Journal of Advanced Digital Communications
https://www.sciltp.com/journals/jadc

Article

Ensemble Technique-Based Short-Term Supply and Demand Forecasting with Features Selection Approach in Decentralized Energy Systems

Adugna Gebrie Jember [1], Ruiyu Bao [1], Zijia Yao [1], Zhao Wang [1,*], Zhenyu Zhou [1] and Xiaoyan Wang [2]

1 State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources, North China Electric Power University, Beijing 102206, China
2 Graduate School of Science and Engineering, Ibaraki University, Mito 310-8512, Japan
* Correspondence: zhao_w@ncepu.edu.cn

How To Cite: Jember, A.G.; Bao, R.; Yao, Z.; et al. Ensemble Technique-Based Short-Term Supply and Demand Forecasting with Features Selection Approach in Decentralized Energy Systems. *Journal of Advanced Digital Communications* **2024**, *1*(1), 5. https://doi.org/10.53941/jadc.2024.100005.

Received: 21 August 2024
Revised: 2 December 2024
Accepted: 2 December 2024
Published: 31 December 2024

Abstract: Decentralized energy systems (DESs) present a paradigm shift toward more sustainable and resilient electricity networks with the increasing integration of renewable energy sources. Accurately forecasting energy supply and electricity demand is crucial for the efficient operation of DES. This paper focuses on accurate short-term energy supply and demand forecasting using machine learning (ML) algorithms across different seasons within small-scale DES, including photovoltaic (PV) generation, wind generation, and load demand. Initially, we develop multiple base model ML algorithms including long short-term memory (LSTM), convolutional neural networks (CNN), eXtreme gradient boosting (XGBoost), and recurrent neural networks (RNN) with feature selection approaches to improve forecasting accuracy and reduce model complexity. These base models leverage key temporal and spatial features and seasonal variations process to improve the model forecasting accuracy and reduce overfitting across different seasons. To further reduce forecast errors, we propose robust ensemble forecasting techniques including simple averaging (SA), weighted averaging (WA), and Stacking. In ML algorithms, the ensemble forecasting technique combines the multiple base models' forecasts to produce a more accurate and robust final forecast by leveraging the strengths and compensating for the weaknesses of individual base models. Finally, numerical simulations are conducted using Python, Keras, and TensorFlow libraries to develop, train, evaluate, and validate the effectiveness and accuracy of the developed forecasting models and the proposed ensemble forecasting techniques. The results demonstrate that the proposed approach offers a robust solution for short-term supply and demand forecasting problems in DES. The work is both novel and effective from the perspectives of application, ML algorithms combination, and performance improvement.

Keywords: forecasting; decentralized energy systems; PV; wind generation; load demand; machine learning; feature selection; ensemble techniques

1. Introduction

Nowadays, the evolution of the energy system towards decentralized energy systems (DESs) marks a significant shift from traditional and centralized networks to distributed networks. This shift is primarily driven by the increasing incorporation of renewable energy sources, advancements in energy storage and management technologies, and a growing emphasis on sustainability and energy efficiency. DES has several advantages including improving energy security, reducing transmission losses and carbon emissions by generating energy closer to consumption points, and promoting environmental sustainability [1–6]. The growing adoption of renewable energy sources such as photovoltaic (PV) and wind power, necessitates accurate forecasting to ensure system reliability and efficiency in DES. Accurately forecasting supply and demand is critical for balancing supply and demand, reducing operational costs, and improving the stability of DES. However, there remains a significant gap between the existing forecasting methodologies and the requirements of DES [7,8].

Currently, machine learning (ML) and deep learning algorithms have emerged as powerful tools for addressing the key challenges in accurate supply and demand forecasting, providing effective solutions by analyzing large amounts of historical data to forecast future outcomes due to their ability to capture complex patterns. This capability is crucial for high-resolution temporal and spatial forecasting accuracy to effectively manage localized generation and consumption [9,10]. Multiple ML algorithms, including long short-term memory (LSTM) networks, convolutional neural networks (CNN), eXtreme gradient boosting (XGBoost), and recurrent neural networks (RNN), are used to forecast PV, wind generation, and load demand [11–15]. The use of a single ML for forecasting tasks in various domains, such as solar radiation and wind speed prediction, has long presented challenges due to the inherent complexity and variability. However, recent studies have demonstrated the effectiveness of hybrid and ensemble learning models in improving forecasting accuracy. These models combine the strengths of multiple ML algorithms, allowing them to better capture the underlying patterns and reduce forecast errors across different applications [16,17].

This paper aims to develop multiple base models for accurate short-term supply and demand forecasting that incorporates feature selection across different seasons. Feature selection is important for effective base model development by precisely identifying the most relevant factors to improve forecasting accuracy. The strengths and limitations of ML models are analyzed to develop multiple base models for forecasting. LSTM is well-suited for capturing long-term temporal dependencies in time series data but struggles to identify spatial patterns and can be computationally intensive. CNN is effective at identifying spatial patterns in data but cannot capture temporal dependencies as effectively as LSTM. XGBoost is known for its high performance in various applications but can struggle with temporal dependencies without additional feature engineering. RNN is good at capturing temporal patterns but can suffer from vanishing gradient problems and may not perform well on longer sequences [15,18–21]. Therefore, the performance of base models varies with their application and suitability for different types of time series data and forecasting tasks.

To address the limitations of a single base model, this paper further proposes and implements ensemble techniques. Ensemble forecasting techniques, such as simple averaging (SA), weighted averaging (WA), and Stacking, are used to combine multiple base models to produce more accurate and robust final forecast. These techniques take advantage of the strengths of each base model and compensate for their weaknesses, leading to better performance in terms of accuracy, stability, and generalization. Despite the potential of these approaches, achieving high and optimal forecasting accuracy for energy supply and load demand forecasting with feature selection across different seasons still faces the following challenges.

(1) Base model development: Developing suitable base models requires selecting the most appropriate ML algorithms and fine-tuning them for specific forecasting tasks. Identifying the most relevant factors for accurate forecasting is complex due to the diverse and dynamic nature of the data.
(2) Feature Selection: Inappropriate feature selection can lead to poor model performance and increased forecast errors. By selecting the most relevant features, base models are less likely to be distracted by noise or irrelevant data. Creating new temporal features from existing data can provide more information to the base model. For instance, time-based features such as hour of the day, day of the week, and season can capture the impact of different seasons on supply (PV and wind generation) and demand (consumption). Incorporating lagged variables for time series data can also improve the accuracy of the base model.
(3) Accuracy improvement and overfitting and forecast error reduction: Improving the forecasting accuracy of multiple base models while reducing overfitting across different seasons is a significant challenge. Individual base models often show varying degrees of accuracy and reliability due to differences in their ability to handle specific forecasting tasks. This variability necessitates the development of more robust forecasting techniques to improve overall performance and ensure consistent and reliable forecasting.

To address the aforementioned challenges, we develop multiple base model algorithms, including LSTM, CNN, XGBoost, and RNN with a feature selection approach for accurate short-term supply and demand forecasting across different seasons. While these base models are indeed widely used in forecasting tasks, each base model algorithm comes with distinct forecasting strengths and limitations. Therefore, we further propose robust ensemble forecasting techniques including SA, WA, and Stacking that strategically combine the forecasts of these base models. These ensemble forecasting techniques leverage the complementary strengths of individual base models while mitigating their weaknesses to improve the forecasting accuracy and reduce overfitting and forecast errors. The main contributions of this paper are as follows:

(1) System model design and implementation procedures: We design a comprehensive system model along with detailed implementation procedures of accurate short-term supply and demand forecasting. This involves integrating advanced data analytics and forecasting algorithms to improve forecasting accuracy.

(2) Multiple base model algorithms development with feature selection approach: We develop multiple base model algorithms with feature selection to improve the forecasting accuracy of PV, wind generation and load demand across different seasons. Feature selection can optimize the input variables for the base model algorithms, leading to more accurate and efficient forecasts.

(3) Robust ensemble techniques proposition: To address the challenges of overfitting and forecast errors, we further propose robust ensemble forecasting techniques including SA, WA, and Stacking. These techniques can significantly reduce forecast errors and overfitting, which leads to improved forecasting accuracy and overall model performance. These techniques integrate and leverage the strengths and mitigate the weaknesses of the individual base models, and optimally combine their forecast outputs using a training process. They are designed to handle both linear, non-linear, and complex patterns in the time series data.

(4) Model performance evaluation: We evaluate the performance of multiple base models and robust ensemble forecasting techniques using various performance metrics, including mean absolute error (MAE), mean absolute percentage error (MAPE), sum of squared errors (SSE), root mean square error (RMSE), standard deviation of error (SDE), normalized mean absolute error (NMAE), and the R-squared (R^2) score.

2. Related Works

To address the aforementioned challenges, it is essential to review relevant and recent existing research work and solutions. The integration of renewable energy sources and the growing participation of prosumers have emphasized the need for accurate supply and demand forecasting methods. Several studies have adopted different ML algorithms for PV, wind generation, and load demand forecasting. Specifically, hybrid models that combine the strengths of different ML algorithms have been implemented to improve forecasting accuracy. In [22], Baul et al. developed a hybrid short-term load forecasting model that combines CNN and gated recurrent unit (GRU). The model identifies key features using a Pearson correlation matrix. The proposed CNN-GRU model was evaluated against other models like LSTM and Transformer, demonstrating that the hybrid model achieves greater forecasting accuracy. In [23], Wang et al. introduced a stacking-based method for short-term wind power forecasting, which employed a combination of different ML models under a stacking framework, such as CBLSTM and XGBoost. The strengths of different algorithms are leveraged to improve forecasting accuracy and stability. In [24], Trivedi et al. proposed a comprehensive data-driven approach for short-term forecasting of PV generation and load demand, which considered feature selection using the random forest-based sequential forward selection algorithm and tree-structured parzen estimator. This method demonstrated substantial improvements in forecasting accuracy by comparing traditional models like ARIMA with advanced ML models such as LSTM, GRU, and CNN. In [25], Moreno et al. proposed a multi-step wind speed forecasting model that integrated multi-stage decomposition including variational mode decomposition (VMD) and singular spectrum analysis (SSA) as preprocessing steps to preprocess raw wind speed time series data. By applying the hybrid VMD-SSA model in combination with LSTM networks, notable improvements were achieved in multi-step-ahead wind speed forecasting accuracy.

In [26], Cao et al. proposed a model for wind power forecasting that integrated a secondary-weighted attention mechanism with LSTM (STAM-LSTM). Feature selection was performed using a random forest algorithm. The hybrid STAM-LSTM model improved forecasting accuracy and stability compared to individual models. In [20], Agga et al. developed a hybrid short-term load forecasting model that combines CNN and LSTM networks. They designed the model to handle both linear and nonlinear patterns in time-series data. This hybrid approach, combining the strengths of CNN and LSTM, demonstrated significant improvements compared to individual models. In [21], Phan et al. developed a short-term PV power forecasting model that integrates kernel principal component analysis (KPCA) with XGBoost. KPCA was utilized to reduce the dimensionality and extract key features from the dataset. The XGBoost algorithm was used to forecast PV power generation due to its robustness and accuracy in handling large datasets. Similarly, in [27], Kumari et al. proposed an ensemble learning method using XGBoost and deep neural networks to forecast hourly solar irradiance. The method highlights the potential of ensemble learning for solar forecasting challenges.

Indeed, recent research methods have successfully employed various ML techniques to improve the forecasting accuracy of PV, wind generation, and load demand. However, these approaches often focus on specific types of energy sources and lack comprehensive integration of multiple base models with appropriate feature selection approaches and model combinations. Our work addresses these limitations by applying these comprehensive integration techniques to short-term supply and demand forecasting within DES. Uniquely, our model integrates multiple ML model algorithms with feature selection including LSTM, CNN, XGBoost, and RNN to improve base model forecasting accuracy and reduce model complexity. While these base models are widely used in forecasting tasks, each base model algorithm has different forecast strengths and limitations. Therefore, in order to overcome the limitations of multiple base models, we further propose and implement ensemble forecasting techniques, including SA, WA, and Stacking. These forecasting techniques optimally combine the forecasting outputs of base models, improving the overall forecasting accuracy and reducing forecast errors and overfitting. This paper aims to provide a robust solution for short-term supply and demand forecasting problems by leveraging the complementary strengths of individual base modes while mitigating their weaknesses using a training process.

3. System Model Design and Implementation Procedures

In this section, we design and define the system model and implementation procedures for supply and demand forecasting. The system model and implementation procedures for short-term supply and demand forecasting in DES involve several key steps to effectively develop and leverage multiple base model algorithms and ensemble techniques with feature selection.

(1) Data collection and preprocessing: Data collection determines the quality and quantity of the data that directly impacts the model's performance and accuracy. Collect historical data of electricity consumption, generation fluctuations of PV and wind power, weather data, and other relevant variables as shown in Figure 1. Data preprocessing is crucial to prepare the raw data for base model training. It contains data cleaning, transforming, and organizing. Key steps in data preprocessing include handling missing values and ensuring data consistency. Data normalization is also essential to ensure that all features contribute equally to the base model training process.

(2) Feature selection: Identify and select the most relevant features for the base model using correlation analysis and domain knowledge techniques. Create new temporal features from existing data to provide more information to the base model, such as creating time-based features like hour of the day, day of the week, seasons to capture the impact of different seasons on energy generation and consumption, and lagged variables for time series data.

(3) Base model development: Developing a base model involves selecting an appropriate ML algorithm and defining the model architecture.

(4) Model training: After defining the base model architecture, the next step is to train the base models using the training data set. It includes feeding the data into the base models, adjusting weights using backpropagation, and optimizing the loss function. First, split the data into training, validation, and test sets. Then, train each base model based on the training set using appropriate loss functions and optimizers.

(5) Model validation: Model validation is a critical step in each base model workflow that ensures the developed model performs well on unseen data. It involves evaluating the base model based on validation data set to tune hyperparameters and prevent overfitting.

(6) Model evaluation: After training the base models, it is essential to evaluate their performance based on the test set. This step includes making forecasts and comparing them to the actual values and calculating evaluation metrics using metrics such as MAE, MAPE, SSE, RMSE, SDE, NMAE and R^2 score.

(7) Ensemble techniques: We propose robust ensemble forecasting techniques algorithms (SA, WA, Stacking) that are used to combine the forecasts of multiple base models to produce a more accurate and optimal final forecast. After developing and validating a base model, ensemble techniques leverage the strengths and compensate for the weaknesses of individual base models, leading to better performance.

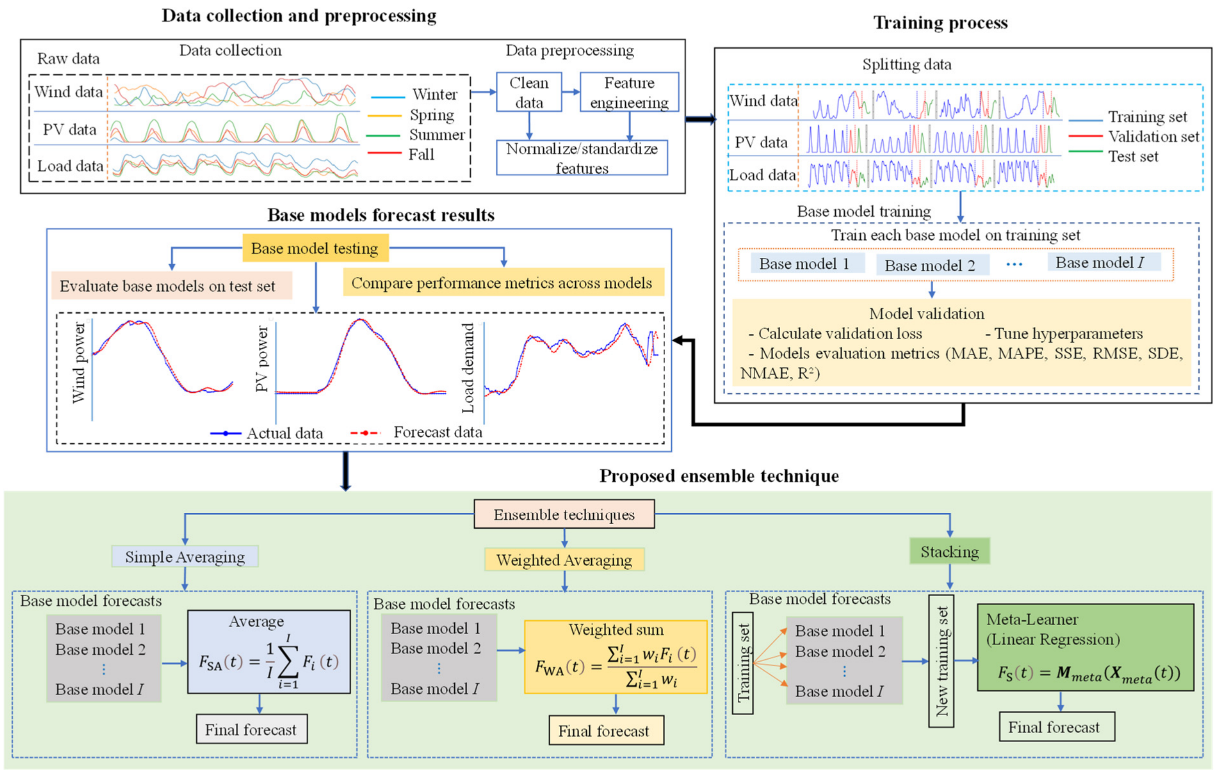

Figure 1. System model and implementation structure for short-term supply and demand forecasting.

4. Multiple Base Model Development and Formulation

In this section, we develop multiple base models with feature selection for time-series supply and demand data forecasting. The process includes model selection, model design, training, and evaluation. The goal is to create comprehensive base models that capture the spatial and temporal dependencies and patterns necessary for accurate forecasting. For all base models, all historical data and features are normalized using MinMaxScaler and sequence to ensure that they contribute equally to the model's learning process. It is crucial to ensure the data are clean, scaled, and properly formatted for the chosen model. Given an original value α , the scaled value α' is computed as

$$\alpha' = \left(\frac{\alpha - \alpha_{\min}}{\alpha_{\max} - \alpha_{\min}} \right) \tag{1}$$

where $\alpha_{\max}$ and $\alpha_{\min}$ are the maximum and minimum values in the dataset. The data are then transformed into sequences suitable for each base model, ensuring that the temporal dependencies, spatial, trends, and patterns are preserved and can be learned by each base model for accurate forecasting.

4.1. LSTM Model

The LSTM forecasting model is designed to capture long-term temporal dependencies in sequential data. It is mostly well-suited for time series forecasting due to the ability to retain information over long periods. LSTM networks consist of units called cells. Each LSTM cell has three main components, i.e., the input gate, the forget gate, and the output gate. These gates regulate the flow of information and control what is stored in the cell state (memory) [28–30]. The LSTM cell is governed as

$$IG_t = \sigma\left(W_{IG} \cdot \left[HS_{t-1}, X_t\right] + B_{IG}\right) \tag{2}$$

$$FG_t = \sigma\left(W_{FG} \cdot \left[HS_{t-1}, X_t\right] + B_{FG}\right) \tag{3}$$

$$CS_t = FG_t \odot CS_t + IG_t \odot CC_t \tag{4}$$

$$CC_t = \tanh\left(W_{CS} \cdot \left[HS_{t-1}, X_t\right] + B_{CS}\right) \tag{5}$$

$$OG_t = \sigma\left(W_{OG} \cdot \left[HS_{t-1}, X_t\right] + B_{OG}\right) \tag{6}$$

$$HS_t = OG_t \odot tanh(CS_t) \tag{7}$$

where $t = 1:T$, t is the time step index, and T is the total number of time steps in the observed data points. X_t represents the current input data at the time step t. IG_t, FG_t, and OG_t are the control gates of input, forget, and output at time step t. W_{IG}, W_{FG}, W_{OG}, and W_{CS} are the weight matrices for each gate and cell state. B_{IG}, B_{FG}, B_{CS}, and B_{OG} are the bias terms. CS_t, CC_t, and HS_t are the cell, candidate, and hidden states. HS_{t-1} is the hidden state at time $t-1$. σ and $\tanh$ are the sigmoid and hyperbolic tangent activation functions in the network. $\odot$ is element-wise multiplication.

4.2. CNN Model

CNN forecasting model is a type of deep neural network primarily used for image recognition and processing. It can also be used for time series forecasting by capturing local patterns and correlations in sequential data. It adaptively learns spatial hierarchies of features from input time series data. Through the use of filters/kernels, CNN mode can efficiently process data with high dimensionality, making it highly effective for tasks involving multi-dimensional time series data. The overall operation of a CNN model for time series forecasting can be described through the convolutional operations followed by activation functions, pooling, and fully connected layers to extract features from the input sequence [29,31,32]. The model begins with convolutional layers, where each layer applies convolution operation, and is given by

$$h_t^{(l)} = \text{ReLU}\left(\sum_{k=1}^{K} (W_k^{(l)} . x_{t,k}^{(l-1)}) + b^{(l)}\right) \tag{8}$$

where $h_t^{(l)}$ is the output of the l-th convolutional layer at time step t. K is the kernel size. ReLU is the activation function. $W_k^{(l)}$ represents the weight of the k-th filter in the layer l. $x_{t,k}^{(l-1)}$ is the input data at time step t from the previous layer. $b^{(l)}$ is the bias term for the l-th layer. The pooling layer reduces the spatial dimensionality of the data, allowing the model to focus on the most significant features. The pooling operation is given by

$$h_t^{(l,\text{pool})} = \text{MaxPool}\left(h_{t:t+p}^{(l)}\right) \tag{9}$$

where $h_t^{(l,\text{pool})}$ is the output of the pooling layer at time step t. MaxPool is the max pooling function. p is the pooling size. After the convolution and pooling layers, the outcome feature maps are flattened into a 1D array. The flattening operation is given by

$$\text{flattened} = \text{flatten}\left(h_t^{(L,\text{pool})}\right) \tag{10}$$

where flattened is the flattened 1D array and $h_t^{(L,\text{pool})}$ is the output of the last pooling layer L. The flattened output is then fed into a fully connected layer, transforming the features into a higher-level representation. The output of the fully connected layer is calculated as

$$h_t^{(fc,m)} = \text{ReLU}\left(W^{(fc,m)} \cdot h_t^{(fc,m-1)} + b^{(fc,m)}\right) \tag{11}$$

where $h_t^{(fc,m)}$ is the output of the m-th fully connected layer. $W^{(fc,m)}$ is the weight matrix of the m-th fully connected layer. $h_t^{(fc,m-1)}$ is the input from the previous fully connected layer (or the flattened input if $m=1$). $b^{(fc,m)}$ is the bias term of the m-th fully connected layer. Finally, the output layer generates the final fully connected layer forecast value, which is given by

$$OL_t = W^{(output)} \cdot h_t^{(fc,M)} + b^{(output)} \tag{12}$$

where OL_t is the final fully connected layer output value. $W^{(output)}$ is the weight matrix of the output layer. $h_t^{(fc,M)}$ is the output from the last fully connected layer M. $b^{(output)}$ is the bias term of the output layer.

4.3. XGBoost Model

The XGBoost forecasting model is a powerful ML technique for regression and classification tasks. It is an efficient and scalable implementation of gradient boosting for supervised learning problems, designed to optimize computational speed and model performance. For the time-series supply and demand data forecasting, the process involves several key steps, including defining the objective function with the associated loss function and calculating and applying regularization terms to prevent overfitting. These steps collectively ensure that the XGBoost model achieves high accuracy while maintaining generalizability [33,34]. The first step is to define the objective function, which provides the framework for what the model aims to minimize. The objective function in XGBoost model integrates both the loss function and a regularization term, which is given by

$$\Theta = \sum_{n=1}^{N} loss(A_n, F_n) + \sum_{y=1}^{Y} \Omega(f_y), \Theta = (f_1, f_2, \ldots, f_Y) \tag{13}$$

where Θ denotes the model parameters, consisting of multiple trees. N is the total number of observations (data points) in the dataset for which the forecast is done. A_n and F_n are the actual and forecast values for the n-th data point. Y is the total number of trees. $\Omega(f_y)$ is the regularization term for the y-th tree. For regression tasks, the loss function is given by

$$loss(A_n, F_n) = \frac{1}{2}(A_n - F_n)^2 \tag{14}$$

Once the objective function is defined, the next step is to calculate the regularization terms. During the model training process, regularization terms are applied to prevent overfitting by penalizing the complexity of the model. For each tree f_y, the regularization term is given by

$$\Omega(f_y) = \gamma Z_y + \frac{1}{2}\lambda \sum_{z=1}^{Z_y} w_z^2 \tag{15}$$

where γ is a regularization parameter controlling the number of leaves in the trees. Z_y is the total number of leaves in the y-th decision tree, which are the points where the tree makes a forecast. λ is the L2 regularization term on leaf weights. L2 regularization helps to prevent overfitting by penalizing large weights and encouraging the model to keep the weights small. w_z is the weight of the leaf z. Putting it all together, the objective function for the XGBoost model is given by

$$\Theta = \sum_{n=1}^{N} \frac{1}{2}(A_n - F_n)^2 + \sum_{y=1}^{Y}\left(\gamma Z_y + \frac{1}{2}\lambda \sum_{z=1}^{Z_y} w_z^2\right) \tag{16}$$

4.4. RNN Model

RNN forecasting model is a type of artificial neural network particularly well-suited for handling time-series data forecasting. The model architecture includes a single RNN layer followed by a linear layer to produce the final output. The training process involves batch processing, optimization, and evaluation [35]. The RNN model implements the hidden state and output components. The basic RNN does not have gate components, and simply updates its hidden state based on the current input and the previous hidden state, which is given by

$$HS_t = \sigma(W_{HS} \cdot X_t + W_{HS} \cdot HS_{t-1} + B_{HS}) \tag{17}$$

where $(\cdot)$ is the dot product. W_{HS} is the weight matrix for the input to the hidden state and the hidden state to the hidden state connections, B_{HS} is the bias vector for the hidden state. The output O_t at time step t is calculated from the hidden state, which is given by

$$O_t = W_O \cdot HS_t + B_O \tag{18}$$

where W_O is the weight matrix for the hidden-to-output connections, and B_O is the bias vector for the output layer.

4.5. Performance Evaluation Metrics for Base Models

In the context of ML models, evaluation metrics are measures used to evaluate how well a model performs. After training, the base model's performance is evaluated using metrics including MAE, MAPE, SSE, RMSE, SDE, NMAE, and R^2, which provides information on the accuracy and consistency. The evaluation metrics are formulated as

$$\mathrm{MAE} = \frac{1}{N}\sum_{n=1}^{N}|A_n - F_n| \tag{19}$$

$$\mathrm{MAPE} = \frac{100\%}{N}\sum_{n=1}^{N}\left|\frac{A_n - F_n}{A_n}\right| \tag{20}$$

$$\mathrm{SSE} = \sum_{n=1}^{N}(A_n - F_n)^2 \tag{21}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(A_n - F_n)^2} \tag{22}$$

$$\mathrm{SDE} = \frac{1}{N}\sum_{n=1}^{N}\left((A_n - F_n) - \overline{(A_n - F_n)}\right)^2 \tag{23}$$

$$\mathrm{NMAE} = \frac{\frac{1}{N}\sum_{n=1}^{N}|A_n - F_n|}{N(\max(A_n) - \min(A_n))} \tag{24}$$

$$R^2 = \frac{\sum_{n=1}^{N}(A_n - F_n)^2}{\sum_{n=1}^{N}(A_n - \overline{A_n})^2} \tag{25}$$

where $\overline{(A_n - F_n)}$ is the mean of the errors, i.e., the average difference between actual and forecast values. $\max(A_n)$ and $\min(A_n)$ are the maximum and minimum actual values in the n-th data points, and $\overline{A_n}$ is the mean of the actual values. The training process for each base model involves using the mean squared error (MSE) loss function for regression problems, which is given by

$$L_{\mathrm{base}} = \mathrm{MSE}(F_n, A_n) = \frac{1}{N}\sum_{n=1}^{N}(A_n - F_n)^2 \tag{26}$$

where L_{base} represents the base model loss function. The training process for each base model aims to minimize the loss function by using backpropagation and gradient descent. The base model parameters are updated as

$$\theta_{\mathrm{base}}(t) = \theta_{\mathrm{base}}(t-1) - \eta\frac{\partial L_{\mathrm{base}}}{\partial\theta_{\mathrm{base}}(t-1)} \tag{27}$$

where $\theta_{\mathrm{base}}(t)$ are the base model parameters at the time step t, and η is the learning rate. $\frac{\partial L_{\mathrm{base}}}{\partial\theta_{\mathrm{base}}(t-1)}$ is the gradient of the loss function with respect to the parameters at the time step $t-1$.

5. Robust Ensemble Forecasting Techniques

In this section, we propose a robust ensemble forecasting techniques algorithm, which combines the forecasts from multiple base models to produce a more accurate and robust final forecast. The main idea is that multiple models working together can outperform a single base model in terms of accuracy, stability, and generalization.

After developing and validating multiple base models, ensemble techniques can improve forecast accuracy that leverages the strengths and compensates for the weaknesses of individual base models [36]. This approach reduces the overfitting risks and forecast errors, as well as improves generalization to unseen data, leading to more stable and optimal forecasts. The ensemble mitigates the impact of any single base model's poor performance. In this section, we implement three types of ensemble forecasting techniques including SA, WA, and Stacking.

5.1. Simple Averaging (SA) Technique

SA is a straightforward ensemble technique where the final forecast is the arithmetic mean of the forecasts from all base models, which is given by

$$F_{\mathrm{SA}}(t)=\frac{1}{I}\sum_{i=1}^{I}F_i(t) \tag{28}$$

where $F_{\mathrm{SA}}(t)$ is the final SA forecast value at the time step t, I is the number of base models, and $F_i(t)$ is the forecast value from the i-th base model at the time step t.

5.2. Weighted Averaging (WA) Technique

The WA technique assigns different weights to the forecasts of each base model based on their performance. Models with better performance are given higher weights reflecting their relative importance or accuracy. The proposed WA forecast can be formulated as

$$F_{\mathrm{WA}}(t)=\frac{\sum_{i=1}^{I}w_iF_i(t)}{\sum_{i=1}^{I}w_i} \tag{29}$$

$$w_i=\frac{1}{\mathrm{RSME}_i} \tag{30}$$

where $F_{\mathrm{WA}}(t)$ is the final WA forecast value at time step t and w_i is the weight assigned to the i-th base model, which is calculated based on the inverse of the RMSE of each base model.

5.3. Stacking Technique

The Stacking technique is an advanced ensemble method that trains a meta-model to combine the forecasts of multiple base models [37]. This meta-model learns the optimal way to integrate the base model forecasts, leading to superior technique performance by capturing complex patterns among the forecasts of base models. The proposed Stacking technique is formulated as

$$F_{\mathrm{S}}(t)=M_{\mathrm{meta}}(x_{\mathrm{meta}}(t)) \tag{31}$$

where $F_S(t)$ is the final stacked forecast value at time step t, M_{meta} is a Meta-model (linear regression) used to combine base model forecasts, $X_{\mathrm{meta}}(t)$ is the meta-features matrix constructed from the forecasts of the base models at time step t. The meta-features matrix X_{meta} is defined as

$$X_{\mathrm{meta}}=\begin{bmatrix} F_{1,1} & F_{2,1} & \cdots & F_{I,1} \\ F_{1,2} & F_{2,2} & \cdots & F_{I,2} \\ \vdots & \vdots & \ddots & \vdots \\ F_{1,J} & F_{2,J} & \cdots & F_{I,J} \end{bmatrix} \tag{32}$$

where $F_{i,j}$ is the forecast value of the i-th base model for the j-th sample, J is the number of samples in the dataset used to train the meta-model. The meta-features matrix X_{meta} is constructed by combining the forecasts from each base model, and serves as the inputs to train the meta-model. The loss function measures the error between the meta-model's forecast and the actual target value of the j-th sample, which is formulated as

$$L_{\text{meta}}(M(x_{\text{meta},j}), A_j) = (M(x_{\text{meta},j}) - A_j)^2 \tag{33}$$

where $x_{\text{meta},j}$ is the j-th row of the meta-features matrix X_{meta}, A_j is the actual target value for the j-th sample, L_{meta} is the loss function for meta-model regression, M is the meta-model parameters being optimized. Then, the meta-model M_{meta} is trained to minimize the loss function over the training dataset, which is formulated as

$$M_{\text{meta}} = \arg\min_{M} \sum_{j=1}^{J} L_{\text{meta}}(M(x_{\text{meta},j}), A_j) \tag{34}$$

Each term $M(x_{\text{meta},j})$ represents the forecast of the meta-model for the j-th sample based on the j-th row of the meta-features matrix.

6. Simulation Results and Validation

In this section, we discuss the numerical simulation study conducted on the proposed multiple base models and robust ensemble forecasting techniques, along with the obtained results. This includes comparative validation and evaluation of both the individual base models and the proposed ensemble forecasting techniques simulation results for energy supply and electricity demand forecasts across different seasons. Additionally, we present a quantitative relevance analysis and result comparisons with different performance evaluation metrics obtained from the multiple base models and the robust ensemble techniques.

6.1. Basic Data

We validate the numerical performance of both the base model and ensemble techniques for supply and demand forecasting through simulations. The supply and demand forecast datasets of the four seasons are the desired target variables in the multiple base models and robust ensemble techniques. The time series data forecast information utilizes 7 days ahead of PV, wind generation, and load demand for each season (Winter: from 21 January 2023 to 27 January 2023, Spring: from 22 April 2023 to 28 April 2023, Summer: from 22 July 2023 to 28 July 2023, and Fall: from 21 October 2023 to 27 October 2023), recorded at 15-min intervals, for conducting simulations.

The simulations are implemented in Python using the Keras and TensorFlow libraries. The dataset is split into training (70%), validation (15%), and testing (15%) subsets. Initially, multiple base model algorithms are developed with a key feature selection approach, and hyperparameters were carefully tuned during the training process. Detailed descriptions of model components and their parameters are provided in Table 1, while key features for the base models are provided in Table 2. These key features include historical electricity usage, generation data from PV and wind power, seasonal variations, and temporal features, such as hour of the day, day of the week, and season of the year, which play a crucial role in accurate short-term supply and demand forecasting. These features allow the model to adapt to daily, weekly, and seasonal variations, including differences between weekdays, weekends, holidays, and seasons. Thus, the results for holiday features are not included intentionally in this paper. Lagged features integrate historical data points, like previous average values and specific past data over intervals such as 24 h and one week (168 h in our model). By incorporating lagged features, the base model can leverage historical data to recognize and learn from past behaviors trends, and cyclical patterns.

As shown in Table 1, the hyperparameters for each base model algorithm are tuned carefully using a combination of random search methods to find the optimal configuration of hyperparameters and to ensure optimal performance in forecasting both energy supply and load demand. Hyperparameter optimization plays a crucial role in model performance as it governs the ability of the model to learn from the data, generalize across unseen samples, and avoid overfitting. We focuse on optimally selecting the following key hyperparameters for each base model:

For LSTM and RNN models, we initially test various hidden sizes (50, 100, and 200), and then set 100 hidden sizes that are optimal for improving the model's ability to capture temporal dependencies for time series forecasting tasks. The Adam optimizer is used with learning rates ranging from 0.0001 to 0.01 and a learning rate of 0.001 consistently obtains the best results, avoiding both underfitting and overfitting. A higher learning rate (0.01) leads to faster convergence, but it causes instability, while a lower value (0.0001) slows down learning without significant improvement in accuracy. Batch sizes are experimented with in the range of 16–128, and batch size 128 is chosen to be optimal, balancing computation time and model convergence. The model training is conducted over 100 epochs, with early stopping criteria to prevent overfitting based on the validation loss. We test with the activation function for the recurrent layer using tanh for its ability to handle vanishing gradients, and for the output layer, we apply the sigmoid activation function to ensure values between 0 and 1.

Table 1. Components and hyperparameter values for base models.

Components	Hyperparameters	Values
LSTM, RNN model (optimizer Adam)	Input size	1
	hidden size	100
	output size	1
	Batch size	128
	η	0.001
	Tanh and σ	(−1, 1) and (0, 1)
	Epochs	100
CNN model (optimizer Adam)	L	3
	l 1	(1, 16)
	l 2	(16, 32)
	l 3	(32, 64)
	K	3
	p, η	2, 0.001
	M	2
	m 1	(64 × 6, 100)
	m 2	(100, 1)
XGBoost model (optimizer Gradient Boosting)	Y	100
	η	0.05
	λ, γ	1.0, 0.2

Table 2. Features engineering.

Feature Index	Feature Selections	Unit/Scale	Data Category
1	Hour of the day	1–24	Seasonality/calendar
2	Day of the week	1–7	
3	Season of the year	1–4	
4	Period of the day	1–4	Temporal indicators
5	Weekend indicator	0–1	
6	Holiday indicator	0–1	
7	Previous 24 h average PV generation	kW	Electricity generated
8	Previous 24 h average wind generation	kW	
9	24 h lagged PV generation	kW	
10	24 h lagged wind generation	kW	
11	168 h lagged PV generation	kW	
12	168 h lagged wind generation	kW	
13	Previous 24 h average load demand	kW	Energy demand
14	24 h lagged load demand	kW	
15	168 h lagged load demand	kW	

For the CNN model, the tuning focuses on designing three convolutional layers to progressively extract features at increasing levels and the model uses kernel sizes ranging from (1, 16), (16, 32), and (32, 64). These choices are based on experimentation with smaller and larger kernels, where the above sizes provide the best performance on validation data, capturing both fine-grained features at lower layers and more abstract features at higher layers. The dropout rate is set to 2, and the Adam optimizer is used with the learning rate set to 0.001. Max-pooling layers are tested with optimal pool sizes 2, providing enough downsampling without losing important feature information. The fully connected layers are sized as (64 × 6, 100) for the first layer and (100, 1) for the second layer, which allows the model to learn complex combinations of extracted features without overly complicating the model process. We apply the ReLU activation function in the convolutional layers, which is standard for CNNs, as it helps mitigate the vanishing gradient problem and accelerates convergence.

For the XGBoost model, we focus on optimally selecting the key hyperparameters learning rate (η), number of estimators (Y), and regularization parameters (λ and γ). The learning rates are tested ranged from 0.01 to 0.1, and a learning rate of 0.05 obtains the best results. Higher learning rate (0.1) leads to overfitting, while lower rate (0.01) slows down learning without a significant gain in accuracy. We also experiment with $\lambda = 1.0$ and $\gamma = 0.2$ as regularization parameters. The λ parameter controls the L2 regularization (reducing overfitting), and γ controls the minimum loss reduction required to make a further partition.

In general, the optimal hyperparameter tuning values lead to a model configuration that consistently outperform other values in terms of reducing overfitting, minimizing forecast errors, and improving accuracy and efficiency.

6.2. Base Models Training and Validation Analysis

Figures 2–4 demonstrate the base models training and validation loss curves of PV, wind, and load demand forecasts across different seasons. The training and validation loss curves (measured as MSE) for multiple base models (LSTM, CNN, XGBoost, RNN) consistently reveal distinct performance patterns across different seasons (Winter, Spring, Summer, and Fall) based on the optimal hyperparameter values. The LSTM, CNN, and RNN models are particularly suited for time-series forecasting tasks because they can capture spatial and temporal dependencies in data. The observation of rapid decreases in training loss and closely aligned validation loss indicates that models are effectively learning the underlying patterns in the data while also generalizing well to unseen data. Specifically, the LSTM model is designed to capture long-term dependencies in time-series data forecasting tasks where seasonal and cyclical trends are important. The smooth decrease in the training and validation loss curves indicate that the LSTM model performed these long-term dependencies well. The CNN model is also effective in time-series forecasting by treating time-series data as a spatial dimension which is produced in the lower loss curves observed for all seasons. The RNN model is another ML model designed for sequential data, while it may not capture long-term dependencies as well as LSTMs, it performs well on time-series tasks, which is shown in the good training and validation loss performance. Conversely, XGBoost models tend to overfit, as shown by higher validation losses compared to training losses. The higher validation loss indicates that XGBoost struggles to directly model temporal correlation on generalized datasets compared to LSTM, CNN, and RNN models. In this case, the proposed ensemble forecasting techniques focus on reducing this overfitting and combining the strengths of the base models to improve forecast accuracy.

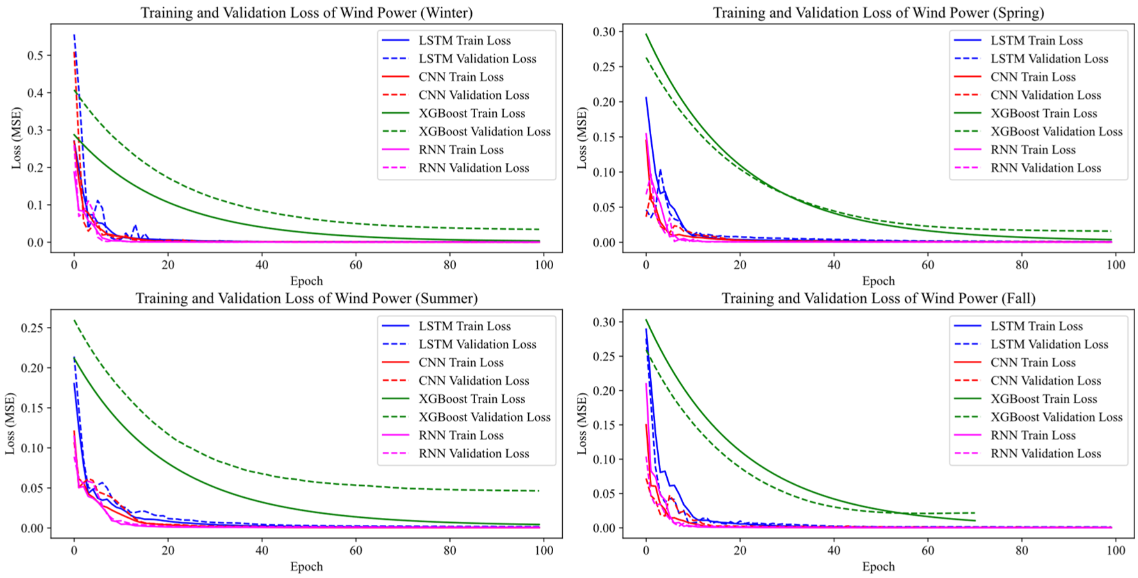

Figure 2. Wind power training and validation loss curves for each base model across all seasons.

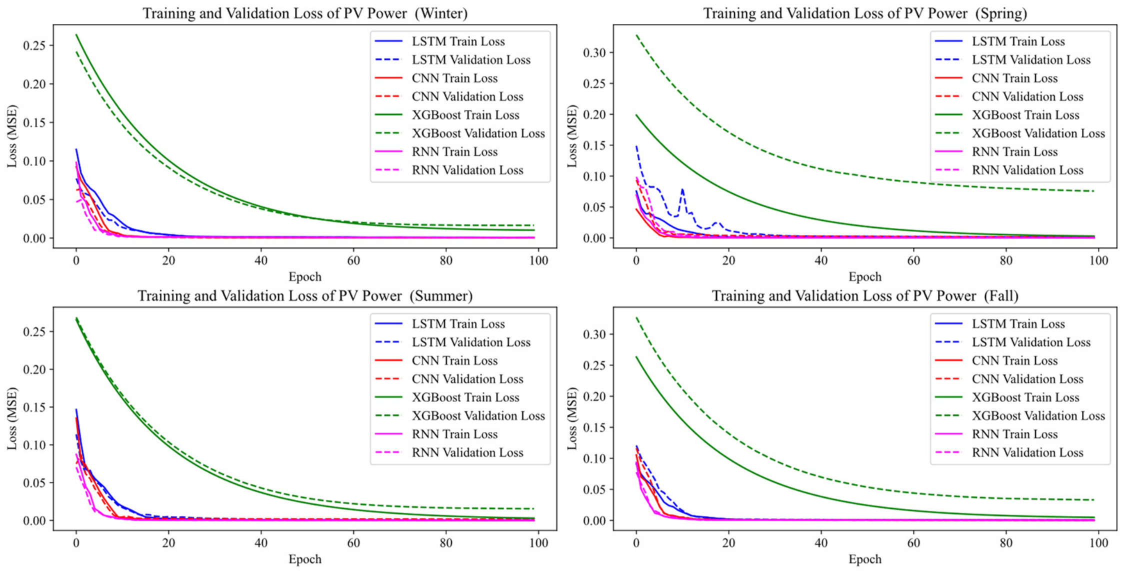

Figure 3. PV power training and validation loss curves for each base model across all seasons.

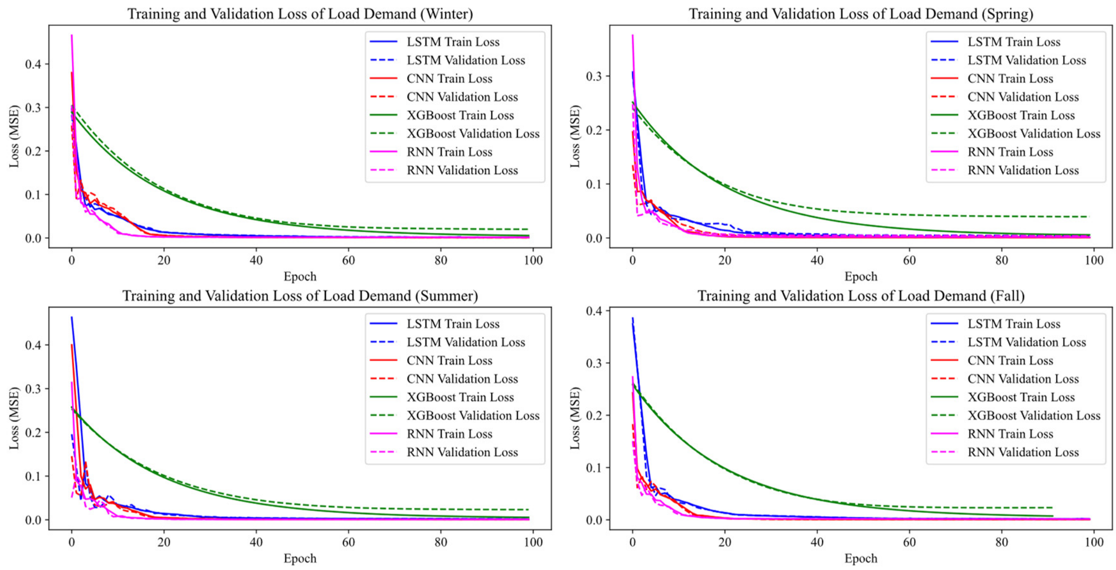

Figure 4. Load demand training and validation loss curves for each base model across all seasons.

6.3. Comparison Analysis of Forecast vs. Actuals for the Base Models and Ensemble Techniques

Figures 5–7 demonstrate the comparison between the forecasted and actual values for PV and wind generations, and load demand data across different seasons. The simulation result validates the performance of multiple base models (LSTM, CNN, XGBoost, and RNN) and the proposed robust ensemble forecasting techniques (SA, WA, and Stacking). Across all seasons, the base model forecast values closely follow the actual values well, with minor deviations (forecast errors). This is attributed to the appropriate feature selection and hyperparameter tuning performed for each base model. The selected features effectively capture key patterns in the data, while the hyperparameter tuning optimizes each model's ability to learn from the data without overfitting, ensuring good generalization. Well-tuned models show lower bias and better generalization but still exhibit small inconsistencies and forecast errors due to the base model's ability to forecast and limitations. Therefore, the base models are not guaranteed to obtain the optimal results, which need to be improved forecast limitations (forecast errors and overfitting) through ensemble forecasting techniques.

The proposed ensemble forecasting techniques (SA, WA, and Stacking) all demonstrated excellent performance. The SA forecasting technique averages the forecasts of the base models and adopts each model to contribute equally, it smooths out forecasts and reduces errors. The WA forecasting technique is more flexible and improves by assigning different weights to each base model based on their relative performance. However, both the SA and WA forecasting techniques still struggle with forecast errors and overfitting compared to Stacking methods. The stacking forecasting technique is the most advanced of the ensemble techniques. It involves training a meta-model to learn the optimal combination of the multiple base models' forecasts. Stacking takes advantage of the diverse strengths of each base model and learns to combine their outputs to minimize overall forecast error. Across all seasons, the Sacking forecast results consistently outperform both the individual base models and SA and WA techniques by providing the most accurate and smooth forecasts, demonstrating its superior forecast accuracy. This superior performance is evident in all seasons. The reason is that the Stacking technique effectively integrates the strengths of individual base models by learning optimal combinations of their forecasts using the meta-model training process. The meta-model takes the forecasts from all base models as input and learns how to best combine them using a training process. It can identify which base models perform better under certain conditions, learning to assign higher weights to the more reliable models and correct individual biases and errors.

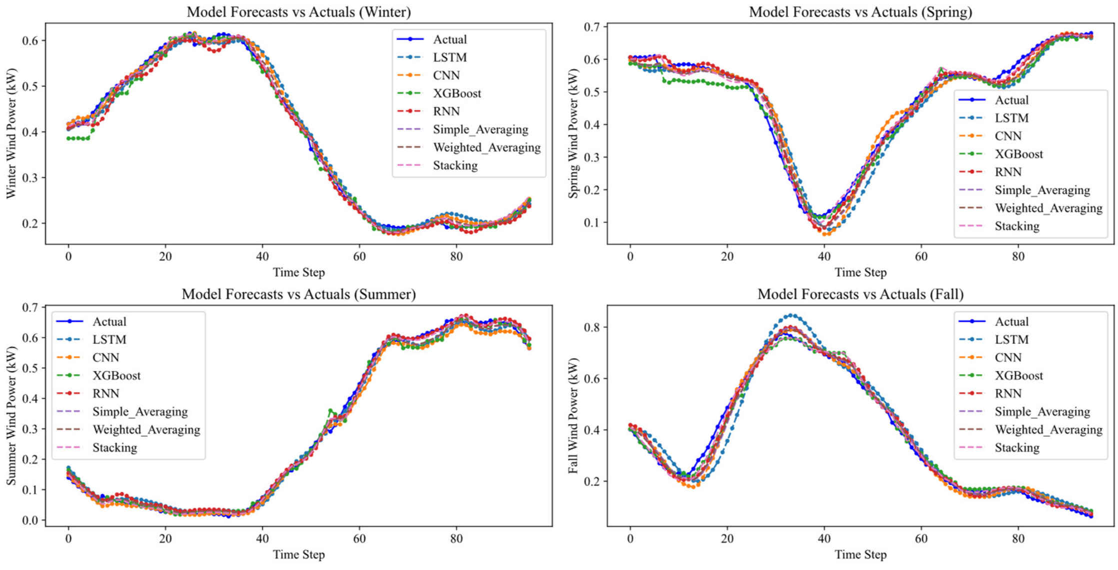

Figure 5. Comparison between forecasted and actual wind generation data across seasons.

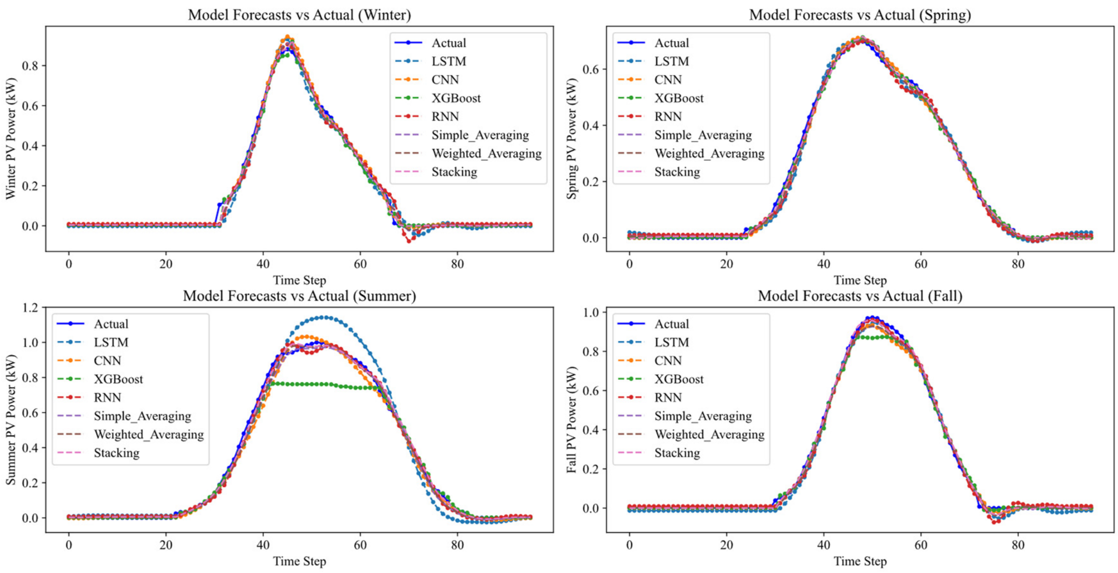

Figure 6. Comparison between forecasted and actual PV generation data across seasons.

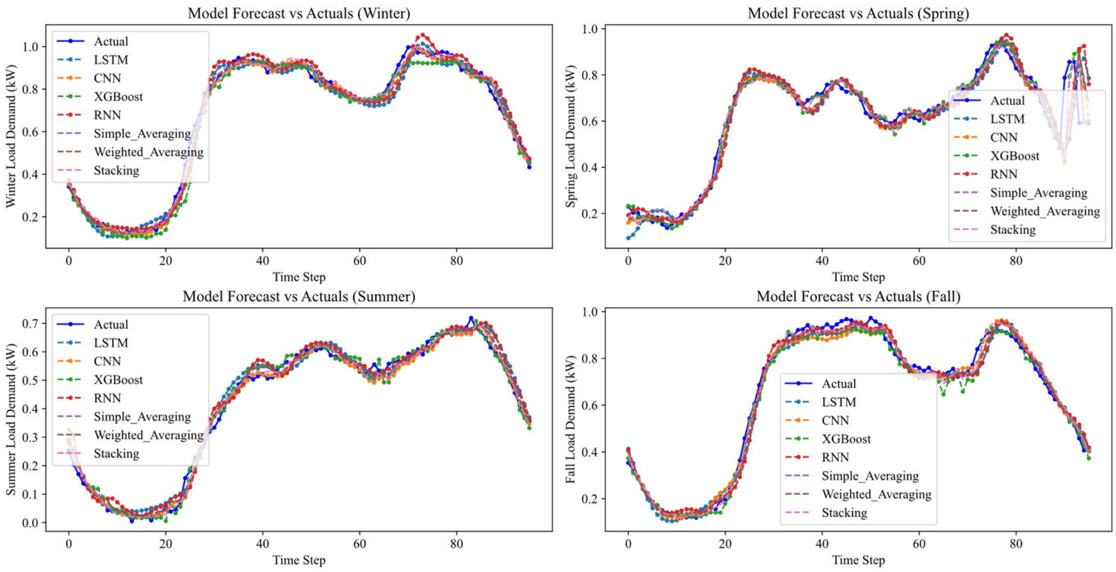

Figure 7. Comparison between forecasted and actual load demand data across seasons.

6.4. Comparison Analysis of Forecast vs. Actuals for the Ensemble Techniques

Figures 8–10 demonstrate the comparisons between forecasted and actual data for energy supply and load demand using the proposed ensemble forecasting techniques (SA, WA, and Stacking) across different seasons. The Stacking ensemble forecasting technique consistently produces the most accurate forecasts. This is a result of its unique algorithmic design which Stacking uses a meta-model that combines the forecasts of multiple base models (LSTM, CNN, RNN, XGBoost) to form a final forecast. The meta-model learns the optimal way to combine the base models' forecasts by weighting their contributions based on performance. This process allows the meta-model to reduce the bias and variance of individual base models, mainly to address overfitting problems that struggle in base models (e.g., XGBoost), and generalization errors in (LSTM, CNN, and RNN) models. The superior accuracy of Stacking across all seasons is a direct result of this adaptive combination process, which allows it to perform well. The SA and WA forecasting techniques also perform well, offering robust forecasts but are slightly outperformed by Stacking. In SA, all base models contribute equally to the final forecast, without any consideration for their relative performance. WA technique improves on SA by assigning different weights to each base model's forecast based on their performance (e.g., lower error models get higher weights). This provides better forecast robustness compared to SA but still lacks the meta-model learning capabilities of Stacking. The weights in WA are static and not learned dynamically from the data, which limits its adaptability in capturing complex relationships within the data compared to the Stacking method.

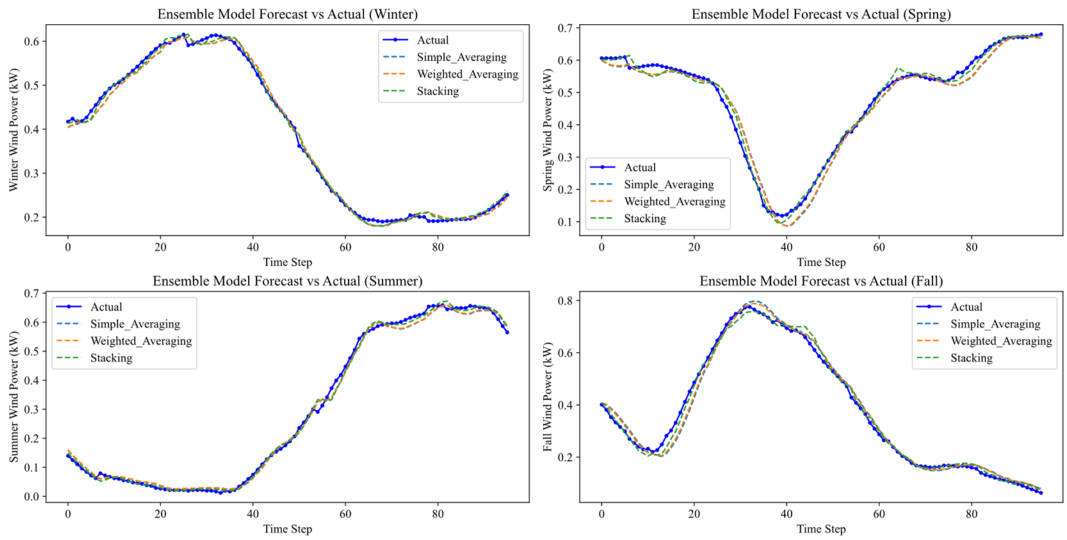

Figure 8. Comparisons between forecasted and actual wind generation data across seasons.

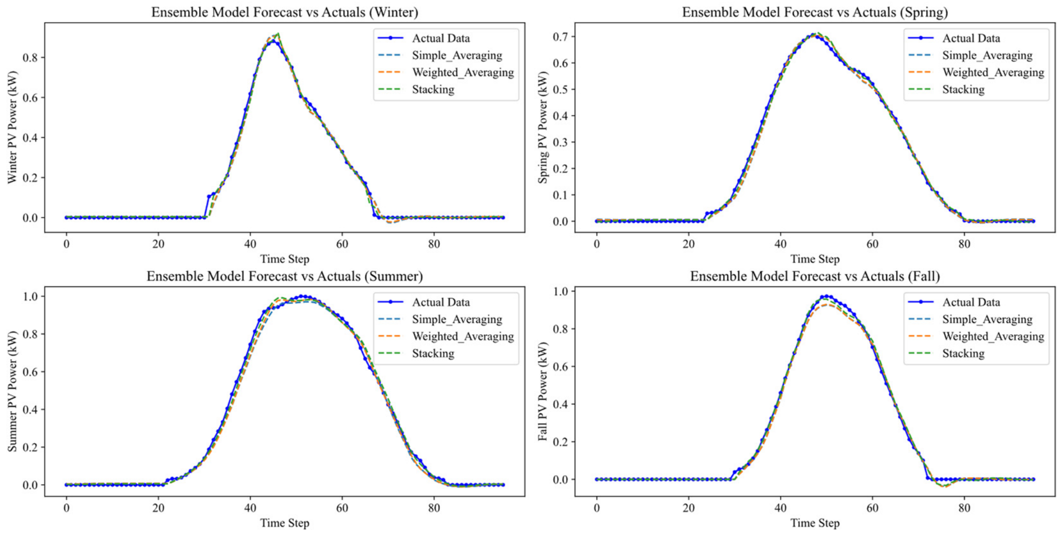

Figure 9. Comparisons between forecasted and actual PV generation data across seasons.

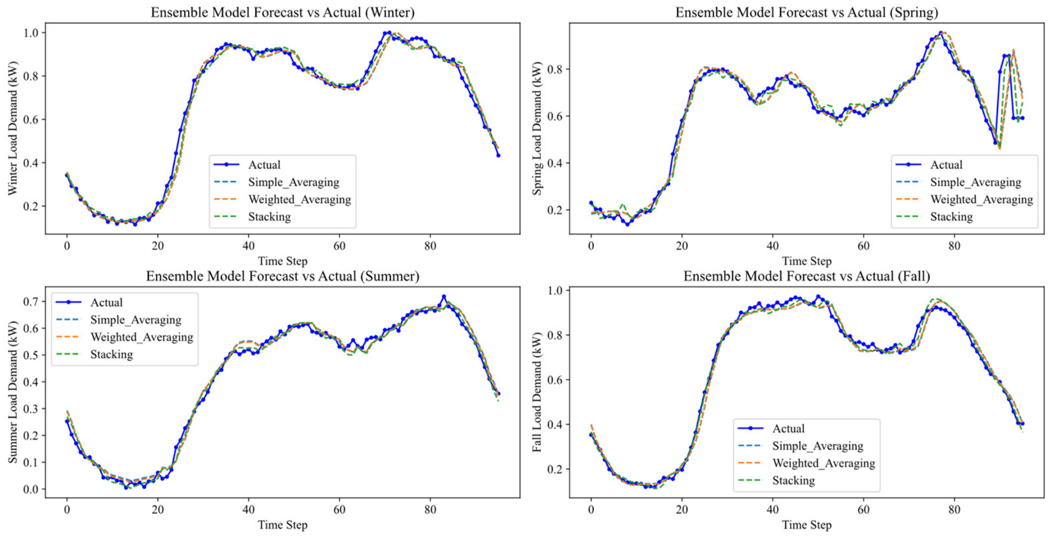

Figure 10. Comparisons between forecasted and actual load demand data across seasons.

6.5. Quantitative Performance Evaluation Analysis for Base Models and Ensemble Techniques

Tables 3–5 present the quantitative performance evaluation metrics for wind and PV generations and load demand forecasting across different seasons, using MAE, MAPE, SSE, RMSE, SDE, NMAE, and R-squared (R^2) as key performance indicators. The base models (LSTM, CNN, XGBoost, and RNN) recorded varied levels of forecast accuracy, indicating minor forecast errors and lower R-squared (R^2) scores compared to the proposed ensemble forecasting techniques (SA, WA, Stacking). For instance, we can see the base models and the ensemble forecasting techniques comparisons for winter season wind generation in terms of MAE, MAPE, RMSE, and R^2 evaluation metrics.

In terms of MAE, the LSTM model shows a value of 0.0106, indicating a moderate level of error. The CNN model, on the other hand, achieves a significantly lower MAE value of 0.0072. The XGBoost model performed with an MAE value of 0.0100, while the RNN model performed slightly better with an MAE value of 0.0096. Among the ensemble techniques, both SA and WA performed with MAE values of 0.0075 and 0.0074, respectively. The Stacking ensemble techniques, however, outperformed all models with the lowest MAE value of 0.0062, indicating the highest accuracy and the ability to minimize forecast errors.

When we look at the MAPE, the LSTM model has the highest error percentage at 2.9944%, indicating that the model struggles to maintain accuracy relative to the actual wind generation. The CNN model performs better with a MAPE of 2.2001%, and XGBoost performed with a MAPE of 2.7569%. The RNN model achieves a MAPE of 2.5827%, slightly better than XGBoost but still higher than CNN. Among the ensemble techniques, both SA and WA achieved better performance with a MAPE of 2.1254% and 2.1059% than the base models but still higher than the Stacking ensemble technique. The Stacking leads with the lowest MAPE value of 1.8607%, demonstrating superior accuracy in forecasting relative to the base models and SA and WA.

In terms of RMSE, the LSTM model performs at 0.0133, while CNN performs better at 0.0089. The XGBoost model is close to LSTM with 0.0142 and the RNN model performs at 0.0125. Among the ensemble techniques, both SA and WA perform with RMSE values of 0.0095 and 0.0093, respectively. The Stacking continues to outperform all models, achieving the lowest RMSE of 0.0079, indicating a superior ability to minimize large errors and offer more reliable forecasts.

The R-squared (R^2) evaluation metric indicates how well the model fits the data, and how much variance in the target variable it can explain. Higher R-squared values indicate better performance and a better fit to the data. The LSTM model performs a value of 0.9932, meaning it explains 99.32% of the variance. The CNN model performs slightly better with an R-squared value of 0.9970, suggesting it is better at capturing the patterns in the data. The XGBoost model performs a value of 0.9923, which is slightly lower than LSTM and CNN. The RNN model performs with a value of 0.9940. Among the ensemble techniques, SA and WA show R-squared values of 0.9966 and 0.9967, respectively. The Stacking ensemble achieves the highest R-squared value of 0.9976, indicating that it explains the most variance and provides the best overall fit to the data.

Generally, similar trends are observed in spring, summer, and fall for wind and PV generations and load demand forecasting as shown in Tables 3–5. In all seasons, the proposed Stacking ensemble forecasting techniques significantly improve forecast accuracy. It is the most effective ensemble technique that consistently achieves the lowest forecast errors across all evaluation metrics and highest R^2 values, indicating superior forecast accuracy. It reduces forecast errors and improves the overall model performance, which is suitable for optimal forecasting techniques due to its ability to leverage the strengths of multiple base models while mitigating their weaknesses.

Table 3. Performance evaluation metrics across different seasons for wind generation.

Season	MAE	MAPE %	SSE	RMSE	SDE	NMAE	R^2
LSTM model wind power forecast performance metrics							
Winter	0.0106	2.9944	0.0174	0.0133	0.0133	0.0276	0.9932
Spring	0.0307	9.6563	0.1457	0.0386	0.0357	0.0639	0.9452
Summer	0.0185	15.6286	0.0513	0.0229	0.0206	0.0633	0.9920
Fall	0.0323	10.6982	0.2105	0.0463	0.0458	0.0860	0.9569
CNN model wind power forecast performance metrics							
Winter	0.0072	2.2001	0.0077	0.0089	0.0087	0.0188	0.9970
Spring	0.0256	7.8091	0.0918	0.0306	0.0291	0.0534	0.9655
Summer	0.0115	14.2949	0.0189	0.0139	0.0138	0.0392	0.9971
Fall	0.0251	9.2082	0.0921	0.0307	0.0296	0.0668	0.9812
XGBoost model wind power forecast performance metrics							
Winter	0.0100	2.7569	0.0198	0.0142	0.0126	0.0261	0.9923
Spring	0.0213	5.0292	0.0676	0.0263	0.0212	0.0444	0.9746
Summer	0.0120	12.6375	0.0272	0.0167	0.0167	0.0409	0.9958
Fall	0.0154	5.7157	0.0374	0.0195	0.0195	0.0411	0.9924
RNN model wind power forecast performance metrics							
Winter	0.0096	2.5827	0.0154	0.0125	0.0124	0.0250	0.9940
Spring	0.0133	4.2529	0.0303	0.0176	0.0176	0.0278	0.9886
Summer	0.0129	11.6133	0.0266	0.0165	0.0155	0.0442	0.9959
Fall	0.0173	5.6676	0.0491	0.0224	0.0223	0.0461	0.9900
Ensemble SA wind power forecast performance metrics							
Winter	0.0075	2.1254	0.0088	0.0095	0.0093	0.0195	0.9966
Spring	0.0202	6.0840	0.0590	0.0245	0.0225	0.0420	0.9778
Summer	0.0131	13.0105	0.0234	0.0154	0.0150	0.0446	0.9964
Fall	0.0194	6.8455	0.0657	0.0259	0.0259	0.0515	0.9866
Ensemble WA wind power forecast performance metrics							
Winter	0.0074	2.1059	0.0084	0.0093	0.0092	0.0192	0.9967
Spring	0.0181	5.5059	0.0483	0.0222	0.0206	0.0377	0.9818
Summer	0.0126	12.8875	0.0217	0.0149	0.0146	0.0430	0.9966
Fall	0.0174	6.2273	0.0513	0.0229	0.0228	0.0464	0.9895
Ensemble Stacking wind power forecast performance metrics							
Winter	0.0062	1.8607	0.0061	0.0079	0.0079	0.0161	0.9976
Spring	0.0100	2.7903	0.0180	0.0135	0.0135	0.0209	0.9932
Summer	0.0087	8.0278	0.0134	0.0117	0.0117	0.0298	0.9979
Fall	0.0125	4.3857	0.0264	0.0164	0.0164	0.0332	0.9946

Table 4. Performance evaluation metrics across different seasons for PV generation.

Season	MAE	MAPE %	SSE	RMSE	SDE	NMAE	R^2
LSTM model PV power forecast performance metrics							
Winter	0.0212	29.1301	0.0718	0.0271	0.0269	0.1219	0.9902
Spring	0.0121	15.2558	0.0297	0.0174	0.0163	0.0569	0.9953
Summer	0.0428	35.1984	0.3516	0.0599	0.0599	0.1295	0.9756
Fall	0.0185	22.5614	0.0468	0.0219	0.0219	0.0800	0.9959
CNN model PV power forecast performance metrics							
Winter	0.0130	16.8642	0.0422	0.0207	0.0199	0.0745	0.9943
Spring	0.0076	13.3889	0.0116	0.0109	0.0108	0.0358	0.9982
Summer	0.0214	14.3383	0.1088	0.0333	0.0291	0.0648	0.9925
Fall	0.0137	29.2259	0.0487	0.0223	0.0216	0.0592	0.9957
XGBoost model PV power forecast performance metrics							
Winter	0.0094	17.0617	0.0301	0.0175	0.0172	0.0538	0.9959
Spring	0.0069	6.8920	0.0106	0.0104	0.0103	0.0323	0.9983
Summer	0.0482	18.2153	0.7782	0.0891	0.0790	0.1459	0.9461
Fall	0.0142	23.6297	0.0691	0.0266	0.0257	0.0613	0.9939
RNN model PV power forecast performance metrics							
Winter	0.0169	32.7721	0.0656	0.0259	0.0257	0.0971	0.9911
Spring	0.0129	14.7781	0.0239	0.0156	0.0155	0.0607	0.9962
Summer	0.0174	12.9255	0.0510	0.0228	0.0226	0.0528	0.9965
Fall	0.0161	18.0759	0.0530	0.0233	0.0223	0.0698	0.9953
Ensemble SA PV power forecast performance metrics							
Winter	0.0129	21.8401	0.0348	0.0188	0.0187	0.0739	0.9953
Spring	0.0088	10.8259	0.0136	0.0118	0.0117	0.0415	0.9979
Summer	0.0184	13.6350	0.0656	0.0259	0.0213	0.0556	0.9955
Fall	0.0142	21.9498	0.0388	0.0199	0.0193	0.0614	0.9966
Ensemble WA PV power forecast performance metrics							
Winter	0.0120	20.7230	0.0315	0.0179	0.0179	0.0692	0.9957
Spring	0.0083	10.1415	0.0122	0.0112	0.0111	0.0390	0.9981
Summer	0.0162	13.5924	0.0514	0.0229	0.0204	0.0490	0.9964
Fall	0.0143	22.0721	0.0385	0.0198	0.0193	0.0619	0.9966
Ensemble Stacking PV power forecast performance metrics							
Winter	0.0080	15.5166	0.0237	0.0155	0.0155	0.0461	0.9968
Spring	0.0065	9.2584	0.0083	0.0092	0.0092	0.0305	0.9987
Summer	0.0107	7.6657	0.0240	0.0156	0.0156	0.0323	0.9983
Fall	0.0091	23.9291	0.0197	0.0142	0.0142	0.0395	0.9983

Table 5. Performance evaluation metrics across different seasons for load demand.

Season	MAE	MAPE %	SSE	RMSE	SDE	NMAE	R^2
LSTM model load demand forecast performance metrics							
Winter	0.0367	7.8920	0.2092	0.0462	0.0424	0.0553	0.9785
Spring	0.0494	9.4250	0.5820	0.0771	0.0768	0.0813	0.8877
Summer	0.0323	29.4246	0.1366	0.0373	0.0370	0.0780	0.9842
Fall	0.0309	6.1850	0.1466	0.0387	0.0377	0.0481	0.9821
CNN model load demand forecast performance metrics							
Winter	0.0205	4.5444	0.0678	0.0263	0.0229	0.0308	0.9921
Spring	0.0311	7.0858	0.3515	0.0599	0.0587	0.0512	0.9216
Summer	0.0171	11.2077	0.0405	0.0203	0.0203	0.0412	0.9918
Fall	0.0242	3.8648	0.0823	0.0290	0.0263	0.0377	0.9899
XGBoost model load demand forecast performance metrics							
Winter	0.0260	5.4786	0.1344	0.0370	0.0336	0.0390	0.9848
Spring	0.0261	5.3031	0.2714	0.0526	0.0526	0.0429	0.9449
Summer	0.0194	21.3487	0.0610	0.0249	0.0239	0.0469	0.9877
Fall	0.0319	5.0690	0.1770	0.0425	0.0352	0.0498	0.9783
RNN model load demand forecast performance metrics							
Winter	0.0296	5.5747	0.1434	0.0383	0.0356	0.0445	0.9845
Spring	0.0482	8.7706	0.7128	0.0853	0.0844	0.0794	0.8616
Summer	0.0230	21.1188	0.0755	0.0277	0.0277	0.0556	0.9835
Fall	0.0246	5.3405	0.1069	0.0330	0.0315	0.0383	0.9858
Ensemble SA load demand forecast performance metrics							
Winter	0.0243	4.2201	0.1060	0.0329	0.0291	0.0366	0.9900
Spring	0.0350	6.8097	0.3882	0.0629	0.0625	0.0575	0.9212
Summer	0.0189	18.5597	0.0520	0.0230	0.0227	0.0455	0.9901
Fall	0.0240	4.4879	0.0921	0.0307	0.0292	0.0374	0.9889

Table 5. *Cont.*

Season	MAE	MAPE %	SSE	RMSE	SDE	NMAE	R^2
Ensemble WA load demand forecast performance metrics							
Winter	0.0234	4.0801	0.0981	0.0316	0.0279	0.0352	0.9905
Spring	0.0326	6.4452	0.3556	0.0602	0.0598	0.0537	0.9278
Summer	0.0179	17.4138	0.0475	0.0220	0.0217	0.0433	0.9915
Fall	0.0239	4.3549	0.0901	0.0303	0.0289	0.0372	0.9900
Ensemble Stacking load demand forecast performance metrics							
Winter	0.0168	3.6452	0.0443	0.0213	0.0213	0.0252	0.9950
Spring	0.0282	6.2755	0.2367	0.0491	0.0491	0.0464	0.9534
Summer	0.0155	12.3752	0.0368	0.0194	0.0194	0.0374	0.9933
Fall	0.0200	3.5093	0.0596	0.0247	0.0247	0.0312	0.9939

6.6. Comparison Analysis for R^2 Scores

Figures 11–13 demonstrate the R^2 scores for a comparison between base models and proposed ensemble forecasting techniques across different seasons. The R-squared (R^2) evaluation metric indicates how well the model fits the data, and how much variance in the target variable it can explain. A higher R^2 value indicates better model performance and a better fit to the data. The R^2 simulation outcome proves that the proposed Stacking ensemble forecasting technique consistently outperforms compared to base models, as well as SA and WA techniques across all seasons. It shows higher R^2 scores observed and superior ability to capture and forecast the complex patterns in supply and demand fluctuations. From an algorithmic design perspective, the Stacking ensemble technique's superior performance can be attributed to its ability to combine the strengths of multiple base models. By stacking a variety of models and using a meta-model to make final forecasts, the technique effectively reduces bias and variance, leading to a more robust and generalizable forecasting model. Additionally, the stacking method allows for the identification and weighting of the most informative features and patterns in the data, thereby improving forecast consistency and accuracy across all seasons in DES. In contrast, the base models, which rely on single algorithms, tend to show more variability and lower R^2 scores, as they are less equipped to handle the complex and diverse data patterns present in supply and demand forecasting. The SA and WA techniques are performed well, but outperformed by Stacking.

In general, the Stacking ensemble technique proves to be the most effective forecasting algorithm for short-term supply and demand forecasting, offering the highest R^2 scores and significantly improving the consistency and accuracy of forecasts across all seasonal variations.

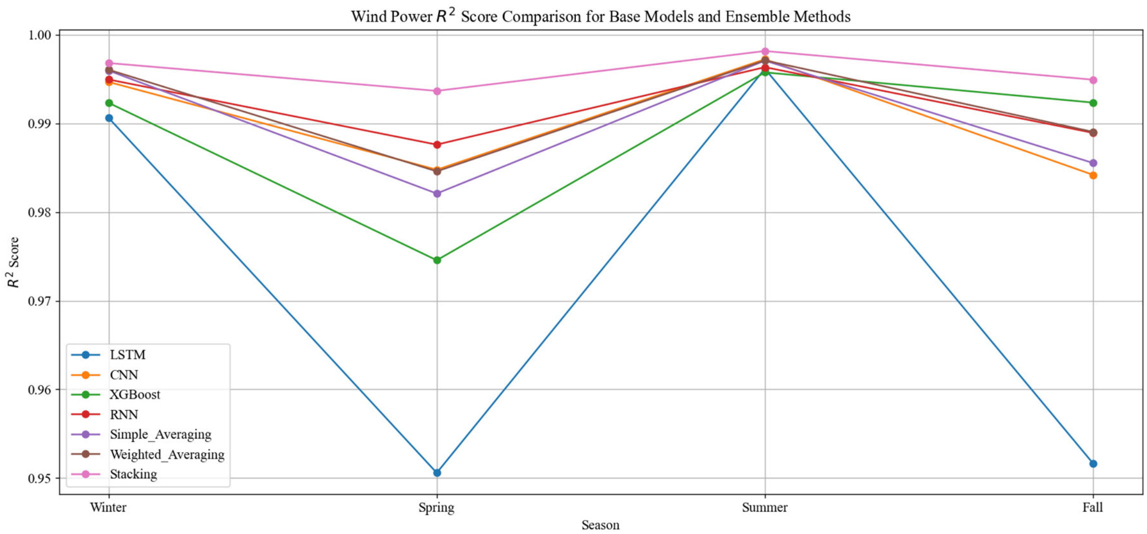

Figure 11. Score comparison for wind generation across seasons.

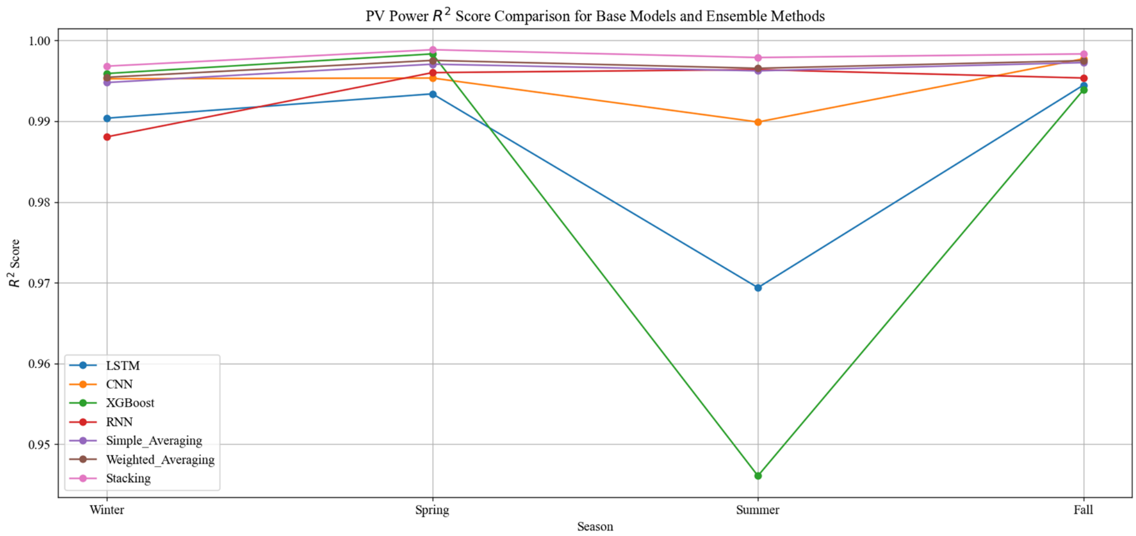

Figure 12. Score comparison for PV generation across seasons.

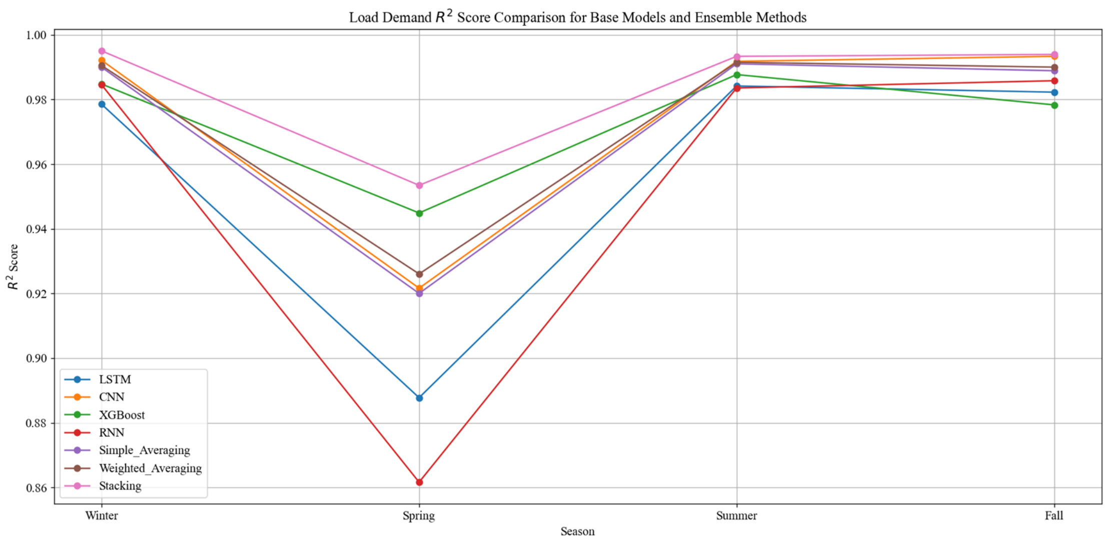

Figure 13. Score comparison for load demand across seasons.

7. Conclusion

This study focuses on short-term energy supply and electricity demand forecasting systems considered within small-scale DES in the smart grid context, including PV, wind generation, and load demand. We developed multiple base model algorithms with feature selection techniques, to improve forecast accuracy across different seasons. Then, robust ensemble forecasting techniques including SA, WA, and Stacking were proposed and implemented to further improve forecast accuracy.

Numerical simulations using Python, Keras, and TensorFlow validated the effectiveness of the base models and ensemble techniques. The results showed that the base models effectively captured temporal, spatial, and seasonal variations, improving forecast accuracy across seasons. Among the ensemble techniques, Stacking consistently outperformed the individual models and other ensemble methods (SA and WA), achieving higher R^2 scores and lower forecast errors. This was due to Stacking's ability to combine the strengths of base models through optimal forecast integration using a meta-model. The ensemble approach reduced forecast errors and improved generalization to new data, providing a robust solution for short-term supply and demand forecasting. In future work, we will further investigate the optimization of ensemble techniques for forecasting in more complex grid environments.

Author Contributions

A.G.J.: Conceptualization, Methodology, Software, Writing—Original Draft Preparation; R.B.: Data Curation, Writing—Original Draft Preparation; Z.Y.: Visualization, Investigation; Z.W.: Supervision; Z.Z. and

X.W.: Validation, Writing—Reviewing and Editing. All authors have read and agreed to the published version of the manuscript.

Funding

This work is supported by Science and Technology Project of China Southern Power Grid Company Limited under Grant Number 1500002023030103JL00320.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

Not applicable.

Acknowledgments

The authors would like to thank the State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources at North China Electric Power University, for its support of this paper.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Adil, A.M.; Ko, Y. Socio-technical evolution of Decentralized Energy Systems: A critical review and implications for urban planning and policy. *Renew. Sustain. Energy Rev.* **2016**, *57*, 1025–1037.
2. Zhang, S.; Xue, J.; Liu, J.; et al. Differential Privacy-Aware Generative Adversarial Network-Assisted Resource Scheduling for Green Multi-Mode Power IoT. *IEEE Trans. Green Commun. Netw.* **2024**, *8*, 956–967.
3. Nadeem, T.B.; Siddiqui, M.; Khalid, M.; et al. Distributed energy systems: A review of classification, technologies, applications, and policies. *Energy Strategy Rev.* **2023**, *48*, 101096.
4. Liao, H.; Zhou, Z.; Jia, Z.; et al. Ultra-Low AoI Digital Twin-Assisted Resource Allocation for Multi-Mode Power IoT in Distribution Grid Energy Management. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 3122–3132.
5. Hong, B.; Li, Q.; Chen, W.; et al. Supply modes for renewable-based distributed energy systems and their applications: Case studies in China. *Glob. Energy Interconnect.* **2020**, *3*, 259–271.
6. Wang, Z.; Zhou, Z.; Zhang, H.; et al. AI-Based Cloud-Edge-Device Collaboration in 6G Space-Air-Ground Integrated Power IoT. *IEEE Wirel. Commun.* **2022**, *29*, 16–23.
7. Zhang, Y.; Pan, Z.; Wang, H.; et al. Achieving wind power and photovoltaic power prediction: An intelligent prediction system based on a deep learning approach. *Energy* **2023**, *283*, 129005.
8. Li, J.; Zhou, Z.; Wu, J.; et al. Decentralized On-Demand Energy Supply for Blockchain in Internet of Things: A Microgrids Approach. *IEEE Trans. Comput. Soc. Syst.* **2019**, *6*, 1395–1406.
9. Luo, X.; Zhang, D.; Zhu, X. Deep learning based forecasting of photovoltaic power generation by incorporating domain knowledge. *Energy* **2021**, *225*, 120240.
10. Li, J.; Wang, K.; Hou, X.; et al. A Dual-Scale Transformer-Based Remaining Useful Life Prediction Model in Industrial Internet of Things. *IEEE Internet Things J.* **2024**, *11*, 26656–26667.
11. Liu, X.; Zhou, J. Short-term wind power forecasting based on multivariate/multi-step LSTM with temporal feature attention mechanism. *Appl. Soft Comput.* **2024**, *150*, 111050.
12. Samanta, A.; Nguyen, T.G.; Ha, T.; et al. Distributed Resource Distribution and Offloading for Resource-Agnostic Microservices in Industrial IoT. *IEEE Trans. Veh. Technol.* **2023**, *72*, 1184–1195.
13. Gao, X.; Guo, W.; Mei, C.; et al. Short-term wind power forecasting based on SSA-VMD-LSTM. *Energy Rep.* **2023**, *9*, 335–344.
14. Li, Y.L.; Zhu, Z.A.; Chang, Y.K.; et al. Short-Term Wind Power Forecasting by Advanced Machine Learning Models. In Proceedings of the 2020 International Symposium on Computer, Consumer and Control (IS3C), Taichung City, Taiwan, 13–16 November 2020; pp. 412–415.

15. Agga, A.; Abbou, A.; Labbadi, M.; et al. CNN-LSTM: An efficient hybrid deep learning architecture for predicting short-term photovoltaic power production. *Electr. Power Syst. Res.* **2022**, *208*, 107908.
16. Krishnan, N.; Kumar, K.R. Solar radiation forecasting using gradient boosting based ensemble learning model for various climatic zones. *Sustain. Energy Grids Netw.* **2024**, *38*, 101312.
17. Samadianfard, S.; Hashemi, S.; Kargar, K.; et al. Wind speed prediction using a hybrid model of the multi-layer perceptron and whale optimization algorithm. *Energy Rep.* **2020**, *6*, 1147–1159.
18. Al-Ja'afreh, M.A.A.; Mokryani, G.; Amjad, B. An enhanced CNN-LSTM based multi-stage framework for PV and load short-term forecasting: DSO scenarios. *Energy Rep.* **2023**, *10*, 1387–1408.
19. Muzaffar, S.; Afshari, A. Short-Term Load Forecasts Using LSTM Networks. *Energy Procedia* **2019**, *158*, 2922–2927.
20. Agga, A.; Abbou, A.; Labbadi, M.; et al. Short-Term Load Forecasting: Based on Hybrid CNN-LSTM Neural Network. In Proceedings of the 2021 6th International Conference on Power and Renewable Energy (ICPRE), Shanghai, China, 17–20 September 2021; pp. 886–891.
21. Phan, Q.T.; Wu, Y.K.; Phan, Q.D. Short-term Solar Power Forecasting Using XGBoost with Numerical Weather Prediction. In Proceedings of the 2021 IEEE International Future Energy Electronics Conference (IFEEC), Taipei, Taiwan, 16–19 November 2021; pp. 1–6.
22. Baul, A.S.; Sikder, G.C.; Mozumder, P.; et al. Data-Driven Short-Term Load Forecasting for Multiple Locations: An Integrated Approach. *BDCC* **2024**, *8*, 12.
23. Wang, N.; Li, Z. A stacking-based short-term wind power forecasting method by CBLSTM and ensemble learning. *J. Renew. Sustain. Energy* **2022**, *14*, 046101.
24. Trivedi, R.; Patra, S.; Khadem, S. A Data-Driven Short-Term PV Generation and Load Forecasting Approach for Microgrid Applications. *IEEE J. Emerg. Sel. Top. Ind. Electron.* **2022**, *3*, 911–919.
25. Moreno, S.R.; da Silva, R.G.; Mariani, V.C.; et al. Multi-step wind speed forecasting based on hybrid multi-stage decomposition model and long short-term memory neural network. *Energy Convers. Manag.* **2020**, *213*, 112869.
26. Cao, W.; Wang, G.; Liang, X.; et al. A STAM-LSTM model for wind power prediction with feature selection. *Energy* **2024**, *296*, 131030.
27. Kumari, P.; Toshniwal, D. Extreme gradient boosting and deep neural network based ensemble learning approach to forecast hourly solar irradiance. *J. Clean. Prod.* **2021**, *279*, 123285.
28. Mbey, C.F.; Kakeu, V.J.F.; Boum, A.T.; et al. Solar photovoltaic generation and electrical demand forecasting using multi-objective deep learning model for smart grid systems. *Cogent Eng.* **2024**, *11*, 2340302.
29. Lim, S.-C.; Huh, J.-H.; Hong, S.-H.; et al. Solar Power Forecasting Using CNN-LSTM Hybrid Model. *Energies* **2022**, *15*, 8233.
30. Abbasimehr, H.; Shabani, M.; Yousefi, M. An optimized model using LSTM network for demand forecasting. *Comput. Ind. Eng.* **2020**, *143*, 106435.
31. Raju, H.; Das, S. CNN-Based Deep Learning Model for Solar Wind Forecasting. *Sol. Phys.* **2021**, *296*, 134.
32. Babalhavaeji, A.; Radmanesh, M.; Jalili, M.; et al. Photovoltaic generation forecasting using convolutional and recurrent neural networks. *Energy Rep.* **2023**, *9*, 119–123.
33. Zhao, Q.; Wen, X.; Boyan, H.; et al. Optimised extreme gradient boosting model for short term electric load demand forecasting of regional grid system. *Sci. Rep.* **2022**, *12*, 19282.
34. Xiong, X.; Guo, X.; Zeng, P.; et al. A Short-Term Wind Power Forecast Method via XGBoost Hyper-Parameters Optimization. *Front. Energy Res.* **2022**, *10*, 905155.
35. Hewamalage, H.; Bergmeir, C.; Bandara, K. Recurrent Neural Networks for Time Series Forecasting: Current status and future directions. *Int. J. Forecast.* **2021**, *37*, 388–427.
36. Wu, H.; Levinson, D. The ensemble approach to forecasting: A review and synthesis. *Transp. Res. Part C Emerg. Technol.* **2021**, *132*, 103357.
37. Oprea, S.-V.; Bâra, A. A stacked ensemble forecast for photovoltaic power plants combining deterministic and stochastic methods. *Appl. Soft Comput.* **2023**, *147*, 110781.