Future Technology
Open Access Journal
ISSN 2832-0379

February 2026| Volume 05 | Issue 01 | Pages 303-313

Journal homepage: https://fupubco.com/futech

https://doi.org/10.55670/fpll.futech.5.1.26

Future Publishing LLC

Article

An efficient novel deep learning-based predicting model optimized by an improved DAOA algorithm for microgrid energy management

Ali Q. Almousawi [1]*, Nabil Jalil Aklo [2], Zaid Alhadrawi [3]

[1]Electrical Engineering Department, Faculty of Engineering, University of Kufa, Iraq
[2]Department of Biomedical Engineering, College of Engineering, University of Thi-Qar, Thi-Qar,64001, Iraq

ARTICLE INFO

Article history:
Received 29 August 2025
Received in revised form
20 October 2025
Accepted 04 December 2025

Keywords:
Energy management, LSTM, Improved Dynamic Arithmetic Optimization, Photovoltaic, Wind Turbine, Deep learning

*Corresponding author
Email address:
ali.almousawi@uokufa.edu.iq

DOI: 10.55670/fpll.futech.5.1.26

ABSTRACT

This paper presents robust energy-demand and renewable power forecasts for the microgrid using deep learning-based forecasting and a metaheuristic-based optimization model. A Long Short-Term Memory (LSTM) is used to model the temporal nonlinear dynamics of the energy datasets. A new Improved Dynamic Arithmetic Optimization Algorithm (IDAOA) is developed to fine-tune LSTM parameters, incorporating inertial weights, a mutation factor, and the triangle mutation operator to balance exploration and exploitation. The model's performance is verified on various datasets, including wind turbines (WT), photovoltaic (PV) systems, load demands, and day-ahead electricity pricing. This work shows that the IDAOA-LSTM model outperforms other strategies. Practically, the Root Mean Squared Error (RMSE) was 0.021 in the forecast of WT power and 0.031 in the case of PV power. The model performs well in predictions, with high coefficient of determination (R^2) values ($R^2 \geq 0.98$) throughout all tasks. These findings strengthen the applicability of the proposed method to enhance energy-saving measures while preserving the stable operation of those microgrid (MG) systems.

1. Introduction

Examples of renewable energy generation systems that have recently gained popularity are Wind turbines (WT) and photovoltaic (PV) systems [1]. This integration into microgrid (MG) systems has the potential to reduce dependence on fossil fuels and their associated greenhouse gas emissions, resulting in a more secure electricity supply and a cleaner environment. However, renewable energy sources exhibit irregular, non-continuous power generation, as well as load demand patterns, which must be addressed to schedule and maintain grid stability [2,3]. Forecasting renewable energy in addition to load demand is essential for MG operations, cost control, and energy storage scheduling [4]. Conventional approaches for predicting MG uncertainty may be ineffective at capturing the temporal characteristics and nonlinearities inherent in renewable energy supplies, as MGs become more complex when multiple energy sources are added [5]. Therefore, more complex models that adequately capture the stochastic nature of renewable energy generation are urgently needed [6,7]. Because of its ability to extract hierarchical features independently and model nonlinear relationships, deep learning approaches have emerged as a viable alternative for processing complex processes [8]. Despite their potential, many fundamental challenges remain unresolved in the literature, particularly in hyperparameter selection and tuning, which are critical to achieving optimal model performance. As a result, ongoing research investigates effective strategies for hyperparameter modification that enhance the robustness, precision, and generalization capabilities of deep learning models [9]. Although deep neural networks, particularly LSTMs, have demonstrated promising results in sequence prediction, their effectiveness depends heavily on the selection of key hyperparameters, such as the number of hidden layers [10]. Most optimization algorithms suffer from premature convergence or limited search capabilities in the global search space, which significantly affects the fine-tuning of complex models [11]. To do this, this study developed a technique that combines LSTM networks with an Improved Dynamic Arithmetic Optimization Algorithm (IDAOA) to select hyperparameters efficiently. The primary goal is to develop a reliable, accurate, and efficient forecasting model for photovoltaic power, wind energy, and load demand in microgrid systems. This means that the proposed methodology in this paper not only seeks to enhance the model accuracy but also to provide real-time energy management in dynamic systems.

Abbreviations

LSTM	Long Short-Term Memory
IDAOA	Improved Dynamic Arithmetic Optimization Algorithm
PV	Photovoltaic
WT	Wind Turbines
RMSE	Root Mean Squared Error
MG	Microgrid
ANN	Artificial-Neural Network
ABMO	Adaptive Barnacle-Mating Optimizer
SVR	Support Vector Regression
GAM	Global Attention Mechanism

The rest of the paper is structured as follows: Section 2 reviews related works on forecasting and optimization methods for renewable energy. Section 3 describes the proposed technique, data preprocessing, the LSTM model structure, and IDAOA optimization. Section 4 discusses the outcomes and performance evaluations, as well as the superiority of the proposed approach. The work is summarized in Section 5, which focuses on major results and suggests potential directions for future research.

2. Related works

Many researchers throughout the world have investigated various strategies and techniques linked to this work. Zarma et al. [12] proposed energy demand forecasting models for hybrid MG energy management, using five algorithms: linear regression, random forest, artificial neural network, extreme gradient boosting, and support vector regression. The study sought to investigate many parameters, including irradiance, temperature, time of day, humidity, and season. These elements specify the performance of generators, grid systems, and photovoltaics. Cavus et al. [13] presented a new energy management technique known as deep-fuzzy logic control, which integrates LSTM-based conceptual modelling with adaptive fuzzy logic to improve the outcome in a microgrid system connected to the grid. Mahmoudabadi et al. [14] introduced a detailed energy management strategy for scheduling distributed generation systems in both normal and abnormal modes within microgrids connected to a distribution network. This strategy utilized an extreme learning machine model to predict solar and wind power outputs.

The authors in [15] analyzed a networked microgrid system that includes biomass, PV panels, a wind turbine, a battery system, and pumped-hydro storage. The paper presents a hierarchical deep learning energy management method that is implemented under normal conditions, high demand, variation in renewable generation, and the worst weather conditions. Authors in [16] developed microgrids that can operate in parallel with the grid-connected mode and the island mode, utilizing synchronizing controllers for voltage, phase, and frequency stability based on a deep learning control scheme. Mahjoub et al. [17] employ LSTM as a prediction strategy in the microgrid energy management, which comprises a PV system, a permanent magnet generator, a wind turbine, and a battery system, to track energy generation and power flow. The authors in [18] suggest short-term load prediction in a hybrid system using two approaches: an Artificial Neural Network (ANN) and an Adaptive Barnacle-Mating Optimizer (ABMO) by selecting and adjusting the key features of parameters to increase the efficiency of prediction, while the job of [19] is to predict output energy from wind turbines and PV in a microgrid, where the authors apply Support Vector Regression (SVR) to increase the accuracy and efficiency of power prediction. The proposed method is compared with a linear regression model to minimize error variance, using historical data on weather, energy use, and a dynamic grid environment. The authors in [20] proposed to predict three main factors in a microgrid: next-day energy prices, energy demand, and generation capacity, using intelligent forecasting based on deep learning. An LSTM network is proposed in conjunction with a global attention mechanism (GAM) and genetic algorithm–adaptive weight particle swarm optimization (GA-AWPSO) to maximize prediction accuracy. Alabi et al. [21] put forward a deep learning-based optimization approach for the day-ahead scheduling of ZCMES-VPPs. Unlike base model approaches, the proposed model incorporates CCS to address emissions and EV flexibility, and features a CEM to ensure system reliability. It should be clearly mentioned that the objectives of the paper are as follows:

- An efficient deep learning and metaheuristic optimization-based framework is introduced that leverages advanced exploratory capabilities to identify and configure the LSTM neural network parameters optimally to maximize the energy prediction performance and management systems in MGs.
- An enhanced dynamic Arithmetic Optimization Algorithm (IDAOA) is proposed, which incorporates a dynamic inertia weight update mechanism along with an adaptive mutation coefficient. This strategy adeptly balances the exploration and exploitation stages, thus reducing the possibility of prematurely converging on local minima.
- A dynamic exploration technique is presented, which combines a dynamic mutation coefficient with a triangle mutation approach. This strategy encourages population variety and improves the capability to search globally by restricting the algorithm from updating candidate solutions only dependent on the proximity of the best local solution at the moment.

3. Methodology

This work aims to create an accurate energy forecast system for effective microgrid energy management. It uses a deep learning framework based on Long Short-Term Memory (LSTM) networks. LSTM is a suitable solution for time-series research since it effectively addresses the disadvantages of classic RNNs by preserving long-term temporal relationships [22, 23]. As a result, LSTMs have achieved excellent predictive accuracy across both sequential and nonlinear datasets. The performance of an LSTM network is highly influenced by its configuration parameters, which are usually selected at random. The size of the hidden layer is an important factor to consider when designing an LSTM. This parameter is optimally set in this work to increase LSTM performance using the proposed Improved Dynamic Arithmetic Optimization Algorithm (IDAOA). Figure 1 shows the suggested methodology.

Our proposed model comprises a sequential process into four principal stages to elicit maximum synergy between its components:

A. Data Preprocessing: This phase gathers pure time-series information on microgrid energy supply and demand. The data is then normalized, allowing the model to learn without scale bias between variables. The last step is to transform the whole normalized data into the input-output sequences expected for training the LSTM network.

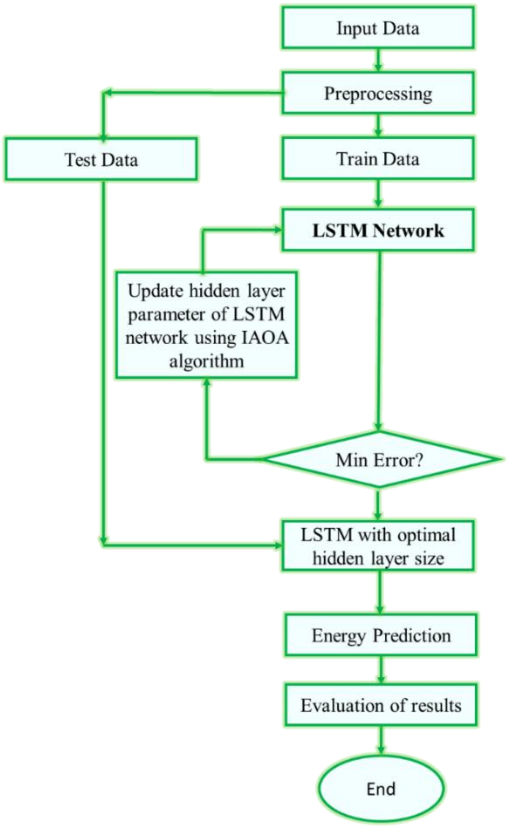

Figure 1. The suggested framework's flowchart

B. Building a Baseline LSTM Model: At this point we have defined and built the baseline architecture of the LSTM neural network. Such decisions entail the design of the underlying LSTM network topology and architecture, the output and input layer sizes via time-series data pre-processing, baseline hyperparameter values prior to optimization, and the choice of activation function. The base topology of this network determines the search space in the IDAOA algorithm.

C. Hyperparameter optimization with IDAOA: during this step, the IDAOA begins searching for the best hyperparameters of the LSTM model. For each cycle in IDAOA, the LSTM model is trained on the training data using a set of hyperparameters. The prediction error(e.g. This process is repeated until the parameters that would give the lowest possible error are obtained.

D. Final modeling and forecasting: Once the optimal hyperparameters are identified and IDAOA converges, the final LSTM model with these optimal parameters is fitted on the entire training dataset. Finally, we run the test dataset through the trained model, which yields the final predictions. These estimated values later guide performance evaluation and integration into the microgrid energy management system.

3.1 Data preprocessing

In this work, data preprocessing is separated into three main steps: data cleaning, normalization, and partitioning.

- **Data cleaning**

Data cleaning is primarily done to remove irregularities from the dataset, such as duplicates, missing items, and invalid entries. Missing values are imputed using the average of the preceding and succeeding values, as expressed in Equation (1):

$$x_i = \frac{x_{i-1}+x_{i+1}}{2} \quad (1)$$

Where x_i, $x_{(i-1)}$, and $x_{(i+1)}$ represent the missing value, the value one hour before, and the value one hour after, respectively.

In our dataset, the missing data rate was very low, and most missing points occurred as single points or within very short time intervals (e.g., 1 or 2 time steps). Under such conditions, where the gap between observed points is minimal, the difference between the output of simple linear interpolation and more complex techniques such as spline interpolation or KNN imputation is negligible in terms of their impact on the final model accuracy. In other words, nonlinear effects over such short time intervals are of lesser importance. In this case, linear interpolation gives a sufficient and consistent local estimate while improving the code's simplicity and efficiency.

- **Data normalization**

Variables with lower values may have a disproportionately low impact on the prediction model because the numerical ranges of distinct attributes differ. This is solved by translating all feature values to the [0, 1] range, a common technique to improve training convergence and model performance. To ensure reproducibility and consistency, min-max normalization was applied to each feature separately (per-feature scaling). The normalized value was determined for each characteristic x_j, as follows:

$$x_j^{norm} = \frac{x_j - min\,(x_j)}{max(x_j) - min\,(x_j)} \quad (2)$$

The max (x_j) and min(x_j) represent the maximum and minimum values of the j-th feature calculated across the training set. This per-feature min-max scaling avoids features with higher numerical ranges from dominating the learning process, while simultaneously ensuring that all input variables contribute appropriately to model optimization.

- **Data segmentation**

The dataset is divided into two sets: training and testing. The current methodology employs 70% of the data for training and 30% for testing. The testing subset is used to evaluate the performance of the proposed model, while the training subset is used to optimize the suggested LSTM's learning process and identify the appropriate size of its hidden layer. To prevent information leakage and ensure methodological accuracy in time-series prediction, the data's chronological sequence was strictly maintained during segmentation. There was no random shuffling used. The dataset was divided into time intervals, with the first 70% of observations (earliest timestamps) assigned to the training set and the remaining 30% reserved for testing.

3.2 LSTM network

This study uses an LSTM network to estimate energy use because it can efficiently handle temporal dependencies and sudden variations in the data. Due to its ability to preserve long-term dependencies and alleviate the vanishing gradient problem, the LSTM network demonstrates excellent performance across a range of sequence-based tasks.

Figure 2 illustrates the scheme of the LSTM recurrent neural network [24]. The cell is the most important part of the structure of the LSTM. Every cell is equipped with a recurrent unit preserving the input and output side information sequences. An LSTM cell consists of four main parts: the input gate, the output gate, the forget gate, and the cell state. The gates use the sigmoid function to generate activation values ranging from zero to one, indicating how much information is lost, updated, or transmitted to the next state. However, the underlying trainable weights associated with these gates are real-valued parameters that lack such limitations. These weights are learned during training to optimize the network's performance. The input gate is responsible for incorporating new information into the cell state; it selectively admits relevant sections of the new input. The output gate determines the output of the LSTM cell based on the updated cell state, controlling which portions of the state are revealed as output. Additionally, the cell state serves to maintain long-term dependencies and is not modified directly; rather, it is regulated by the operations of the forget, input, and output gates.

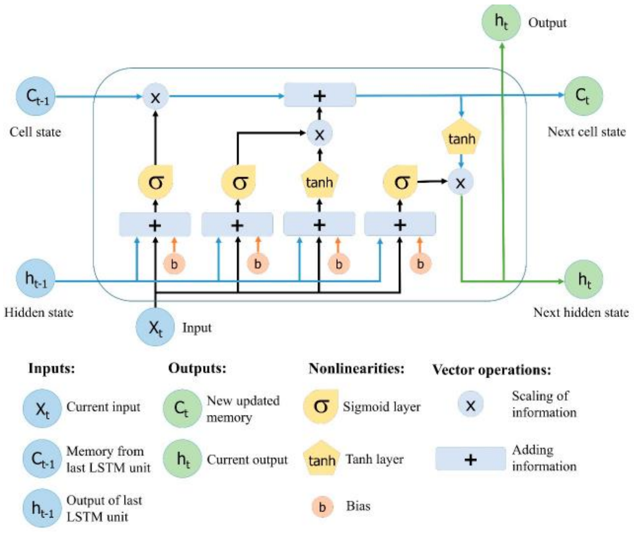

Figure 2. Structure of the LSTM recurrent neural network

3.3 LSTM neural network optimization using IDAOA

The size of its hidden layer heavily affects an LSTM network's learnability, and larger hidden layers can capture overly complex patterns, leading to overfitting. To improve the model's accuracy, IDAOA is recommended to solve this problem. It is particularly well adapted to this task, as its unique structure enables highly efficient sampling of the high-dimensional space and increases the likelihood of finding a near-optimal solution. The key IDAOA steps for optimizing the hidden layer size are summarized here:

- **Step 1: Parameter initialization**

This step comprises initializing the IDAOA's control parameters (α and μ). The parameter α, set to 5, determines the level of exploitation precision in each iteration. With an initial value of 0.5, the parameter μ balances exploration and exploitation equally.

- **Step 2: Generation of initial candidate solutions**

Every IDAOA possible solution X indicates a potential LSTM network configuration. These solutions are iteratively improved after being randomly initialized within the problem's search space. Each solution X encodes a possible hidden layer size, and the algorithm progressively updates them to move toward the optimal configuration. The initial candidate solutions are produced using Eq. (3) with a size of N*n (where N represents the population size and n is the dimension), and during the iterative process, the optimum solution set in each iteration is retained as the current optimal or near-optimal value:

$$X = \begin{bmatrix} x_{1,1} & \cdots & \cdots & x_{1,j} & x_{1,n-1} & x_{1,n-1} \\ x_{2,1} & \cdots & \cdots & x_{2,j} & \cdots & x_{2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N-1,1} & \cdots & \cdots & x_{N-1,j} & \cdots & x_{N-1,n} \\ x_{N,1} & \cdots & \cdots & x_{N,j} & x_{N,n-1} & x_{N,n} \end{bmatrix} \qquad (3)$$

- **Step 3: Fitness function evaluation**

A core aspect of any optimization algorithm is the formulation of an appropriate fitness function (or objective function). In this study, the RMSE is used to evaluate the quality of each candidate solution. For each candidate hidden layer size proposed by the IDAOA, an LSTM model is trained and evaluated on the energy prediction task. The fitness score is then determined using the corresponding RMSE:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_{abs.i} - X_{model.i})^2} \qquad (4)$$

Where N denotes the total count of observations, Xmodel.i is the forecasted value from the LSTM model, and Xobs.i is the actual value.

- **Step 4: Determining the best solution**

Based on the fitness values (calculated in the previous step), the best solution is selected at each iteration. This solution represents the most promising configuration of the hidden layer size at the current stage of the optimization process.

- **Step 5: Updating the math optimizer accelerated function (MOA)**

Before the Arithmetic Optimization Algorithm (AOA) begins its main process, it must decide whether to enter the exploration or exploitation phase. The MOA function is computed using the following equation:

$$MOA(C_{Iter}) = Min + C_{Iter}\left(\frac{Max-Min}{M_{Iter}}\right) \qquad (5)$$

Definition: MOA(CIter) is the MOA function value at iteration t computed by (4). CIter represents the current iteration. Min and Max are the smallest and largest values of the acceleration function.

- **Step 6: Updating the math optimizer probability (MOP)**

The MOP function determines the probability of entering either the exploration or exploitation phase and is defined as:

$$MOP(C_{Iter}+1) = 1 - \frac{C_{Iter}^{\frac{1}{\propto}}}{M_{Iter}^{\frac{1}{\propto}}} \qquad (6)$$

Where MOP(CIter) is the current iteration, α is a control parameter (set in Step 1), and MIter is the maximum count of iterations.

- **Step 7: Updating the positions of the candidate solutions**

For each candidate solution in the population X, three random values r1,r2,r3∈[0,1] are initialized. If r1>MOA, the exploration phase is triggered; otherwise, the exploitation phase is executed.

- **Step 8: Exploration phase**

If r2<0.5, the division operator (D) is applied to update the position; otherwise, the multiplication operator (M) is used. The position update equations in this phase are defined as:

$$x_{i.j}(C_{Iter}+1) = \begin{cases} best(x_i) \div (MOP+\varepsilon) \times (\left(UB_j\right) - LB_i) \times \mu \times LB_i), \; r_2 < 0.5 \\ best(x_j) \div (MOP) \times (\left(UB_j\right) - LB_i) \times \mu \times LB_i), \quad otherwise \end{cases} \tag{7}$$

Where xi.j (CIter+1) is the new position of the j-th dimension of solution I, best (xi) is the best-known solution so far, UBj and LBj are the upper and lower bounds of dimension j, ε is a small positive number to prevent division by zero.

- **Step 9: Exploitation phase**

If r3 is less than 0.5, the subtraction operation (S) is applied to update the position; otherwise, the addition operation (A) is used. The update rules for this phase are:

$$x_{i.j}(C_{Iter}+1) = \begin{cases} best(x_j) \div (MOP) \times (\left(UB_j\right) - LB_i) \times \mu \times LB_i), \; r_3 < 0.5 \\ best(x_j) \div (MOP) \times (\left(UB_j\right) - LB_i) \times \mu \times LB_i), \quad otherwise \end{cases} \tag{8}$$

- **Step 10: Updating the dynamic inertia weights**

The Arithmetic Optimization Algorithm (AOA) often suffers from local optima and slow convergence, mainly because it relies on the global best solution to update candidate positions. To address this limitation, IDAOA is employed to incorporate dynamic inertia weights, thereby accelerating AOA convergence. The rate at which the solutions are updated throughout each optimization step is directly controlled by the inertia weight. Greater exploration is enabled in the early stages of the optimization process by using larger inertia weights, which cause high-potential solutions within the search space to evolve more slowly. On the other hand, solutions might move more finely within a smaller area as the inertia weights are decreased later. To put it another way, exploration is facilitated by bigger inertia weights, whereas exploitation is supported by smaller weights. To enhance IDAOA's search effectiveness and accelerate convergence, this work proposes a dynamic inertia weight mechanism that reduces nonlinearity with increasing iterations. The dynamic inertia weight computation is shown in Eq. (9):

$$w(t) = c \times w_{begin}\left(\frac{w_{begin}}{w_{end}}\right)^{\frac{1}{(\frac{1+t}{T})}} \tag{9}$$

In this equation, wbegin and wend are the maximum and minimum weights, and c is a randomly generated coefficient that dynamically varies around 1. Here, t refers to the current iteration, and T denotes the maximum number of iterations. By integrating the dynamic inertia weights into the position updating equations of the AOA, Equations (7) and (8) are respectively reformulated as Equations (10) and (11):

$$x_{i.j}(C_{Iter}+1) = \begin{cases} w(t) \times best\ (x_i) \div (MOP + \in) \times \left(\left(UB_j\ \right) - LB_i\right) \times \mu \times LB_i), \; r_2 < 0.5 \\ w(t) \times best\ \left(x_j\right) \times MOP \times \left(\left(UB_j\ \right) - LB_i\right) \times \mu \times LB_i), otherwise \end{cases} \tag{10}$$

$$x_{i.j}(C_{Iter}+1) = \begin{cases} w(t) \times best\ \left(x_j\right) - MOP \times \left(\left(UB_j\ \right) - LB_j\right) \times \mu + LB_j), \quad r3 < 0.5 \\ w(t) \times best\ \left(x_j\right) + MOP \times \left(\left(UB_j\ \right) - LB_j\right) \times \mu \times LB_j), \quad otherwise \end{cases} \tag{11}$$

- **Step 11: Dynamic mutation**

In this study, a dynamic coefficient mutation is introduced, which grows as the number of iterations progresses. Through this strategy, the algorithm's ability to avoid local optima is enhanced by increasing the probability that parts of the population will explore alternative search areas. The following equation computes the mutation coefficient:

$$p = 0.2 + 0.5 \times \frac{t}{T} \tag{12}$$

where p is the mutation probability coefficient, which is gradually increased as t repetitions are carried out. Three solutions are chosen at random and merged in this strategy by the following function:

$$X(t) = \frac{X_{r1}+X_{r2}+X_{r3}}{3} + (p2 - p1) \times (X_{r1} - X_{r2}) + (p3 - p2) \times (X_{r2} - X_{r3}) + (p1 - p3) \times (X_{r3} - X_{r1}) \tag{13}$$

where Xr1, Xr2 and Xr3 are three solutions selected randomly by (p2-p1), (p1-p3) and (p1-p3) respectively, and computed as follows:

$$p1 = \frac{|f(X_{r1})|}{\acute{p}} \tag{14}$$

$$p2 = \frac{|f(X_{r2})|}{\acute{p}} \tag{15}$$

Here, f() is the fitness function, and p´ is defined as follows:

$$\acute{p} = |f(X_{r1})| + |f(X_{r2})| + |f(X_{r3})| \tag{16}$$

The triangular mutation technique facilitates the generation of datasets from a randomly chosen pattern while maintaining updated solutions at the local optimum. Therefore, triangular mutation increases the capability of the algorithm to escape local optimal points.

- **Step 12: Stopping criterion**

Steps 3 to 11 are repeated until the stopping condition is met or the maximum number of iterations is reached.

4. Results and discussion

The outcomes presented in this section aim to evaluate the effectiveness and efficiency of the proposed algorithm in predicting the four main factors: PV, WT, day-ahead price, and load, respectively. The ultimate aim of this strategy is to balance demand with supply, so that grid stress is minimized, energy prices drop, and end-use consumer satisfaction improves. Additionally, forecasting the above four aspects can provide clear prospects for the system in the medium and long term and be beneficial for constructing a sound management system that uses renewables on the supply-demand side to moderate the system's uncertainty or randomness. To evaluate the proposed model, the computational cost challenge posed by repeated training of the LSTM network within the GA-AWPSO algorithm's fitness function was carefully managed. To achieve an optimal balance between result accuracy and operational runtime, the main parameters of the GA-AWPSO algorithm were set as follows: the population size was set to 15 particles and the iteration number to 20, resulting in 300 fitness evaluations throughout the entire optimization process. It should be noted that the hyperparameter optimization process is performed only once, and after convergence, the optimal parameters are fixed in the model; this approach ensures that during the evaluation (testing) phase, the model is applied only once with the optimized parameters on new data, thereby significantly reducing the testing time.

Simulations were performed using MATLAB on hardware comprising a 10th-generation Intel Core i7 CPU and an NVIDIA GeForce GTX 1660 Ti GPU. Under these conditions, the time required to train the LSTM model with optimized parameters was 25 minutes on average, while the time required for prediction on the entire test dataset was only 3 seconds. Given the limited number of evaluations (300) and the very low runtime during testing, this model is operationally assessed as highly suitable for forecasting applications. To verify the validity and stability of the proposed method, the predicted results will be compared based on three statistical parameters (MSE, RMSE, and R^2). These statistics are critical for comparing real and predicted performance using a model error calculation and for assessing generalization across families of materials. The model's performance is measured at several key points, where R2 indicates the degree of convergence between actual and estimated values, and MSE and RMSE assess the predictive precision. The three parameters are shown in Equations (17) - (19).

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n} (X_{obs.i} - X_{\text{model }.i})^2} \qquad (17)$$

$$MSE = \frac{1}{N}\sum_{i=1}^{n} (X_{obs.i} - X_{\text{model }.i})^2 \qquad (18)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (X_{obs.i} - X_{\text{model }.i})^2}{\sum_{i=1}^{n} (X_{obs.i} - \bar{X}_{obs.i})^2} \qquad (19)$$

Where N is the number of observations, $X_{\text{model }.i}$ is the predicted value, $X_{obs.i}$ is the actual value, $\bar{X}_{obs.i}$ is the mean of the observed values.

4.1 Dataset

To assess the efficiency of the suggested prediction model, a real dataset on microgrid energy management was employed. This dataset covers wind turbines, photovoltaic (PV), and load demand. Data quality was ensured through preprocessing, enabling the model to achieve high accuracy during training. The dataset used for forecasting model evaluation was obtained from the PJM Interconnection database. The data refer to the PJM West Zone [25]. This zone, which includes Western Pennsylvania, the states of Ohio, and West Virginia, is regarded as one of the most important areas for power generation and load management. In total, 800 hours of data (800 samples) were selected for the experiments, corresponding to a 34-day span in 2020. This period lasts from January 1, 2020, to February 3, 2020. The data were split into training and test subgroups. 30% of the total data was allocated to the test set to assess the model's validity during the trials [20]. Before applying our algorithms, the dataset underwent initial data cleaning. This stage identified missing values, which were recovered using a simple linear interpolation method (Eq. 1). Figure 3 provides an overview of the data used in this work. All parameters for the LSTM structure and training requirements are listed in Table 1.

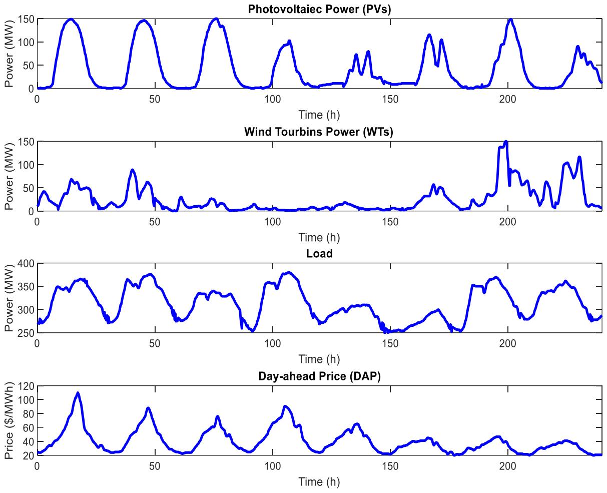

Figure 3. Dataset used in the work

Table 1. The essential parameters for the LSTM Structure

Parameters	Values
Number of hidden units in LSTM layers	100
Dropout rate	0.2
Max Epochs	100
Mini Batch Size	32
Learning Rate	0.001
Activation function	sigmoid
Optimizer type	adam
Gradient Threshold	10
Number of time steps in each input sequence	10

4.2 Evaluation of proposed method performance in the training phase

Figure 4 shows the convergence curve of the proposed method during training. The red curve shows the network loss over iterations, while the blue curve shows the RMSE at each iteration. As shown, the network loss decreases steadily with the number of iterations. The network loss converges after 3000 iterations, at which point the loss variations become minimal. This indicates that the model has successfully captured the patterns.

4.3 Performance evaluation in PV power forecasting

The effectiveness of the suggested approach in predicting the output power of photovoltaic (PV) systems is analyzed in Figure 5. The actual PV power curve (solid blue line) and the prediction curve from the suggested model (dashed red line) are shown in the upper part of the figure. The nearly perfect match between the actual and expected curves demonstrates the model's high accuracy and ability to track dynamic trends and changes in solar power generation over time. Furthermore, the error is limited to the range [-0.1, 0.1], suggesting that the approach is stable and generalizable across a wide range of cases.

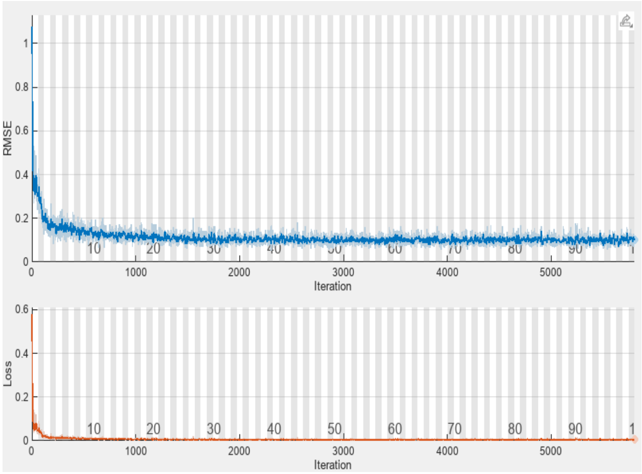

Figure 4. The LSTM convergence curve tuned with IDAOA

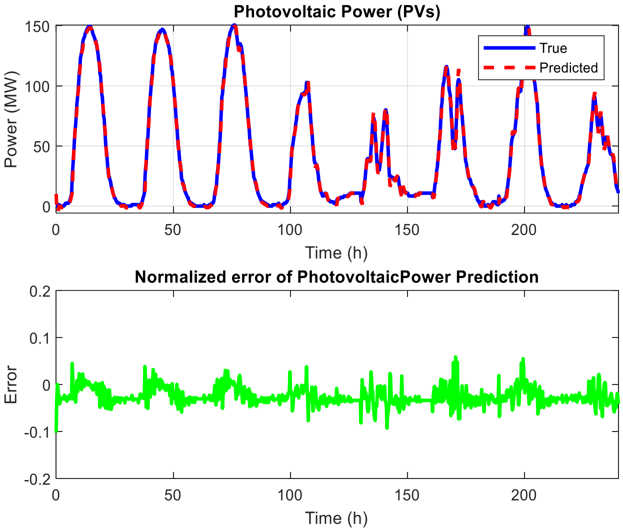

Figure 5. Performance evaluation in PV prediction

The regression indicator between forecast and actual PV power, shown in Figure 6, is used to assess the accuracy of the suggested method. A: This graph shows the x, y coordinates of real value (x) against the output value of the model (y). The black circles show the data points, and the blue line shows the regression line between the two variables. The dashed diagonal line shows the optimal fit line (Y=T). Most of the points lie around the perfect line, and the regression line overlaps this line clearly. This illustrates that the predicted values fit very well with the actual values for our proposed model. Real vs model output form a linear relationship with R = 0.99852, confirming strong linearity. This value, which is close to one, indicates that the learning model did well in producing a linear function that accurately relates the inputs to the target output. Thus, the curves in Figure 6 prove that the prediction of PV production using the proposed method behaves very closely to the actual curve, and it can confidently be used in renewable energy-based energy management systems.

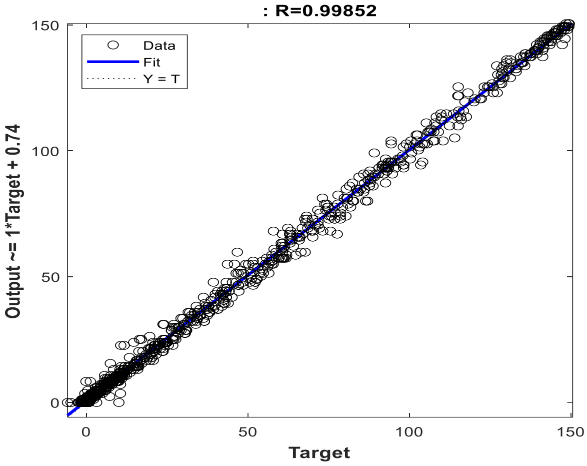

Figure 6. Regression plot comparing the actual and forecasted PV power values

4.4 Performance evaluation in wind turbine (WT) power prediction

The results of wind turbine (WT) output power forecasting using the model described are shown in Figure 7. The top half of the plot shows the expected values (dashed red line) and the actual power (solid blue line). As shown, the proposed model captures the complex temporal-spatial structures and seasonal variations in wind power generation across different timescales. The model's hallmark is its ability to detect rapid, nonlinear variations in delivered power. Energy systems based on wind, which can be quite variable, depend heavily on this. The bottom half of the figure shows the normalized prediction error over time. Global error distribution is narrow. Most of the data is between ±0.2. This shows the model's consistency and reliability during evaluation. The lack of clear systematic errors indicates proper generalization and no overfitting on the training dataset.

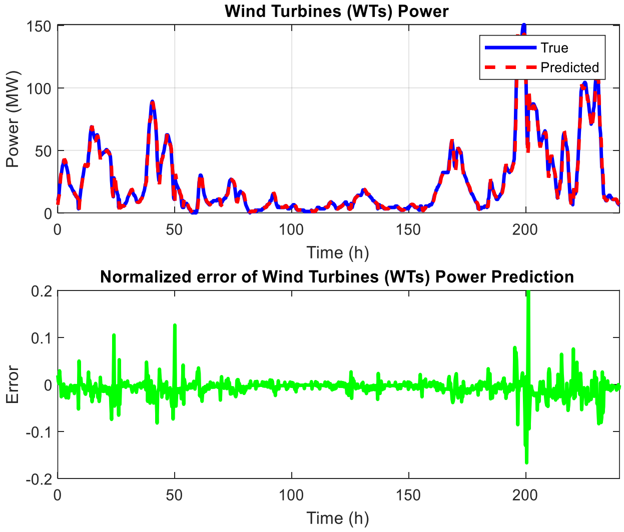

Figure 7. Evaluation of the WTs' power prediction using the proposed model

Expected and real wind turbine output power values are presented in Figure 8. The data points (black circles) are close to the Y = T line, showing that the model output represents well real values. The slope of the regression line (blue) is near one, and the intercept is small (around -0.21). We can tell that the recommended model performs uniformly through the spectrum of target values due to the high density of points around the ideal line and the lack of significant point scattering on high and low value ranges. This is particularly useful in applications such as wind power planning, where it is important to have an accurate estimate of performance across the full range of operation. The fact that the model accurately reproduces wind production on this dimension, as shown by the close match between actual values and projections, indicates its structural accuracy and its ability to effectively portray variability among wind production.

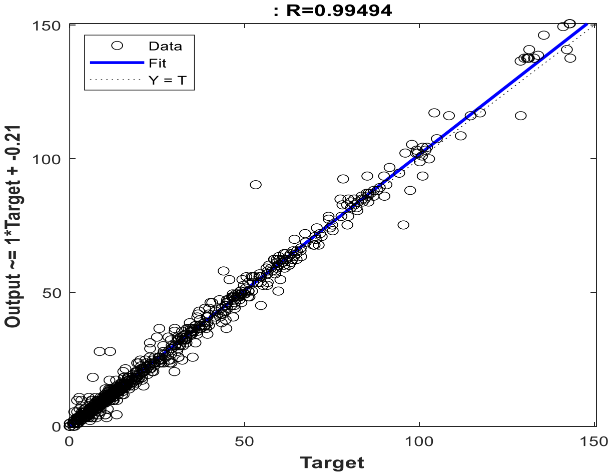

Figure 8. Regression analysis of the WT output power

4.5 Performance evaluation in load power prediction

The performance of the proposed method for load demand forecasting is shown in Figure 9. The dashed red line corresponds to estimated quantities, whereas the solid blue line displays real data; their correlation is obviously significant, as shown by the curves shown earlier. It has enabled us to observe temporal variations in electricity demand, especially at load troughs and peaks. Reflecting this, the model can capture short-term fluctuations and cyclical consumption patterns, indicating that it has great potential to identify time-related and nonlinear characteristics of load signals. The normalized error between the predicted and actual quantities is displayed in the lower portion of the figure. There is no evident pattern of systematic departure in the error distribution, which is uniform and roughly symmetric around the zero axis. This indicates there is no systemic bias in the model.

Figure 10 examines the association between the actual and predicted load. With a correlation coefficient of R = 0.99634, the linear regression line fits the data very well, demonstrating a high degree of agreement between the predicted and actual values. The large R-value indicates the model's ability to faithfully capture complex consumption behavior. In general, the regression line indicates that the proposed model has strong reliability for practical load demand control and performs well across a wide range of load demand from real data, with strong statistical coherence with the real data.

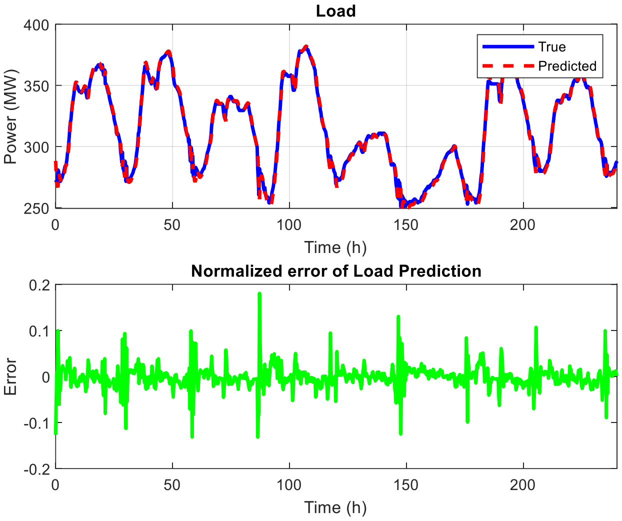

Figure 9. Performance of the proposed method for load prediction

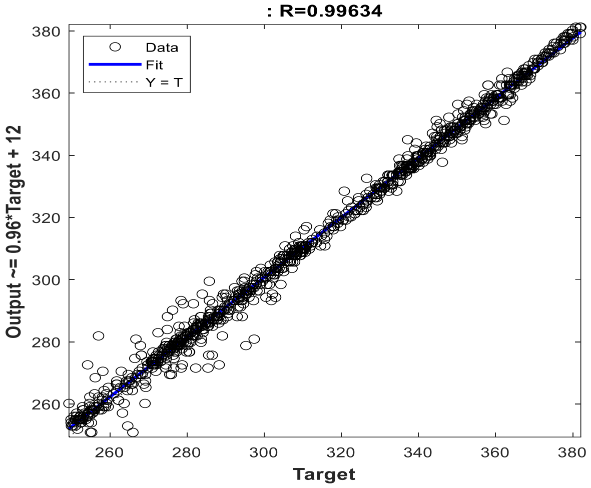

Figure 10. Regression analysis of the actual and predicted load

4.6 Performance evaluation in day-ahead price (DAP) forecasting

Figure 11 shows that the proposed method is highly effective at forecasting the day-ahead price. By comparing the actual and predicted price curves, it is evident that the system has successfully reconstructed price fluctuations. This is especially important in competitive energy markets with significant price sensitivity, where even small price movements can have exaggerated effects on buying behavior. A second important characteristic that illustrates the model's potential to learn complex economic and temporal features is its ability to capture both nonlinear dynamics and periodic behavior in price time series. The normalized error (forecast vs. actual price) is shown in the lower part of the figure. The low amplitude and symmetric spread of errors support the idea that the model is stable and not biased toward any price level, while showing low-amplitude errors, especially during price-making price fluctuations. As seen in Figure 12, the comparison is the day-ahead energy cost versus system production. The correlation coefficient (R = 0.99863) of the regression line we computed is almost perfect, which, as seen below, shows that we can accurately replicate the price levels we see in the real world. In other words, the model predicts absolute numbers and also the overall tendency. The model's predictions show no apparent systematic bias, as evidenced by the regression curve's intercept of around 0.0037, which is small in the context of energy market prices. Real-time and operational forecasting situations benefit significantly from this level of accuracy in estimating variable pricing values. In general, the analysis of Figure 11 and Figure 12 shows that the suggested model is a viable tool for maximizing electricity trading strategies in day-ahead markets since it produces extremely accurate predictions, successfully adjusts to market volatility, and maintains an impartial behavior.

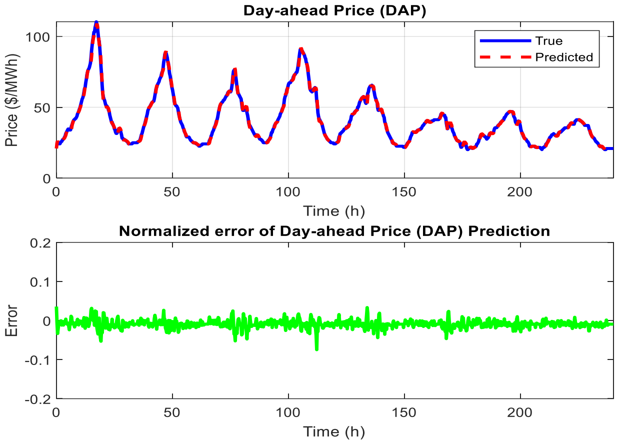

Figure 11. The proposed model's DAP effectiveness

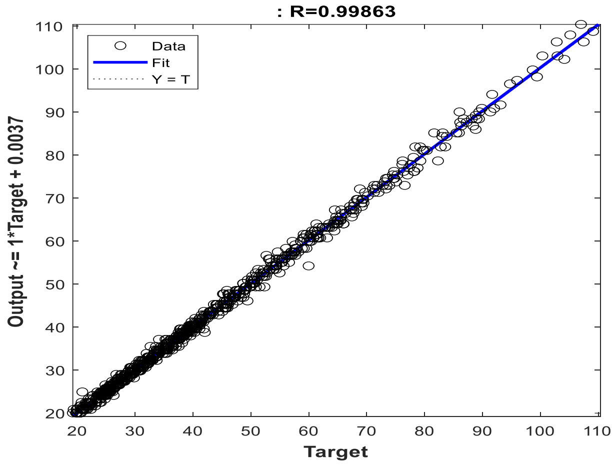

Figure 12. The regression plot between the DAP's actual and predicted values

4.7 Comparative evaluation of prediction methods

Table 2 provides a quantitative comparison of the suggested method, IDAOA-LSTM, with other comparable state-of-the-art models for PV and WT power prediction. Three common metrics are used to evaluate the performance: RMSE, MSE, and R^2. The suggested model much outperforms the best previous approach, GA-AWPSO-LSTM-GAM, which produced an RMSE of 0.055 and an MSE of 0.031, in PV power prediction, achieving an RMSE of 0.031 and an MSE of 0.0009. Additionally, the model's excellent ability to correctly reproduce fluctuations in PV production is confirmed by an R^2 of 0.98. Again, the suggested model outperforms the best reference model in wind power forecasting, with RMSE = 0.021 and MSE = 0.0004 as compared to 0.064 and 0.037. In this case, the model's robustness and strong relationship over the whole range of real wind power outputs are supported by the R^2 value of 0.98. The simultaneous improvement across all three measures in both types of renewable energy sources is an essential result from Table 2, showing that the IDAOA-LSTM model retains excellent statistical alignment while simultaneously lowering numerical prediction errors. These enhancements are the result of the LSTM architecture's intelligent layout and parameter optimization via the IDAOA approach, which enables it to learn complex, nonlinear patterns in renewable energy.

Table 2. Performance evaluation for forecasting renewable energy (PV and WT Power)

Method	PVs			WTs		
	RMSE	MSE	R^2	RMSE	MSE	R^2
LSTM[20]	0.099	0.065	0.85	0.112	0.077	0.80
LSTM-GAM[20]	0.071	0.047	0.89	0.083	0.051	0.83
PSO-LSTM[20]	0.066	0.042	0.91	0.073	0.045	0.86
GA-AWPSO-LSTM[20]	0.062	0.039	0.93	0.071	0.042	0.88
GRU-BiLSTM[21]	0.058	0.034	0.94	0.067	0.039	0.89
GAN[21]	0.057	0.033	0.94	0.069	0.040	0.89
GA-AWPSO-LSTM-GAM[20]	0.055	0.031	0.96	0.064	0.037	0.91
IDAOA-LSTM(proposed)	0.031	0.0009	0.98	0.021	0.0004	0.98

Table 3. Performance analysis of MG, load, and DAP

Method	Load			DAP		
	RMSE	MSE	R^2	RMSE	MSE	R^2
LSTM[20]	0.058	0.043	0.84	0.066	0.047	0.85
LSTM-GAM[20]	0.67	0.049	0.86	0.059	0.042	0.88
PSO-LSTM[20]	0.060	0.045	0.91	0.058	0.042	0.91
GA-AWPSO-LSTM[20]	0.043	0.033	0.84	0.052	0.040	0.93
GRU-BiLSTM[21]	0.035	0.024	0.96	0.049	0.037	0.93
GAN[21]	0.039	0.030	0.95	0.041	0.033	0.94
GA-AWPSO-LSTM-GAM[20]	0.029	0.021	0.97	0.039	0.028	0.95
IDAOA-LSTM(proposed)	0.024	0.0005	0.99	0.013	0.0001	0.99

The IDAOA-LSTM approach's prediction ability on two additional key parameters, load demand and DAP forecast, is studied in Table 3. This comparison uses the same metrics as benchmark approaches: R^2, MSE, and RMSE. The suggested method accurately predicts load patterns, with an RMSE of just 0.024, an R^2 of 0.99, and an extraordinarily low MSE of 0.0005. These results outperform even the best-performing models, such as GA-AWPSO-LSTM-GAM (RMSE = 0.029, R^2 = 0.97). The prediction of DAP by IDAOA-LSTM has a high

R^2=0.99, RMSE=0.013, and MSE=0.0001 against the best benchmark model (GA-AWPSO-LSTM-GAM) that achieves an RMSE of 0.039 and R^2 = 0.95, showing a significant improvement. The R^2 value, which is close to 1, indicates that the proposed model's fit to the actual data is generally high. The small RMSE and MSE values, as shown in the table, exhibit how efficiently it mitigates prediction error. With these characteristics, IDAOA-LSTM is a reliable and practical model that can be applied in intelligent energy management systems.

5. Conclusion

In this paper, a new fusion model is proposed, which can improve the energy prediction accuracy of a microgrid by combining an IDAOA algorithm and an LSTM network. Essentially, modeling temporal data should exploit the strengths of deep learning. The adaptive determination of hyperparameters, including the number of nodes in the hidden layer, is also done via IDAOA. The proposed IDAOA-LSTM model demonstrated high forecasting accuracy across key energy parameters in microgrid systems. For photovoltaic (PV) power prediction, the model achieved an RMSE of 0.031, and for wind turbine (WT) output, an RMSE of 0.021. In forecasting electrical load demand and DAP, the model achieved RMSE values of 0.024 and 0.013, respectively. Across all prediction tasks, the model consistently achieved coefficient of determination (R^2) values of 0.98 or higher, indicating strong alignment between predicted and actual values. These results corroborated the utility of the method developed herein for simulating and quantifying the nonlinear energy response of a structure in order to achieve accurate energy control decisions. The findings indicate potential for integrating deep learning architectures that predict challenges in microgrid and renewable-oriented problems into metaheuristic optimization at the upper level.

Ethical issue

The authors are aware of and comply with best practices in publication ethics, specifically regarding authorship (avoidance of guest authorship), dual submission, manipulation of figures, competing interests, and compliance with research ethics policies. The authors adhere to publication requirements that the submitted work is original and has not been published elsewhere.

Data availability statement

The manuscript contains all the data. However, more data will be available upon request from the authors.

Conflict of interest

The authors declare no potential conflict of interest.

References

[1] Khalid, M, ‘Smart grids and renewable energy systems: Perspectives and grid integration challenges’ Energy Strategy Reviews, 51, 101299, 2024. DOI: https://doi.org/10.1016/j.esr.2024.101299

[2] Nassar, Y. F., El-Khozondar, H. J., & Fakher, M. A. ‘The role of hybrid renewable energy systems in covering power shortages in public electricity grid: An economic, environmental and technical optimization analysis’, Journal of Energy Storage, 108, 115224, 2025. DOI: https://doi.org/10.1016/j.prime.2024.100887

[3] Thirunavukkarasu, G. S., Seyedmahmoudian, M., Jamei, E., Horan, B., Mekhilef, S., & Stojcevski, A. ‘Role of optimization techniques in microgrid energy management systems—A review’, Energy Strategy Reviews, 43, 100899, 2022. DOI: https://doi.org/10.1016/j.esr.2022.100899

[4] Teixeira, R., Cerveira, A., Pires, E. J. S., & Baptista, J. ‘Advancing renewable energy forecasting: A comprehensive review of renewable energy forecasting methods’, Energies, 17(14), 3480, 2024. DOI: https://doi.org/10.3390/en17143480

[5] González-Niño, M. E., Sierra-Herrera, O. H., Pineda-Muñoz, W. A., Muñoz-Galeano, N., & López-Lezama, J. M. ‘Exploring Technology Trends and Future Directions for Optimized Energy Management in Microgrids’, Information, 16(3), 183, 2025. DOI: https://doi.org/10.3390/info16030183

[6] Aslam, S., Herodotou, H., Mohsin, S. M., Javaid, N., Ashraf, N., & Aslam, S. ‘A survey on deep learning methods for power load and renewable energy forecasting in smart microgrids’, Renewable and Sustainable Energy Reviews, 144, 110992, 2021. DOI: https://doi.org/10.1016/j.rser.2021.110992

[7] Benti, N. E., Chaka, M. D., & Semie, A. G. ‘Forecasting renewable energy generation with machine learning and deep learning: Current advances and future prospects’, Sustainability, 15(9), 7087, 2023. DOI: https://doi.org/10.3390/su15097087

[8] Hazim Obaid, Z., Mirzaei, B., & Darroudi, A. ‘An efficient automatic modulation recognition using time–frequency information based on hybrid deep learning and bagging approach’, Knowledge and Information Systems, 66(4), 2607-2624, 2024. DOI: https://doi.org/10.1007/s10115-023-02041-y

[9] Wazirali, R., Yaghoubi, E., Abujazar, M. S. S., Ahmad, R., & Vakili, A. H. ‘State-of-the-art review on energy and load forecasting in microgrids using artificial neural networks, machine learning, and deep learning techniques’, Electric power systems research, 225, 109792, 2023. DOI: https://doi.org/10.1016/j.epsr.2023.109792

[10] Tayab, U. B., Yang, F., Metwally, A. S. M., & Lu, J. ‘Solar photovoltaic power forecasting for microgrid energy management system using an ensemble forecasting strategy’, Energy Sources, Part A: Recovery, Utilization, and Environmental Effects, 44(4), 10045-10070, 2022. DOI: https://doi.org/10.1080/15567036.2022.2143945

[11] Joshi, A., Capezza, S., Alhaji, A., & Chow, M. Y. ‘Survey on AI and machine learning techniques for microgrid energy management systems’, IEEE/CAA Journal of Automatica Sinica, 10(7), 1513-1529, 2023. DOI: 10.1109/JAS.2023.123657

[12] Zarma, T. A., Ali, E., Galadima, A. A., Karataev, T., & Usman, S. ‘Energy Demand Forecasting for Hybrid Microgrid Systems Using Machine Learning Models’, Proceedings of Engineering and Technology Innovation, 29, 68-83, 2025. DOI: https://doi.org/10.46604/peti.2024.14098

[13] Cavus, M., Dissanayake, D., & Bell, M. ‘Deep-Fuzzy Logic Control for Optimal Energy Management: A Predictive and Adaptive Framework for Grid-Connected

Microgrids’, Energies, 18(4), 995, 2025. DOI: https://doi.org/10.3390/en18040995

[14] Mahmoudabadi, N. D., Khalaj, M., Jafari, D., Herat, A. T., & Ahranjani, P. M. ‘Energy management of microgrid coalitions considering renewable energy prediction based on machine learning algorithms’, AIP Advances, 15(4), 2025. DOI: https://doi.org/10.1063/5.0236597

[15] Khosravi, N., Oubelaid, A., & Belkhier, Y. ‘Energy management in networked microgrids: A comparative study of hierarchical deep learning and predictive analytics techniques’, Energy Conversion and Management: X, 25, 100828, 2025. DOI: https://doi.org/10.1016/j.ecmx.2024.100828

[16] Ashok Babu, P., Mazher Iqbal, J. L., Siva Priyanka, S., Jithender Reddy, M., Sunil Kumar, G., & Ayyasamy, R. ‘Power control and optimization for power loss reduction using deep learning in microgrid systems’, Electric Power Components and Systems, 52(2), 219-232, 2024. DOI: https://doi.org/10.1080/15325008.2023.2217175

[17] Mahjoub, S., Chrifi-Alaoui, L., Drid, S., & Derbel, N. ‘Control and implementation of an energy management strategy for a PV–wind–battery microgrid based on an intelligent prediction algorithm of energy production’, Energies, 16(4), 1883, 2023. DOI: https://doi.org/10.3390/en16041883

[18] Indira, G., Bhavani, M., Brinda, R., & Zahira, R. ‘Electricity load demand prediction for microgrid energy management system using hybrid adaptive barnacle-mating optimizer with artificial neural network algorithm’, Energy Technology, 12(5), 2301091, 2024. DOI: 10.1002/ente.202301091

[19] R. Singh, A., Kumar, R. S., Bajaj, M., Khadse, C. B., & Zaitsev, I. ‘Machine learning-based energy management and power forecasting in grid-connected microgrids with multiple distributed energy sources’, Scientific Reports, 14(1), 19207, 2024. DOI: https://doi.org/10.1038/s41598-024-70336-3

[20] Kim, H. J., & Kim, M. K. ‘A novel deep learning-based forecasting model optimized by heuristic algorithm for energy management of microgrid’, Applied Energy, 332, 120525, 2023. DOI: https://doi.org/10.1016/j.apenergy.2022.120525

[21] Alabi, T. M., Lu, L., & Yang, Z. ‘Data-driven optimal scheduling of multi-energy system virtual power plant (MEVPP) incorporating carbon capture system (CCS), electric vehicle flexibility, and clean energy marketer (CEM) strategy’, Applied energy, 314, 118997. DOI: https://doi.org/10.1016/j.apenergy.2022.118997

[22] Youyuan Peng, Feng Huang, Xin Xie, Guocai Gui,Fei Zhao, Yuliu Ou, and Hai Xu. ‘A Predictive Approach for Lithium-Ion Battery SOH using LSTM Neural Networks Enhanced by Health Matrix Optimization’, Distributed Generation & Alternative Energy Journal, 39(04), pp. 831–850, 2024. DOI: https://doi.org/10.13052/dgaej2156-3306.3947

[23] Srinivas Singirikonda and Yeddula Pedda Obulesu. ‘Lithium-Ion Battery State of Charge Estimation Using Deep Neural Network’, Distributed Generation & Alternative Energy Journal, 38(03), pp. 761–788, 2023. DOI: https://doi.org/10.13052/dgaej2156-3306.3833

[24] Aklo, N.J., Rashid, M.T. ‘Mitigation of take-or-pay concept drawbacks in hybrid microgrid’, Electrical Engineering, 105(1), pp. 61–75, 2023. DOI: https://doi.org/10.1007/s00202-022-01682-6

[25] PJM Interconnection, “Data Miner 2,” PJM Interconnection LLC. [Online]. Available: https://dataminer2.pjm.com.