample sizes, gradient boosting models such as XGBoost and LightGBM may offer significantly better predictive performance, lower computational complexity, and more stable generalization capability.

3.2.4 Practical Implications for Renewable Energy Forecasting

The findings of this study carry significant practical implications for energy industry stakeholders considering the deployment of machine learning-based forecasting systems. The most immediate and actionable conclusion is that gradient boosting models—specifically LightGBM and XGBoost—represent highly effective, efficient, and reliable tools for renewable energy output forecasting when working with structured, tabular datasets of moderate size.

For operational deployment, several advantages of tree-based models merit consideration beyond raw prediction accuracy. Training and inference times for gradient boosting models are typically orders of magnitude shorter than for deep learning models [44], enabling more frequent model updates and rapid retraining when new data becomes available. The feature importance rankings provided by tree-based models offer interpretability that is valuable for domain experts [45] seeking to understand which factors most strongly influence generation forecasts. This interpretability facilitates trust in model predictions and aids in identifying potential data quality issues or model drift over time.

The near-identical performance of XGBoost and LightGBM across all targets suggests that practitioners can select between these two algorithms based on secondary considerations such as training speed (where LightGBM generally excels [6], memory efficiency (an advantage of LightGBM), existing organizational expertise, or integration requirements with existing software systems. For most practical purposes, either algorithm will deliver comparable forecasting accuracy.

In practical IoT-enabled smart grid environments, computational efficiency is a critical consideration because forecasting models may need to operate on distributed edge devices with limited hardware resources and strict latency constraints. During experimentation, LightGBM demonstrated the fastest training process and lowest computational overhead due to its histogram-based optimization and leaf-wise growth strategy. XGBoost also maintained stable computational performance with relatively low inference latency. In contrast, the LSTM model required substantially longer training times because of iterative sequence processing and backpropagation operations. These findings further support the suitability of gradient boosting models for real-time renewable energy forecasting applications in intelligent smart grid infrastructures.

In addition to predictive accuracy, computational efficiency was also observed during experimentation. LightGBM demonstrated the fastest training process among all models due to its leaf-wise growth strategy and histogram-based optimization. XGBoost required slightly longer training time but remained computationally efficient and stable. In contrast, the LSTM model required significantly longer training time because of its iterative backpropagation process and sequence-based computations. These findings indicate that gradient boosting methods are not only more accurate for this dataset but also more practical for real-time renewable energy forecasting applications.

The superior accuracy demonstrated in this study—with errors in the range of 2.5 to 3.7 units (MAE) and R^2 values exceeding 0.986—translates to meaningful economic benefits in operational contexts. Accurate forecasts enable more efficient scheduling of dispatchable generation assets, reduced procurement of expensive peaking power, better management of energy storage systems, and more competitive participation in electricity wholesale markets [46]; [47].

3.2.5 Limitations of the Study

Computational efficiency analysis further supports the superiority of the gradient boosting models for IoT and smart grid applications. During experimentation, LightGBM showed the fastest training time and lowest memory utilization due to its histogram-based optimization and leaf-wise growth mechanism. XGBoost required slightly higher computational resources but maintained stable inference performance with low latency. Conversely, the LSTM model demanded substantially longer training durations and greater memory allocation because of iterative sequence processing and backpropagation operations. These differences are particularly important for edge computing environments, where hardware resources, communication bandwidth, and energy consumption are often constrained.

Several limitations of this study should be acknowledged to contextualize the findings and guide interpretation. First, the LSTM model was implemented with a relatively standard architecture and limited hyperparameter optimization. A more exhaustive exploration of network architectures—including stacked LSTM layers, bidirectional configurations, attention mechanisms, dropout regularization, and systematic hyperparameter search—might yield substantially improved deep learning performance. The results therefore represent a comparison between well-optimized gradient boosting models and a baseline LSTM implementation rather than each method's theoretical maximum performance.

Second, the study used a single dataset from a specific geographic and temporal context. The relative performance of the three models may differ across datasets with different characteristics, including larger sample sizes, different climate zones, different temporal resolutions, or different mixes of renewable energy technologies. Generalization of these findings to other contexts should be undertaken cautiously.

Third, the evaluation was based solely on point forecasts using standard accuracy metrics (RMSE, MAE, R^2). Probabilistic forecasting capabilities—the ability to estimate prediction uncertainty intervals—were not assessed. In many operational contexts, uncertainty quantification is as valuable as point forecast accuracy, and the relative strengths of tree-based and deep learning models in probabilistic forecasting may differ from their point forecasting performance.

Fourth, external variables such as weather forecasts, satellite imagery, and atmospheric measurements were not included as input features. The incorporation of such exogenous data could potentially improve the performance of all models, particularly the LSTM, which might benefit from richer input representations.

3.2.6 Future Research Directions

Several promising avenues for future research emerge from this study's findings and limitations. The most immediate priority is a comprehensive optimization study of the LSTM architecture, employing systematic hyperparameter search methodologies such as Bayesian optimization or neural architecture search to identify configurations that maximize deep learning performance for renewable energy forecasting. Such a study should also investigate the impact of input sequence length, feature selection strategies, and data augmentation techniques on LSTM prediction accuracy.

The exploration of advanced deep learning architectures represents another high-priority direction. Transformer-based models, which have revolutionized natural language processing [48] and have shown promising results in time-series forecasting [49], offer an alternative sequential modeling approach that may overcome some of LSTM's limitations. Self-attention mechanisms in Transformers can capture long-range dependencies more effectively than recurrent architectures, potentially yielding improved performance on energy generation data with complex seasonal patterns. Temporal Convolutional Networks (TCNs) represent another noteworthy architecture that combines the parallelizable training of convolutional networks with the ability to process sequential data through dilated causal convolutions.

Hybrid and ensemble approaches that combine the strengths of tree-based and deep learning models offer a particularly intriguing research direction. A stacking ensemble, for example, could use gradient boosting models and neural networks as base learners, with a meta-learner that optimally combines their predictions. Such approaches can potentially achieve better performance than any individual model by exploiting the complementary strengths of different modeling paradigms—gradient boosting's effectiveness on tabular features and deep learning's ability to learn complex sequential representations.

The integration of exogenous weather data, satellite-derived irradiance measurements, and atmospheric pressure fields as additional input features could significantly enhance forecasting performance across all model types. Multi-modal learning approaches that combine tabular energy generation data with spatial weather imagery represent a frontier in renewable energy forecasting that leverages the unique capabilities of deep learning architectures.

Finally, the development of online learning and adaptive forecasting systems that continuously update model parameters as new data becomes available represents an important practical research direction. Such systems could adapt to gradual changes in generation patterns caused by equipment aging, grid topology changes, or long-term climate trends, maintaining forecasting accuracy over extended deployment periods.
Transfer learning approaches, where models pre-trained on large datasets from one geographic region or energy installation are fine-tuned for deployment at new sites with limited historical data, represent another avenue with significant practical value. This approach could accelerate the deployment of accurate forecasting systems for newly installed renewable energy facilities.

4. CONCLUSION

This study conducted a rigorous comparative evaluation of three machine learning and deep learning models—XGBoost, LightGBM, and LSTM—for forecasting solar photovoltaic output, wind power output, and total renewable energy production. The investigation employed identical data preprocessing, feature engineering, chronological data splitting, and evaluation protocols across all models to ensure a fair and meaningful comparison.
The results unequivocally demonstrate the superiority of gradient boosting methods for the renewable energy forecasting tasks examined in this study. LightGBM achieved the best performance for solar PV output (RMSE: 3.27, R^2: 0.987) and total renewable energy (RMSE: 4.66, R^2: 0.988), while XGBoost excelled in wind power output prediction (RMSE: 3.26, R^2: 0.987). Both tree-based models demonstrated remarkably consistent performance across all targets, with R^2 values exceeding 0.986 in every case. The LSTM network, in contrast, produced negative R^2 values across all targets (ranging from −0.024 to −0.083), indicating predictions consistently worse than a simple mean baseline.
The performance disparity between gradient boosting and deep learning approaches is attributed to several synergistic factors: the structured, tabular nature of the feature-engineered dataset aligns naturally with tree-based learning mechanisms; gradient boosting algorithms are inherently robust with moderate-sized datasets due to built-in regularization; and the extensive feature engineering pipeline effectively encodes temporal patterns into a format optimized for tabular learning. Conversely, the LSTM's underperformance likely stems from insufficient training data volume, suboptimal hyperparameter configuration, potential redundancy between engineered features and the network's internal temporal learning, and sensitivity to preprocessing choices.
These findings carry important practical implications. For operational renewable energy forecasting with structured datasets of moderate size, gradient boosting models—particularly LightGBM and XGBoost—represent the recommended approach, offering a compelling combination of prediction accuracy, computational efficiency, training speed, and model interpretability. The choice between LightGBM and XGBoost can be made based on secondary considerations, as both algorithms deliver comparable forecasting performance.
However, these conclusions should not be generalized to dismiss deep learning for all energy forecasting applications. With larger datasets, systematic hyperparameter optimization, advanced architectures (such as Transformers or attention-enhanced LSTMs), and appropriate feature engineering strategies, deep learning approaches have demonstrated competitive performance in numerous published studies(Hamdoun, Sagheer and Youness, 2021, p. 12496). Future research should prioritize comprehensive LSTM optimization, exploration of alternative deep learning architectures, hybrid ensemble methods, integration of exogenous weather data, and development of adaptive online learning systems to continue advancing the state of the art in renewable energy forecasting.
Additional diagnostic evaluation confirmed that the negative R^2 values produced by the LSTM model were not caused by preprocessing errors, tensor reshaping issues, or scaling inconsistencies. Instead, the findings indicate that the selected LSTM configuration was unable to generalize effectively on the structured tabular dataset used in this study. This observation reinforces the conclusion that model suitability is highly dependent on dataset characteristics, feature engineering strategies, and data volume.
Overall, the study confirms that gradient boosting approaches, particularly LightGBM and XGBoost, provide highly reliable and computationally efficient solutions for renewable energy forecasting in IoT-enabled smart grid systems. Their low latency, reduced memory requirements, and fast inference

capabilities make them highly suitable for deployment on distributed edge devices, IoT gateways, and intelligent network management platforms. These characteristics are essential for modern smart grids that require real-time energy monitoring, adaptive communication, and efficient decision-making across interconnected energy infrastructures.

ACKNOWLEDGMENTS

The authors would like to express their deepest gratitude to the Electronics Engineering Technology Study Program, Department of Information Technology, Politeknik Negeri Lampung, for providing the necessary laboratory facilities, technical resources, and academic environment that made this research possible. The authors also acknowledge the support of open-source communities and developers of scientific computing libraries, including Python, Scikit-learn, TensorFlow, XGBoost, LightGBM, and SHAP, which significantly facilitated the data analysis and modeling processes in this research. In addition, the authors are grateful to colleagues and reviewers for their constructive feedback and insightful suggestions that helped improve the quality of this manuscript. Finally, the authors appreciate the availability of renewable energy datasets and resources that enabled the successful implementation of this study on renewable energy forecasting and intelligent smart grid systems.

REFERENCES

[1] G. Yakoub, S. Mathew, and J. B. Leal, “Direct and indirect short-term aggregated turbine- and farm-level wind power forecasts integrating several NWP sources,” Heliyon, vol. 9, no. 11, Nov. 2023, Art. no. e21479, doi: 10.1016/j.heliyon.2023.e21479.

[2] N. E. Benti, M. D. Chaka, and A. G. Semie, “Forecasting renewable energy generation with machine learning and deep learning: Current advances and future prospects,” Sustainability, vol. 15, no. 9, p. 7087, May 2023, doi: 10.3390/su15097087.

[3] Y. Hu et al., “Temporal collaborative attention for wind power forecasting,” Appl. Energy, vol. 357, p. 122502, Mar. 2023, doi: 10.1016/j.apenergy.2023.122502.

[4] D. Pourkeramati et al., “Emerging paradigms in the energy sector: Forecasting and system control optimisation,” arXiv preprint arXiv:2507.12373, Jul. 2025, doi: 10.48550/arxiv.2507.12373.

[5] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[6] G. Ke et al., “LightGBM: A highly efficient gradient boosting decision tree,” Adv. Neural Inf. Process. Syst., vol. 30, pp. 3146–3154, 2017. [Online]. Available: https://hal.science/hal-03953007

[7] Y. Peng, S. Zhao, Z. Zeng, X. Hu, and Z. Yin, “LGBMDF: A cascade forest framework with LightGBM for predicting drug-target interactions,” Front. Microbiol., vol. 13, Jan. 2023, Art. no. 1092467, doi: 10.3389/fmicb.2022.1092467.

[8] Y. Zhang et al., “Comparison and explanation of forecasting algorithms for energy time series,” Mathematics, vol. 9, no. 21, p. 2794, Nov. 2021, doi: 10.3390/math9212794.

[9] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[10] J. Silva-Rodriguez, E. Raffoul, and X. Li, “LSTM-based net load forecasting for wind and solar power-equipped microgrids,” arXiv preprint arXiv:2408.00136, Aug. 2024, doi: 10.48550/arxiv.2408.00136.

[11] D. Venkatraman and V. Pitchaipillai, “Deep learning-based Auto-LSTM approach for renewable energy forecasting: A hybrid network model,” Trait. Signal, vol. 41, no. 1, pp. 525–530, Feb. 2024, doi: 10.18280/ts.410148.

[12] M. L. Hossain, S. M. N. Shams, and S. Ullah, “Time-series and deep learning approaches for renewable energy forecasting in Dhaka: A comparative study of ARIMA, SARIMA, and LSTM models,” Discov. Sustain., vol. 6, no. 1, 2025, doi: 10.1007/s43621-025-01733-5.

[13] L. Sua, H. Wang, and J. Huang, “Deep learning in renewable energy forecasting: A cross-dataset evaluation of temporal and spatial models,” arXiv preprint arXiv:2505.03109, May 2025, doi: 10.48550/arxiv.2505.03109.

[14] Y. Baghoussi, C. Soares, and J. Mendes-Moreira, “Corrector LSTM: Built-in training data correction for improved time-series forecasting,” Neural Comput. Appl., vol. 36, no. 26, pp. 16213–16231, Sep. 2024, doi: 10.1007/s00521-024-09962-x.
[15] J. Ren et al., “Deep learning and machine learning -- object detection and semantic segmentation: From theory to applications,” arXiv preprint arXiv:2410.15584, Oct. 2024, doi: 10.48550/arxiv.2410.15584.
[16] I. Nino-Adan, E. Portillo, I. Landa-Torres, and D. Manjarres, “Normalization influence on ANN-based models performance: A new proposal for features contribution analysis,” IEEE Access, vol. 9, pp. 125462–125477, 2021, doi: 10.1109/ACCESS.2021.3110647.
[17] A. Forootani, D. E. Aliabadi, and D. Thraen, “Climate aware deep neural networks (CADNN) for wind power simulation,” arXiv preprint arXiv:2412.12160, Dec. 2024, doi: 10.48550/arxiv.2412.12160.
[18] E. Regnani, “Machine learning meets mental training -- a proof of concept applied to memory sports,” arXiv preprint arXiv:2307.08712, Jul. 2023, doi: 10.48550/arxiv.2307.08712.
[19] J. Terven, D. Cordova-Esparza, J.-A. Romero-Gonzalez, A. Ramirez-Pedraza, and E. A. Chavez-Urbiola, “A comprehensive survey of loss functions and metrics in deep learning,” Artif. Intell. Rev., vol. 58, no. 7, 2025, doi: 10.1007/s10462-025-11198-7.
[20] S. Zaki, M. T. Benchekroun, M. Aboussaleh, B. Hezzem, and H. Laacha, “Prediction of raw mill fan vibrations based on machine learning models,” Research Square preprint, Jun. 2023, doi: 10.21203/rs.3.rs-3067230/v1.
[21] T. Chai and R. R. Draxler, “Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature,” Geosci. Model Dev., vol. 7, no. 3, pp. 1247–1250, May 2014, doi: 10.5194/gmd-7-1247-2014.
[22] M. R. R. Muhammed and C. Yinka-Banjo, “Machine-learning based prediction of generator runtime: A predictive maintenance solution for a telecommunications site,” Research Square preprint, Jan. 2024, doi: 10.21203/rs.3.rs-3936477/v1.
[23] X.-R. Ni, P. Pan, and J. Lin, “What makes a good BIM design: Quantitative linking between design behavior and quality,” arXiv preprint arXiv:2411.09481, Nov. 2024, doi: 10.48550/arxiv.2411.09481.
[24] D. Shao et al., “Uncertainty modeling method for wind and solar power output in building integrated energy systems under continuous anomalous weather,” arXiv preprint arXiv:2504.11100, Apr. 2025, doi: 10.48550/arxiv.2504.11100.
[25] S. Marton, S. Ludtke, C. Bartelt, and H. Stuckenschmidt, “GRANDE: Gradient-based decision tree ensembles for tabular data,” arXiv preprint arXiv:2309.17130, Sep. 2023, doi: 10.48550/arxiv.2309.17130.
[26] R. M. Jewel et al., “Comparative analysis of machine learning models for accurate retail sales demand forecasting,” J. Comput. Sci. Technol. Stud., vol. 6, no. 1, pp. 204–210, Jan. 2024, doi: 10.32996/jcsts.2024.6.1.23.
[27] A. Jolicoeur-Martineau, K. Fatras, and T. Kachman, “Generating and imputing tabular data via diffusion and flow-based gradient-boosted trees,” arXiv preprint arXiv:2309.09968, Sep. 2023, doi: 10.48550/arxiv.2309.09968.
[28] L. Ressel and H. A. A. Gardi, “Linear dimensionality reduction for word embeddings in tabular data classification,” arXiv preprint arXiv:2509.12346, Sep. 2025, doi: 10.48550/arxiv.2509.12346.
[29] R. D. A. Ferreira et al., “A pilot, predictive surveillance model in pharmacovigilance using machine learning approaches,” Adv. Ther., vol. 41, no. 6, pp. 2435–2445, Jun. 2024, doi: 10.1007/s12325-024-02870-5.
[30] L. Tani, D. A. J. Rand, C. Veelken, and M. Kadastik, “Evolutionary algorithms for hyperparameter optimization in machine learning for application in high energy physics,” Eur. Phys. J. C, vol. 81, no. 2, Feb. 2021, doi: 10.1140/epjc/s10052-021-08950-y.
[31] S. May, S. Hartmann, and F. Klawonn, “Combined pruning for nested cross-validation to accelerate automated hyperparameter optimization for embedded feature selection in high-dimensional data with very small sample sizes,” Research Square preprint, Sep. 2022, doi: 10.21203/rs.3.rs-2059858/v1.

[32] J. Jian et al., “Boosting tree-assisted multitask deep learning for small scientific datasets,” J. Chem. Inf. Model., vol. 60, no. 3, pp. 1235–1244, Mar. 2020, doi: 10.1021/acs.jcim.9b01184.
[33] Y. Guan and T. Plotz, “Ensembles of deep LSTM learners for activity recognition using wearables,” Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., vol. 1, no. 2, pp. 1–28, Jun. 2017, doi: 10.1145/3090076.
[34] E. Jafarigol and T. B. Trafalis, “The paradox of noise: An empirical study of noise-infusion mechanisms to improve generalization, stability, and privacy in federated learning,” Research Square preprint, Oct. 2023, doi: 10.21203/rs.3.rs-3492014/v1.
[35] K. Benidis et al., “Deep learning for time series forecasting: Tutorial and literature survey,” ACM Comput. Surv., vol. 55, no. 6, pp. 1–36, Dec. 2022, doi: 10.1145/3533382.
[36] N. Luaran and R. Alfred, “Assessment of the optimization of hyperparameters in deep LSTM for time series sea water tidal shift,” Research Square preprint, May 2022, doi: 10.21203/rs.3.rs-1669035/v1.
[37] R. Marco, S. Sakinah, S. Ahmad, and S. Ahmad, “An improving long short term memory-grid search based deep learning neural network for software effort estimation,” Int. J. Intell. Eng. Syst., vol. 16, no. 4, pp. 164–180, Aug. 2023, doi: 10.22266/ijies2023.0831.14.
[38] D. T. Tran, A. Iosifidis, J. Kanniainen, and M. Gabbouj, “Temporal attention-augmented bilinear network for financial time-series data analysis,” IEEE Trans. Neural Networks Learn. Syst., vol. 30, no. 5, pp. 1407–1418, May 2018, doi: 10.1109/TNNLS.2018.2869225.
[39] M. Sarkar and A. De Bruyn, “LSTM response models for direct marketing analytics: Replacing feature engineering with deep learning,” SSRN Electronic Journal, May 2020, doi: 10.2139/ssrn.3601025.
[40] M. Chaudhary, M. S. Adams, S. Mukhopadhyay, M. Litoiu, and L. E. Sergio, “Sabotage detection using DL models on EEG data from a cognitive-motor integration task,” Front. Hum. Neurosci., vol. 15, Jun. 2021, doi: 10.3389/fnhum.2021.662875.
[41] A. L. Mur, “Single-pixel imaging: Compressed video acquisition and reconstruction using deep learning,” Ph.D. dissertation, Universite de Lyon, France, 2021. [Online]. Available: https://theses.hal.science/tel-03670825
[42] O. Adegoke, “Predicting the unpredictable: Reproducible BiLSTM forecasting of incident counts in the global terrorism database (gtd),” arXiv preprint arXiv:2510.15136, Oct. 2025, doi: 10.48550/arxiv.2510.15136.
[43] X. Wang, H. Wang, B. Bhandari, and L. Cheng, “AI-empowered methods for smart energy consumption: A review of load forecasting, anomaly detection and demand response,” Int. J. Precis. Eng. Manuf. Technol., vol. 11, no. 3, pp. 963–993, Jun. 2023, doi: 10.1007/s40684-023-00537-0.
[44] C. Song, S. Bouguerra, E. Krogen, and D. J. Abadi, “Evaluating learned query performance prediction models at LinkedIn: Challenges, opportunities, and findings,” arXiv preprint arXiv:2504.17181, Apr. 2025, doi: 10.48550/arxiv.2504.17181.
[45] J. Huang, G. Li, Z. Yan, F. Luo, and S. Li, “Joint learning of interpretation and distillation,” arXiv preprint arXiv:2005.11638, May 2020. [Online]. Available: http://arxiv.org/abs/2005.11638
[46] R. Widiss and K. M. Porter, “Review of variable generation forecasting in the West: July 2013 - March 2014,” National Renewable Energy Laboratory (NREL), Golden, CO, USA, Tech. Rep. NREL/TP-5500-61111, 2014, doi: 10.2172/1126838.
[47] S. Goodarzi, H. N. Perera, and D. W. Bunn, “The impact of renewable energy forecast errors on imbalance volumes and electricity spot prices,” Energy Policy, vol. 134, p. 110827, Nov. 2019, doi: 10.1016/j.enpol.2019.06.035.
[48] M. Arsalan, “Transformers in natural language processing: A comprehensive review,” Int. J. Res. Appl. Sci. Eng. Technol., vol. 12, no. 5, pp. 5591–5597, May 2024, doi: 10.22214/ijraset.2024.62863.
[49] L. Su, X. Zuo, R. Li, X. Wang, H. Zhao, and B. Huang, “A systematic review for transformer-based long-term series forecasting,” Artif. Intell. Rev., vol. 58, no. 3, 2025, doi: 10.1007/s10462-024-11044-2.