Case Studies in Thermal Engineering 72 (2025) 106356

Contents lists available at ScienceDirect

Case Studies in Thermal Engineering

journal homepage: www.elsevier.com/locate/csite

Optimized CNN-LSTM with hybrid metaheuristic approaches for solar radiation forecasting

İrem Fatma Şener [a], İhsan Tuğal [b,*]

[a] Muş Alparslan University, Department of Nuclear Energy and Energy Systems, Muş, Türkiye
[b] Muş Alparslan University, Department of Software Engineering, Muş, Türkiye

ARTICLE INFO

Handling Editor: Dr Y Su

Keywords:
Solar radiation forecasting
CNN-LSTM
Metaheuristic optimization
Regression
Hybrid models

ABSTRACT

The increasing reliance on solar energy has underscored the need for precise forecasting of photovoltaic power outputs, with solar radiation forecasting being a critical factor. This study proposes a novel model for solar radiation forecasting using meteorological and solar radiation data. The performance of several machine learning and deep learning models, including Long Short-Term Memory, Autoregressive Integrated Moving Average, Multilayer Perceptron, Random Forest, XGBoost, Support Vector Regression, and a hybrid CNN-LSTM model, is evaluated for daily solar radiation forecasting. To improve the accuracy of the model, hyperparameter optimization is applied to the CNN-LSTM model using three metaheuristic algorithms: Particle Swarm Optimization, Grey Wolf Optimization, and Starfish Optimization Algorithm. A hybrid ensemble approach is then proposed, integrating the predictions of the three optimized CNN-LSTM models to reduce error and enhance forecasting stability. The results demonstrate that the hybrid model outperforms the individual models, achieving the lowest MAE, MSE, and RMSE while maximizing the R^2 score. The proposed methodology showcases the effectiveness of combining hybrid deep learning with metaheuristic optimization in solar radiation forecasting, offering a robust and adaptable framework for renewable energy applications.

1. Introduction

In recent years, the rapid rise in global energy demand and the adverse environmental impacts of fossil fuels have driven significant changes in energy policies worldwide. The importance of sustainable and clean energy sources has been better understood due to environmental challenges such as air pollution, global warming, and climate change. Optimizing renewable energy resources and formulating sustainable energy policies have become essential. Among renewable resources, solar energy stands out as an eco-friendly, inexhaustible source. Due to its broad geographic availability and limitless potential, solar energy plays a critical role globally. Compared to fossil fuels, solar energy substantially reduces environmental pollution and greenhouse gas emissions. In power generation, it delivers zero carbon emissions, making it a pivotal asset in combating climate change, environmental crises, and natural disasters. Additionally, advances in technology and economies of scale have led to a significant reduction in the costs of solar energy systems in recent years, rendering solar energy more accessible and economically viable.

Maximizing the global utilization of solar energy should be among our primary objectives. Solar energy generation capacity can potentially meet the rising global energy demand. At this stage, strategic planning and measures are essential to ensure uninterrupted

* Corresponding author.
E-mail address: i.tugal@alparslan.edu.tr (İ. Tuğal).

https://doi.org/10.1016/j.csite.2025.106356
Received 16 February 2025; Received in revised form 22 April 2025; Accepted 16 May 2025
Available online 16 May 2025

supply that meets demand effectively. Meeting demand through renewable, clean energy sources and reducing reliance on other resources depend on accurate forecasting and planning. When planning, it is critical to accurately predict the production capacity of both existing and planned solar power plants. Successfully implementing solar energy generation necessitates focusing on the regional solar radiation potential. For solar investments to be efficient, solar radiation forecasts must be made with high accuracy. Predicting solar radiation, however, is challenging due to the direct impact of variable weather conditions. One of the biggest obstacles to the widespread use of solar energy is the unpredictable changes in solar radiation. It varies due to climatic conditions. Therefore, it is considered an unreliable energy source [1]. Artificial intelligence, with its ability to anticipate such anomalies and sudden shifts, can minimize disruptions in solar power generation and enhance the resilience of energy systems. Furthermore, these insights are valuable for analyzing long-term trends related to climate change.

Solar radiation forecasting can be conducted through various methods, primarily utilizing cloud imagery combined with physical models and machine learning techniques [2]. Solar energy potential varies significantly by region, influenced by geographic and climatic factors. Solar radiation in a specific geographic area fluctuates due to atmospheric conditions, temperature, humidity, wind speed, pressure, etc. As solar radiation is influenced by multiple meteorological factors, developing a reliable forecasting model remains a challenging task.

Our study area is located in Türkiye, a region with a significant advantage in terms of solar energy potential. According to the Türkiye Solar Energy Potential Atlas, the country has an average annual sunshine duration of 2741 hours and an average annual total irradiation value of 1527.46 kWh/m^2 [3]. The sunshine duration in the region is quite high throughout the year, which makes Muş attractive for solar energy production. As interest in renewable energy sources increases throughout Türkiye, the importance of solar energy in regions like Muş becomes even more evident [4]. As shown in Fig. 1, Türkiye's southern and eastern regions stand out in terms of annual radiation intensity. Muş Province, the focus of this study, is located in this region and presents suitable potential for solar energy investments. Many renewable energy projects in the areas of solar, wind, and hydraulic have been implemented in Muş. However, to fully harness this potential, accurate solar radiation forecasting in the region is essential.

Accurate regional solar radiation forecasting is crucial for the planning of energy projects and for establishing a balance between energy supply and demand. Reliable and precise predictions play a vital role in guiding energy investments in regions with high solar energy potential, such as Muş Province. This study primarily aims to investigate how artificial intelligence prediction methods perform in predicting daily global solar radiation for Muş, using data from 2018 to 2023. The accuracy and performance of various artificial intelligence methods were compared to identify the most suitable models for solar radiation forecasting.

In this study, Long Short-Term Memory (LSTM) and Autoregressive Integrated Moving Average (ARIMA) time series methods were utilized alongside machine learning methods, including Multi-Layer Perceptron (MLP), Random Forest, Extreme Gradient Boosting (XGBoost), and Support Vector Regression (SVR). Additionally, the Convolutional Neural Network-LSTM (CNN-LSTM) model, which yielded promising results, was used. Moreover, the CNN-LSTM model was optimized using the Starfish Optimization Algorithm (SFOA), Grey Wolf Optimizer (GWO) and Particle Swarm Optimization (PSO) and better results were obtained. Separately, CNN-LSTM models optimized for proper hyperparameter detection with PSO, GWO and SFOA were trained on the same dataset. Each of these models was used to make forecasting on the test dataset. The predictions from all models were averaged to obtain the final prediction. The idea behind averaging is that it helps reduce overfitting and increases the generalization ability of the model. Weighted averaging is an extension of the basic averaging technique. The effect weight of each model on the outcome can be chosen differently. However, in this study, all models were chosen equally because they gave similar results individually. This ensemble learning approach provided improvements and more accurate prediction results in CNN-LSTM by using different optimization methods together.

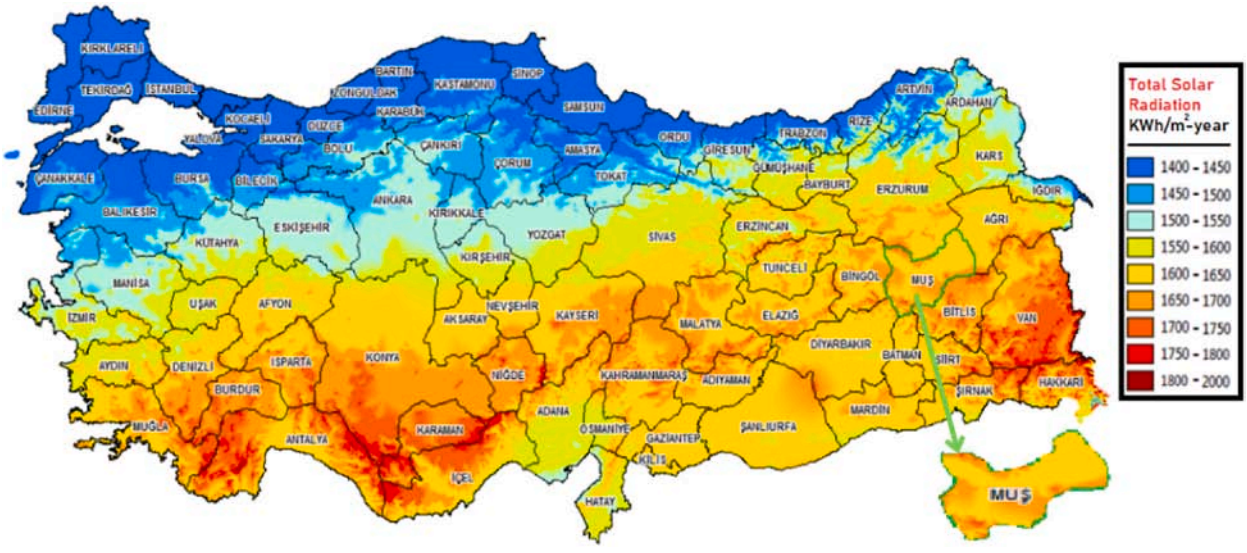

Fig. 1. Total solar radiation of Türkiye with Muş [3].

Time series methods are widely used for forecasting future radiation values based on past data. In time series models, the variable's past observations are analyzed to create a model, which then aims to forecast the variable's future values [5–7]. Time series data can reflect seasonal changes and the effects of weather conditions, which can increase the accuracy of the predictions. In this study, the ARIMA [8] and LSTM [9] models were used as part of the time series approach. While the ARIMA model forecasts based on historical linear data, the LSTM model is particularly effective with complex, nonlinear data due to its ability to learn long-term dependencies in time series [10,11].

Regression-based methods have also shown success in solar radiation prediction. Specifically, regression models such as MLP [12], Random Forest [13], XGBoost [14], and SVR [15] offer flexibility and robustness by utilizing multiple input variables for prediction. In this study, meteorological data (temperature, humidity, wind speed, and pressure) were used to predict solar radiation with these methods.

In various studies, time series, artificial neural networks (ANN), and machine learning methods have been evaluated for regional solar radiation prediction with various error measures [2,16–20]. In some studies, the efficiency of the models has been further improved with various optimization methods [21,22]. Optimization provides better results in determining the model parameters and hyperparameter adjustments. Various ensemble learning models can also give accurate prediction results [23,24]. It can be seen from different studies that successful results have been obtained especially with time series and deep learning methods [25,26]. The results of the studies have demonstrated the effectiveness of ANN in estimating solar energy predictions. Furthermore, incorporating hybrid systems that combine ANN with other algorithms enhances the accuracy of these estimations [27–29]. Natural events such as solar radiation show regular cycles and seasonal changes over a certain period. Time series analysis is ideal for detecting such regularities and trends. Therefore, time series methods have been the ones that have given the best results in many studies [30,31].

Recent advancements in ensemble learning have significantly enhanced energy forecasting accuracy by integrating various deep learning architectures and optimization techniques. A notable study introduced a stacking-based ensemble model combining Long Short-Term Memory (LSTM), Bi-directional LSTM (BiLSTM), and Gated Recurrent Unit (GRU) networks, optimized through a genetic algorithm for feature selection, resulting in improved energy demand forecasting accuracy [32]. In the context of smart cities, a CNN-LSTM model enhanced with the Namib Beetle Optimization (NBO) algorithm was proposed to predict energy consumption. This approach involved balancing the dataset using a generative adversarial network based on game theory and selecting optimal features with the NBO algorithm, leading to more accurate predictions [33]. Another research effort developed a hybrid ensemble deep learning model that integrates MLP, CNN, LSTM, and a hybrid CNN-LSTM architecture. This model demonstrated superior performance in time series energy forecasting tasks [34]. Furthermore, a study focusing on solar energy forecasting examined ensemble deep neural network architectures. The findings indicated enhanced stability and robustness in forecasting under variable and extreme meteorological conditions [35]. Models such as LSTM, GRU and CNN-LSTM have been shown to have significant potential in renewable energy related predictions [36].

This study creates a multidimensional foundation that can be expanded with future studies and also has the potential to contribute to areas such as local energy. The significant innovations and contributions of this study are summarized as follows:

- This study is the combination of time series analysis methods (LSTM, ARIMA) and regression techniques (Random Forest, MLP, SVR, XGBoost), which offer diverse perspectives. Additionally, the CNN-LSTM model, which outperforms these methods, was also utilized.
- PSO, GWO, and SFOA are applied to optimize CNN-LSTM, demonstrating the impact of metaheuristic tuning on predictive performance. The use of optimization algorithms to enhance model performance adds scientific depth and practical applicability to the study. SFOA, a newly proposed algorithm, is utilized for the first time to improve CNN-LSTM.
- A significant improvement was achieved through a novel ensemble learning approach that incorporates three optimization methods. Most notably, the ensemble strategy is not based on conventional averaging or voting mechanisms, but instead on integrating the outputs of CNN-LSTM models that have each been independently optimized using distinct metaheuristic algorithms (PSO, GWO, and SFOA). This method introduces algorithmic diversity into the ensemble, which helps reduce overfitting and increases the generalization capability of the model. Each optimization algorithm guides the CNN-LSTM model to explore different regions of the hyperparameter space, leading to complementary learning behaviors. The final ensemble leverages this diversity, resulting in more robust and stable predictions compared to single-model or homogeneously optimized ensembles. A comparative analysis with various models and approaches was conducted to demonstrate the reliability of the proposed method.
- Due to the differences in solar radiation patterns from region to region, there is a need for region-specific forecasting models. For a region like the Eastern Anatolia Region of Türkiye, which has high solar energy potential but lacks data and research, the application of forecasting models has been an innovative approach. Furthermore, the model is specifically adapted to the regional characteristics of Muş, which is known for its high variability in solar radiation due to geographical and meteorological factors such as altitude, snow cover, and seasonal shifts.
- Unlike many previous studies that apply models to relatively stable or tropical climates, this study focuses on a region with sharp temporal fluctuations and frequent outliers in radiation levels. The integration of localized meteorological features and customized preprocessing allows the model to better capture the non-linear patterns unique to this environment. This region-specific design enhances the model's applicability for operational use in similar high-variability settings, making it both locally relevant and methodologically novel.

The rest of the paper is organized as follows: Section 2 introduces the dataset used in the study and describes the preprocessing steps taken. Section 3 provides detailed explanations of the methods used for forecasting. In Section 4, the prediction results are evaluated.

Section 5 presents the conclusions.

2. Dataset and solar energy

2.1. Data and preprocessing

Meteorological data were obtained from two stations in Muş Central, provided by the Muş Provincial Directorate of Meteorology. These stations were located at Muş/Berce Alparslan Agricultural Enterprise and the Directorate campus. The distance between these stations is around 3 km. Since they are located close to each other, combining the data will not cause any problems. Daily total global solar radiation (kWh/m^2) data were collected from the Muş/Berce Alparslan Agricultural Enterprise, covering values from 2016 to 2023. Additionally, daily average actual pressure (hPa), daily average relative humidity (%), daily average temperature (°C), and daily maximum wind speed (m/s) data were obtained from the Directorate campus, encompassing values from 1964 to 2023. However, no solar-related data was collected at this station. Therefore, it was necessary to combine values from different stations. The solar radiation data for the years 2016 and 2017 contained anomalies; thus, the dataset was created using values from 2018 to 2023. Predictions were made using 2191 days of data. Of these, 80 % were used for training, while 20 % were used for testing. The reason for this partitioning is to evaluate more than one year.

Various factors affect the scope and quality of the dataset used. The collected data was first reorganized to correct any possible errors. There were time intervals during which measurements were not taken due to technical issues with the devices. These missing values were filled in with average values over a long period. The average value for each day of the year was calculated, and the missing data segments were completed with the average values for those days. Due to environmental and technical reasons, there may also be inaccurate measurements in the data. Since it is difficult to identify these erroneous values, they were used as they are. The statistical analysis of the dataset can be seen in Table 1.

2.2. Climate change and solar energy in Muş

The radiation measurement values have exhibited similar behavior over the years, as shown in Fig. 2. The series values exhibit strong seasonality, ensuring their repeatability. Forecasting information is classified according to meteorological conditions into short-term, medium-term, and long-term forecasts. Daily solar radiation forecasts are considered part of short-term forecasting.

The monthly and consequently seasonal radiation distribution based on the values from 2018 to 2023 is shown in Fig. 3. Daily data were taken into account and the total daily time series values for the period 2018–2023 were used as kWh/m^2 for the prediction.

The monthly solar radiation data shows a clear seasonal pattern, with the lowest values in winter (December: 1.50, January: 2.21) and the highest in summer (July: 7.71) as seen in Fig. 4. Solar radiation steadily increases from February to May, peaking in July, followed by a gradual decline through autumn and winter. This trend reflects longer daylight hours and higher solar angles in summer, while winter sees shorter days and reduced solar intensity. The sharp rise from February to May and the symmetrical decline after August indicate strong seasonality. Such patterns are crucial for optimizing solar energy systems, planning agricultural activities, and studying climate dynamics.

However, the winter months show much lower radiation, indicating a need for energy storage solutions or complementary energy sources to maintain energy supply year-round. Overall, Muş demonstrates promising potential for solar energy utilization, especially with appropriate infrastructure to balance seasonal variations.

When looking at whether there is an increasing and decreasing trend in solar radiation values over the years, on average, the increase is approximately 0.0249 units of radiation per year, as seen in Fig. 5. The positive slope suggests that solar radiation has been increasing slightly each year. While this indicates a potential upward trend, the magnitude of the change is relatively small. The R-squared value measures how much of the variability in annual solar radiation data is explained by the trend. In this case, only about 8.12 % of the variability in the data is explained by the trend line, indicating that annual solar radiation levels are largely influenced by other factors (e.g. natural climate variability, local weather conditions). The p-value determines the statistical significance of the trend. A p-value greater than 0.05 (such as 0.5841 in the data) indicates that the observed trend is not statistically significant. This means that the increase in solar radiation may be due to random fluctuations rather than a significant long-term trend. The data shows a slight upward trend in annual solar radiation over the analyzed period, but the trend is weak and not statistically significant. This implies that any observed increase is likely due to natural variability rather than a sustained change attributable to climate change or other long-

Table 1
Statistical values of the variables.

	Humidity	Wind Speed	Temperature	Pressure	Solar Radiation
Number	2191	2191	2191	2191	2191
Average	57.47	5.26	12.16	868.68	4.68
Standard Deviation	23.98	2.52	10.82	4.69	2.46
Minimum	13.5	0.5	−19.6	852.7	0
25 %	34	3.6	3.2	865.2	2.6
50 %	57.2	4.6	12.4	868.6	4.9
75 %	79.5	6.7	21.6	872.1	6.8
Maximum	99.6	18.5	31.6	883	12.6

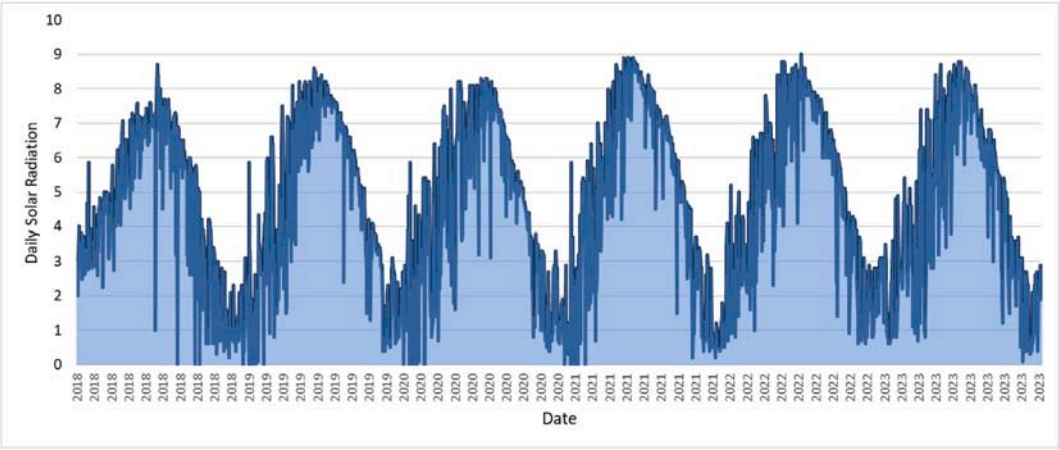

Fig. 2. Total daily solar radiation (kWh/m^2) time series (2018–2023).

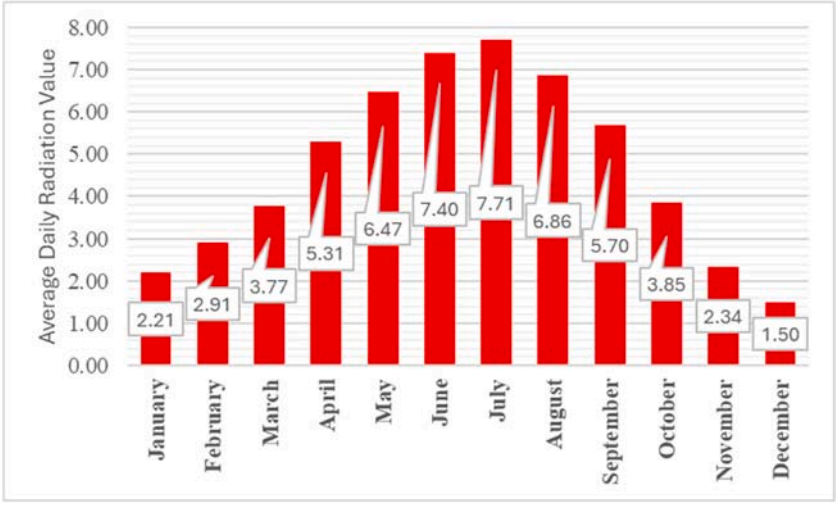

Fig. 3. 2018–2023 average daily global solar radiation in Muş.

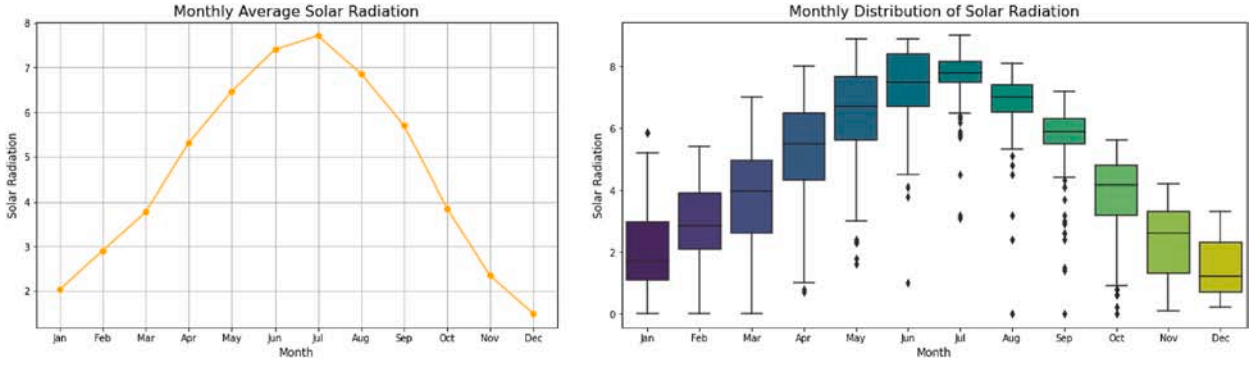

Fig. 4. Monthly solar radiation (2018–2023).

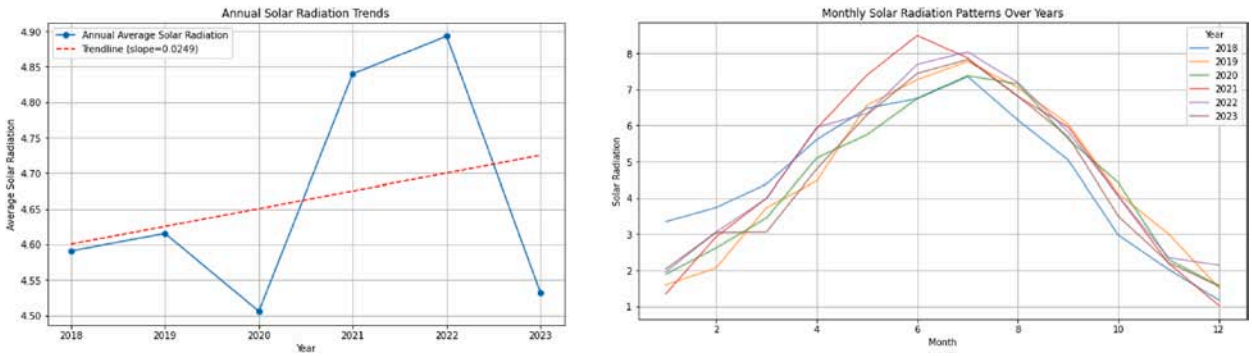

Fig. 5. Annual solar radiation changes (2018–2023).

term factors. Further analysis over a longer period or incorporating additional climate variables (e.g., cloud cover, temperature, aerosols) might provide deeper insights into the potential drivers of this trend.

3. Methods

3.1. Time series models

Time series models are powerful tools for analyzing patterns obtained from past data to predict future trends and values. They play a critical role in fields that generate continuous data, such as finance [37], economics [38], energy [39], healthcare [40], and climate [41]. These models enable more accurate predictions by capturing important features such as seasonal patterns, trends, and auto-correlation in the time-dependent sequential relationships of the data. In this study, the LSTM and ARIMA models used in time series analysis can learn complex and long-term dependencies within the data, while simple linear models cannot fully capture these patterns.

3.1.1. ARIMA

ARIMA model is a powerful method frequently used in time series forecasting. This model was first introduced by Box and Jenkins in 1976 [42]. ARIMA consists of three fundamental components: autoregressive (AR), differencing (I), and moving average (MA) components. Each component is related to the past values of the series, the differencing process for stationarizing the series, and the past values of the error terms, respectively. Its accuracy largely depends on the proper identification of the model's parameters (p,d,q), which represent the order of the AR, I, and MA components, respectively [43–45].

$$x_t = c + \sum_{i=1}^{p} \phi_i x_{t-i} + \varepsilon_t \tag{1}$$

In Eq. (1), x_t is the value of the variable at the current time point, c is a constant term, ϕ_i are the autoregressive coefficients, or model parameters, p is the number of past values to be used, and ε_t represents the noise or error term.

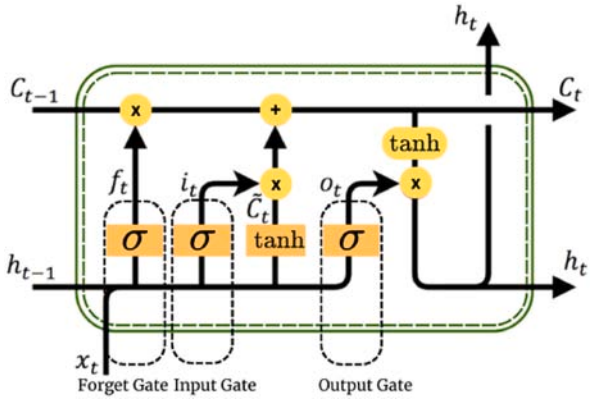

Fig. 6. LSTM model.

$$x_t = \mu + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \tag{2}$$

In Eq. (2), μ represents the mean, θ_i are the moving average coefficients, and q is the number of past error terms. The integration component determines whether the time series is stationary. If it is non-stationary, it is made stationary by differencing to some degree.

3.1.2. LSTM

LSTM is an exceptionally effective model for tasks like time series forecasting, primarily because of its capability to learn long-term dependencies within data sequences. At the core of LSTM's success are the cell state and the three gate mechanisms (forget, input, and output gates). The gate mechanisms, which are fundamental building blocks of the LSTM cell, manage the flow of information within the model. These gates regulate the retention of previous information and the incorporation of new data into the cell [9,46].

As seen in Fig. 6, Forget Gate specifies the proportion of information from the previous cell state that should be discarded. The Input Gate determines the amount of new information to be integrated into the cell state, while the Output Gate controls the extent to which information from the cell state is forwarded to the next time step. By utilizing these three gating mechanisms, LSTM effectively learns long-term dependencies and enhances prediction accuracy by maintaining essential information while filtering out what is no longer relevant [46].

The following equations demonstrate the functions performed by LSTM units.

$$i_t = \sigma(U_i x_t + W_i h_{t-1} + b_i) \tag{3}$$

$$f_t = \sigma\left(U_g x_t + W_g h_{t-1} + b_g\right) \tag{4}$$

$$c_t^* = tan\, h(U_c x_t + W_c h_{t-1} + b_c) \tag{5}$$

$$c_t = g_t \odot c_{t-1} + i_t \odot c_t^* \tag{6}$$

$$o_t = \sigma(U_o x_t + W_o h_{t-1} + b_o) \tag{7}$$

$$h_t = o_t \odot \tanh\,(c_t) \tag{8}$$

In the equations, x_t represents the input, W_* and U_* denote the weight matrices, b_* indicates the bias term vectors, σ refers to the sigmoid function, and $\odot$ is the component-wise multiplication operator [47]. The input gate i_t, along with a second gate c_t^*, controls the new information stored in the memory state c_t at time t. The forget gate f_t checks which past information should be lost or retained in the memory cell at time $t-1$, while the output gate o_t determines which information from the memory cell can be used for the output [46]. The hidden state h_t, which forms the output of the memory cell, is calculated as follows Eq. (8).

In LSTM cells, the sigmoid and tanh functions are often preferred because these functions play an effective role in controlling the flow of information. Both the memory state c_t and the hidden state h_t of each LSTM layer are passed as input to the next LSTM layer. The activation functions used in each cell of the LSTM model control the outputs and the flow of information [46].

3.1.3. CNN-LSTM

Convolutional neural networks (CNN), known for their feature extraction capabilities, have been effectively integrated with LSTM to form CNN-LSTM hybrid models, which leverage the spatial feature extraction power of CNNs and the sequential learning capability

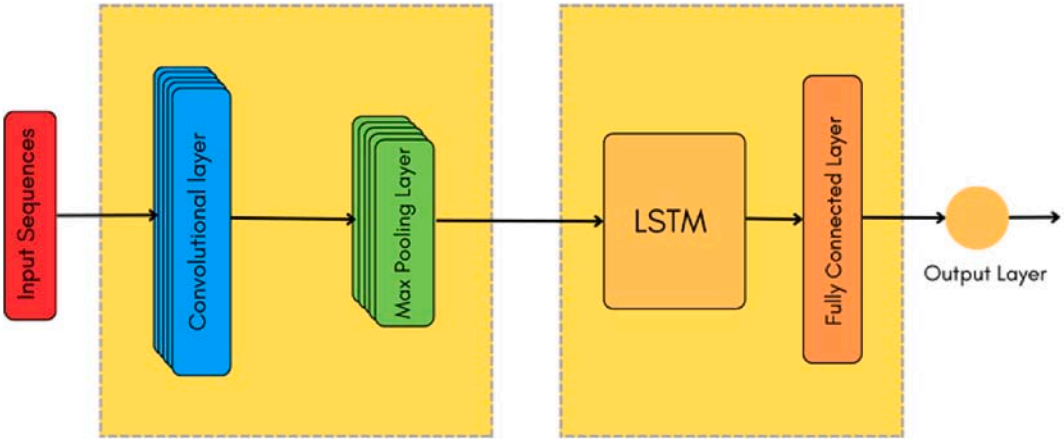

Fig. 7. CNN-LSTM.

of LSTMs. Despite their effectiveness, CNN-LSTM models are highly sensitive to hyperparameters, which directly influence their predictive performance. This model is specifically designed to process temporal and spatial data together. With the CNN Layer, meaningful features are extracted from images, time series or other multidimensional data. Spatial features in the data are captured through the convolution (filtering) process. Short-term patterns can also be determined in time series data using 1D CNN. For solar radiation estimation, features can be extracted from meteorological factors and temporal dependencies can be captured using CNN-LSTM [48,49]. The core component of a CNN is the convolution operation, which applies a set of small filters (kernels) to the input data. Each filter slides across the input, computing dot products and generating feature maps that capture patterns such as edges, textures, or more complex structures. The process helps in automatic feature extraction without requiring manual engineering. The pooling layer reduces the spatial dimensions of feature maps while retaining the most important information. After several convolutional and pooling layers, the extracted features are flattened and passed through fully connected (dense) layers as seen in Fig. 7.

3.2. Regression models

3.2.1. MLP

Multi-Layer Perceptron (MLP) is among the most widely utilized types of artificial neural networks within deep learning frameworks. It features a configuration of fully connected layers, where each neuron is linked to every neuron in the preceding layer. This design enables the model to capture interactions among all features in the input data. Essentially, MLP processes input through one or more hidden layers to generate outputs as seen in Fig. 8. This architecture equips it with the ability to learn complex and non-linear relationships, making it suitable for both classification and regression tasks. The effectiveness of MLP hinges on appropriately determining the number of layers, the quantity of neurons, the choice of activation functions, and the tuning of hyperparameters.

In MLP model, learning algorithm primarily depends on the gradient descent method combined with the backpropagation technique. The primary aim of backpropagation is to minimize the network errors that can be computed during the transition to the output layer [41,50,51].

$$e = 0.5 \times \sum_{k=1}^{N} (o_k - t_k)^2 \tag{9}$$

N represents the number of neurons, o_k denotes the network output of the k-th neuron, and t_k is the target value [52].

3.2.2. Random Forest

Random Forest (RF) operates by combining many different decision trees [53]. Each tree is trained on a random subset of the dataset, and the model's results are obtained by combining the predictions made by these trees [54]. Random Forest is commonly used in both classification and regression problems. This model utilizes randomness in the tree generation to provide diversity and generalizability. Each tree in the ensemble is trained using a random subset of the dataset as shown in Fig. 9. Each tree generates its own prediction independently. The prediction result is obtained by combining the predictions of all trees. This method improves the overall performance of the model while also mitigating the risk of overfitting. The randomness makes the model more generalizable by allowing different trees to provide diverse predictions [55].

3.2.3. XGBoost

Extreme Gradient Boosting (XGBoost) is an improved version of gradient boosting algorithms and is a popular ensemble learning method used for high-performance modeling [57]. XGBoost offers effective performance, especially in large datasets and complex problems. Its advanced features enhance the overall success of the model and are used in various machine learning tasks. XGBoost offers high efficiency in both training and prediction phases. Particularly, the innovative algorithms used in the tree-growing process enable the model to learn quickly. The general function for prediction at step t is presented as follows in Eq. (10). $f_k(x_i)$ represents the score of the i-th example in the k-th tree at step t. $f_i^{(t)}$ and $f_i^{(t-1)}$ are the predictions at steps t and $t-1$, respectively, and x_i is the i-th input variable.

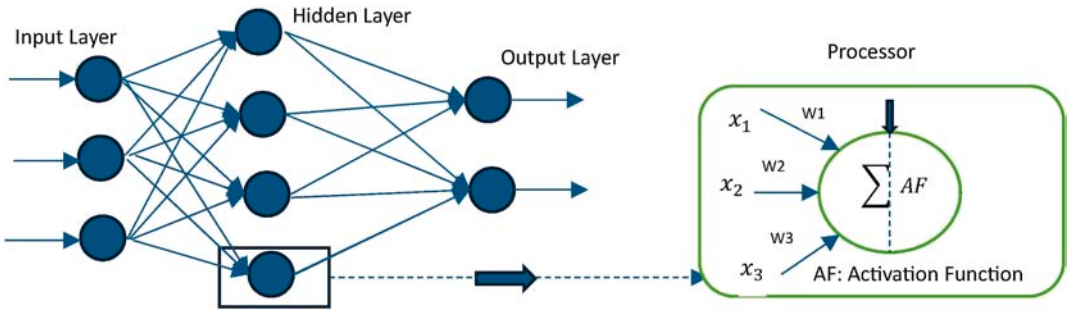

Fig. 8. MLP sample architecture.

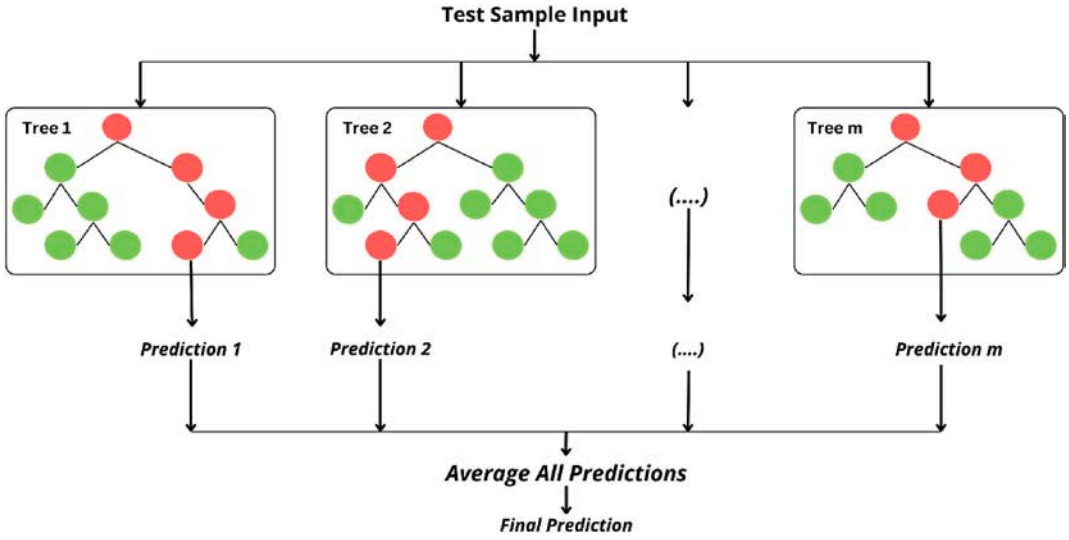

Fig. 9. Random Forest Regression [56].

$$f_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = f_i^{(t-1)} + f_t(x_i) \tag{10}$$

To prevent overfitting, the objective function of XGBoost consists of two main components: the training loss and the regularization, as shown in the equation.

$$Obj^{(t)} = \sum_{k=1}^{n} l(\overline{y}_i, y_i) + \sum_{k=1}^{t} \Omega(f_i) \tag{11}$$

The objective function that needs to be minimized during the model's learning process includes both the loss function and the regularization terms as illustrated in Eq. (11). l is the loss function, n is the number of observations used and Ω is the regularization term [58]. Loss Function measures the prediction errors of the model. This function is a value that needs to be minimized at each iteration to improve the model's accuracy. Regularization Terms control the complexity of the model to prevent overfitting. These terms are added to limit the complexity of the model in order to enhance its overall performance. XGBoost adds a new tree at each iteration to correct errors as shown in Fig. 10. This process allows the model to perform better at each iteration. Each iteration is optimized to minimize the prediction errors from the previous iterations. This iterative structure enhances the model's learning ability, enabling it to make more accurate predictions [59].

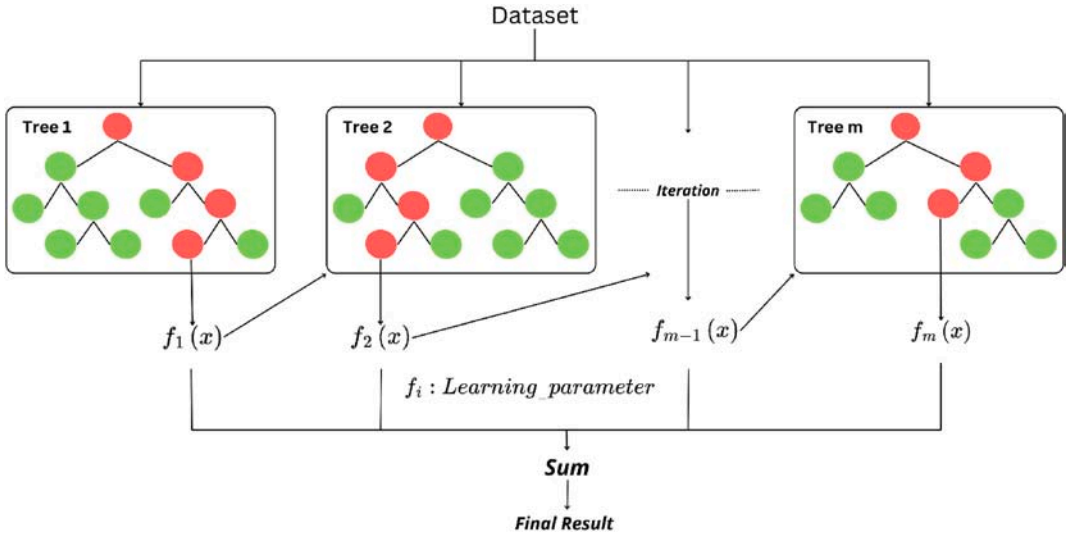

Fig. 10. XGBoost [60].

3.2.4. Support Vector Regression

SVR (Support Vector Regression) is a powerful machine learning algorithm that is effectively used in regression problems. SVR can demonstrate superior performance, especially in situations involving high-dimensional datasets and complex decision boundaries. Its fundamental principle is to find a hyperplane that provides the widest margin when classifying data. This improves the model's ability to generalize and lowers the likelihood of overfitting. The margin represents the distance between the closest examples of two classes, and the goal is to maximize this distance. SVR can use kernel functions to model nonlinear boundaries. SVR uses support vectors, to determine the boundary as shown in Fig. 11. These vectors are important data points that define the margin boundary and influence the model's performance.

Support Vector Regression (SVR) finds the best hyperplane in such a way that the predicted values do not fall outside a specified error tolerance (ε). The hyperplane can be expressed as follows in Eq. (12).

$$f(x) = w^T x + b \tag{12}$$

w is the weight vector. x is the input data point. b is the bias term. The optimization solution for SVR can be expressed as follows.

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i + \xi_i^* \tag{13}$$

$$\min_{w,b} \frac{1}{2}\|w\|^2 \; subject\; to \begin{cases} y_i - \left(w^T \varnothing(x_i) + b\right) \le \varepsilon + \xi_i \\ \left(w^T \varnothing(x_i) + b\right) - y_i \le \varepsilon + \xi_i^* \end{cases} and\; \xi_i, \xi_i^* \ge 0 \tag{14}$$

SVR uses an ε-insensitive loss function, meaning that predicted values are not penalized as long as they remain within a certain error tolerance (ε) from the actual values. In cases where exceeding the error tolerance is allowed, it attempts to minimize these deviations with penalty terms. The predicted value y_i is allowed to deviate from the actual value by at most ε. It may not always be possible for all data points to remain within this margin. Therefore, SVR allows for errors by using slack variables ξ_i and ξ_i^*. Thanks to kernel functions, it also provides effective results in non-linear datasets. The C parameter controls the model's error tolerance. High values of C allow the model to fit the training data more closely and reduce the margin of error, but they may increase the risk of overfitting. Low values of C, on the other hand, provide a wider margin but accept errors in the training data [15,23,61].

3.3. CNN-LSTM with optimization

3.3.1. Starfish Optimization

The Starfish Optimization Algorithm is a bio-inspired metaheuristic optimization technique modeled after the unique movement patterns of starfish. It leverages the exploration, preying and regeneration behaviors of starfish to navigate the search space, aiming to find the optimal solution as seen in Fig. 12. The algorithm dynamically balances exploration, to broadly search the solution space, and exploitation, to refine promising areas. This balance is achieved through the simulation of starfish arms' movements, adapting their positions based on the best-known solutions. The exploration phase adapts to the problem's dimensionality, employing a five-dimensional search for high-dimensional spaces ($D > 5$) and a unidimensional search for lower dimensions ($D \le 5$). In the exploitation phase, the algorithm employs a two-directional preying strategy to refine candidate solutions and includes a regeneration mechanism for the last starfish to enhance global convergence. By balancing exploration and exploitation effectively, SFOA demonstrates robust optimization capabilities. SFOA's flexible framework makes it suitable for tackling a wide range of optimization problems by effectively navigating complex, multidimensional search spaces [62]. r_1 is randomly generated for each candidate and update position in the iteration. X_i is used to determine the current location of starfish. X_{best} indicates the current best position.

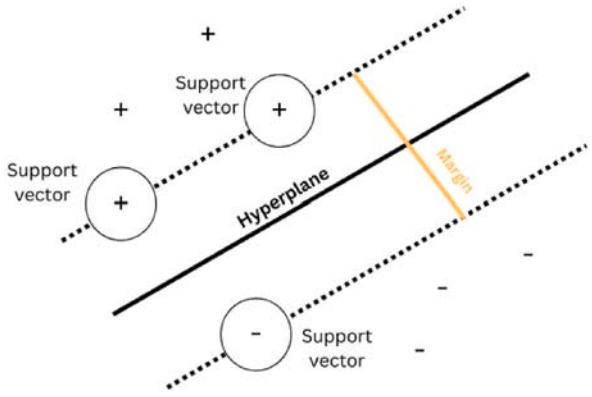

Fig. 11. Support vector regression.

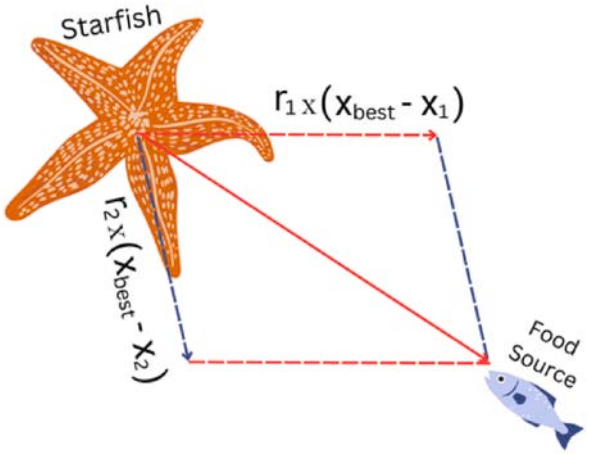

Fig. 12. Starfish behavior.

3.3.2. *Particle Swarm Optimization*

Particle Swarm Optimization is an optimization technique inspired by nature and based specifically on swarm behavior. It was proposed by Eberhart and Kennedy [63]. In the algorithm, each "particle" represents a possible solution and moves in a multi-dimensional solution space. Each individual has a unique perception ability; they identify the optimal local and global positions and determine their next move based on the current situation [64]. Each particle constantly updates its direction and speed by considering its own best position and the overall best position of the swarm in order to minimize or maximize the value of the target function. PSO is a preferred method especially in solving complex problems due to its simple structure and fast adaptation in the search for a solution [65]. The position x_i of each particle indicates the solution for the problem. The optimal solution is the point where this particle achieves the best result. This optimal solution determines the direction in which the particle moves and becomes the origin of the solution. Each particle possesses a memory that stores both its own best position achieved so far and the best position found by the entire population (to which it belongs). Based on the information retained in each particle's memory and its velocity v_i, the particle can update its position $x_i(t+1)$ as illustrated in Fig. 13 [66].

3.3.3. *Grey Wolf Optimizer*

Grey Wolf Optimizer is a nature-inspired metaheuristic algorithm that mimics the social hierarchy and hunting strategy of grey wolves. The algorithm simulates the leadership structure of a wolf pack, where the alpha, beta, delta, and omega wolves represent different roles. During the optimization process, the wolves follow the alpha, beta, and delta leaders to explore and exploit the search space, updating their positions based on the best solutions found. This hierarchy helps balance exploration and exploitation, making GWO an effective technique for solving various optimization problems [67]. Fig. 14 shows how wolves track prey for exploration and exploitation and how they change location in relation to each other. X indicates the position vector of the grey wolves. a is distance

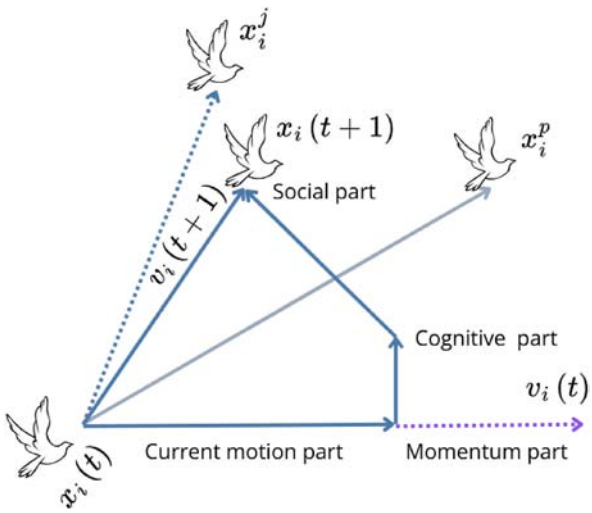

Fig. 13. Velocity and position updates of particles in PSO.

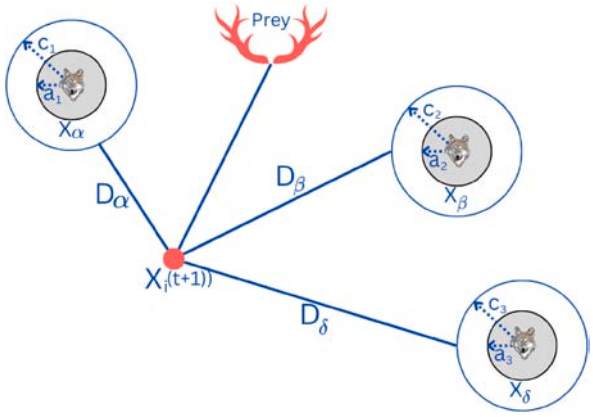

Fig. 14. GWO with wolves' update parameters.

control parameter. $a = 2 - T \times (2 \ / \ Max\_iter)$, with A ($a_1$, a_2,a_3) and C (c_1, c_2,c_3) vectors derived from a and random components to simulate hunting behavior. T is the current iteration.

3.3.4. Hybrid Optimized CNN-LSTM

The one purpose of this study is to use metaheuristic algorithms to optimize CNN-LSTM model for forecasting solar radiation based on historical data, aiming to enhance prediction accuracy by fine-tuning hyperparameters. CNN extracts features from data while LSTM learns time series dependencies. This combination is especially used in areas such as time series forecasting [47], text analysis [68], video classification [69], and financial forecasting [70]. For the prediction of daily solar radiation in Muş, CNN-LSTM along with PSO, GWO, and SFOA was used to determine the optimal values for learning rate, dropout rate, and LSTM units. Then, for each day, we summed the daily predictions obtained from each optimization method and divided the result by the number of optimization methods used. The final prediction value was selected based on this approach. This method follows the ensemble forecasting approach, combining different predictions obtained from various optimization methods to construct a more generalized model. The proposed approach is known as averaging ensemble, which serves as a good strategy to balance the strengths of different optimization algorithms. Taking the average of predictions from multiple optimization methods helps prevent overfitting, which may occur when relying on a single model. Since PSO, GWO, and SFOA have distinct characteristics. Their combination results in a more balanced prediction. Additionally, incorrect predictions made by one optimization algorithm can be counterbalanced by the others, making the model more stable.

Taking a weighted average of the predictions obtained from each optimization method based on their past accuracy can yield more precise results as seen in Eq. (15). In this study, the prediction results had similar error metric values, so weighting was not necessary. However, this approach can be considered when needed.

$$y_{ensemble} = w_1 y_{optimization-1} + w_2 y_{optimization-2} + \ldots + w_n y_{optimization-n} \tag{15}$$

Here, w_i represents the weights, which can be determined based on past error metrics. For example, a method with lower RMSE (Root Mean Square Error) or MSE values can be assigned a higher weight.

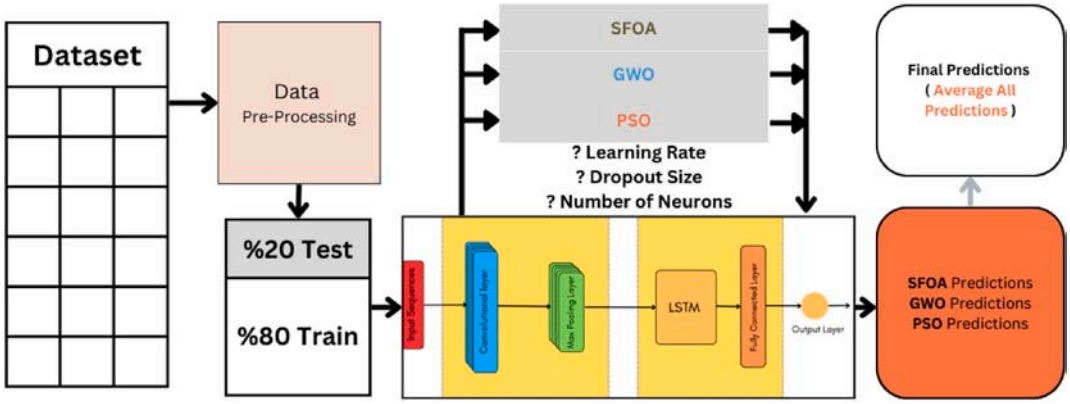

Fig. 15. Proposed CNN-LSTM model.

As illustrated in Fig. 15, the steps of proposed model;

1-Data Preparation	The historical data is normalized and structured as input-output pairs for model training and testing.
2-LSTM Model Construction	An CNN-LSTM model is designed to process data with optimization algorithms used to optimize hyperparameters such as **the learning rate, dropout rate** and **the number of neurons**.
3-Optimization Process	Optimization algorithms iteratively search for the best hyperparameter values, evaluating each model configuration based on Mean Squared Error (MSE) to minimize predict errors.
4-Model Training	Using the offered hyperparameters from optimization algorithms. CNN-LSTM model is trained to make predictions on the test data.
5- Final Prediction	Taking the average of predictions from multiple optimization methods. Implementing an ensemble learning approach.
5-Performance Evaluation	The model's accuracy is measured using various error metrics, offering a detailed assessment of its predictive power.

By leveraging optimization algorithms, this approach seeks to achieve a robust, fine-tuned CNN-LSTM model for accurately forecasting solar radiation, supporting data-driven decision making in energy-related applications.

4. Experimental results and discussion

The LSTM model used for forecasting was created using the Python Keras library. The model contains two LSTM layers. The first LSTM layer used 50 neurons. These cells help learn the long-term dependencies of the time series data. The outputs of this layer at all time steps were passed to the next layer. A single feature was used at each time step. The second and final LSTM layer was also composed of 50 neurons. This layer only returns the output of the last time step. The output of this layer was sent to the following dense layer. A dense layer with a single neuron was added. This layer takes the output from the LSTM layers and makes the prediction. The optimization of the model's weights was done using the Adam algorithm. Adam is an optimization algorithm that uses adaptive learning rates. The mean squared error was used as the loss function. The training dataset was presented to the model 100 times (epochs). The dataset was presented to the model in groups of 64 samples (batch size). To prevent overfitting, a validation dataset was used. As a result of various trials, it was observed that the method provided better results with these parameters.

When solar radiation measurements in Muş province were forecasted using the LSTM model, the model showed highly successful results, as can be seen from the error metric values in Fig. 16. The model's predictions were in strong agreement with the measured solar radiation values. The low values of the error metrics and the visual analysis of the predictions in the time series graphs indicated that the model successfully captured the seasonal and daily variations of solar radiation.

The ARIMA model used for prediction sequentially forecasts each observation in the test dataset. The parameters p, d and q in the model are important and need to be selected correctly. p was chosen as 4, representing the number of lags (Auto-Regressive). The value of d was chosen as 1, which is used in the Integrated part to make the series stationary. The Moving Average (MA) value q was set to 0, indicating no lags. Each new prediction incorporates information obtained from previous predictions, and the model is updated at each step.

ARIMA models have found wide application in energy management due to their flexibility and relatively simple structures. This demonstrates that the proposed model successfully analyzes historical solar radiation data and can make reliable predictions for future periods as seen in Fig. 17. Low error metric values highlight the effectiveness and accuracy of the ARIMA model. The results indicate that the ARIMA model serves as a reliable tool for forecasting solar radiation in Muş. This information will help gain further insights

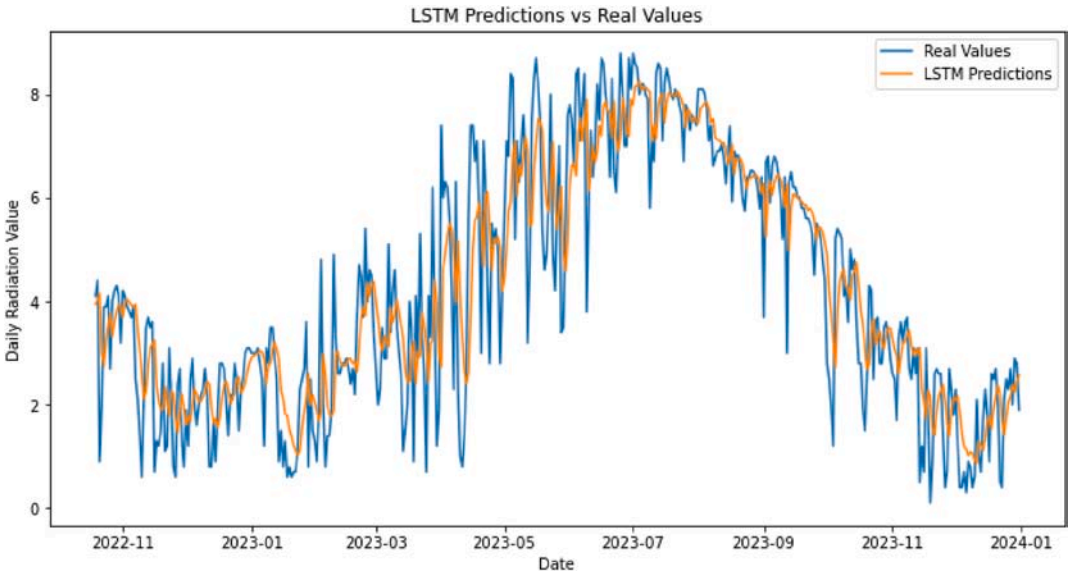

Fig. 16. Forecasting with LSTM

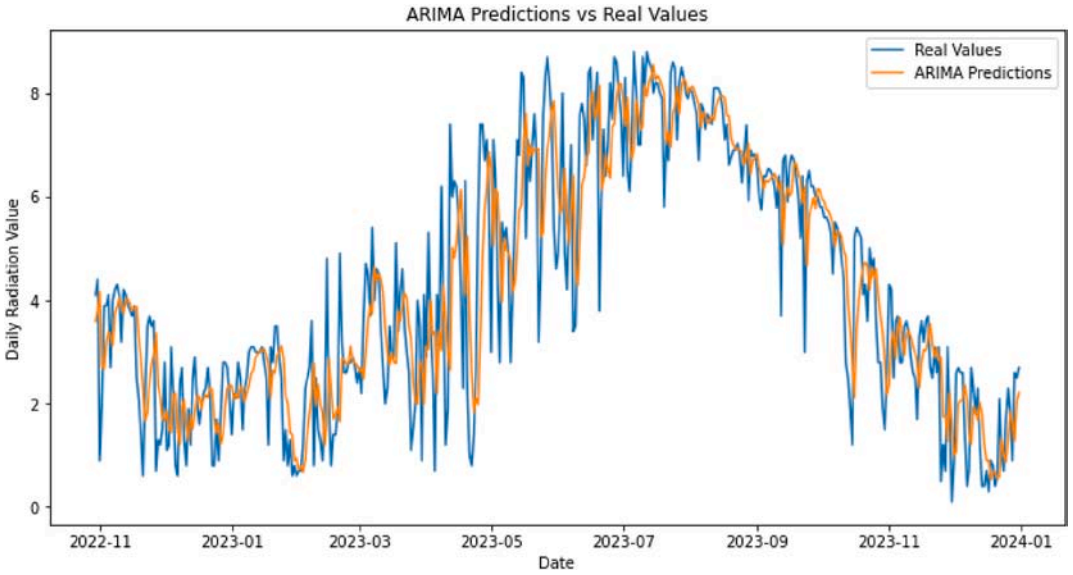

Fig. 17. Forecasting with ARIMA

into the solar energy potential of Muş and facilitate more informed decisions in planning and managing solar energy projects in the region. The performances of time series models according to error metrics are given in Fig. 18.

Forecasting with regression methods is different from forecasting with time series. Therefore, the values of humidity, wind speed, temperature, and pressure obtained from the Central Station were used. Solar radiation predictions were made based on these values. The statistical values of these variables are provided in Table 1. The obtained prediction results are presented with the error metrics RMSE, MAE (Mean Absolute Error), MSE, R^2(determination coefficient).

The MLP, SVR, and Random Forest models were implemented using the scikit-learn library, with default hyperparameters left unchanged; information about these hyperparameters can be found in the scikit-learn documentation. The MLP model consists of two hidden layers, each containing 50 neurons. The Rectified Linear Unit (ReLU) activation function is used to calculate the output of each neuron, which preserves positive values while zeroing out negative ones. The Adam optimization method was chosen for optimizing the weights. The 'rbf' (Radial Basis Function) kernel was chosen for the SVR model, as it is widely used to capture non-linear relationships. The number of decision trees in the Random Forest model was set to 100, while the XGBoost model utilized 'squarederror' as the loss function, aiming to minimize the mean of the squared errors. The parameters of each model directly affect its performance and operation, and they were optimized during the training process to achieve the best performance. Error metrics can be seen in Fig. 19.

The RMSE and MSE values indicate how much the predicted values deviate from the actual values. Low RMSE and MSE values represent better prediction performance. The RMSE and MSE values for the MLP and SVR models are lower than those of the others, suggesting that their predictions are closer to the actual values. In contrast, the Random Forest and XGBoost models have lower R^2 values and higher RMSE and MSE values, indicating their prediction performance is somewhat weaker than the others. The MAE value shows the average deviation of the predictions from the actual values, with the values for MLP and SVR being nearly identical for MAE. The prediction results are shown in Fig. 20.

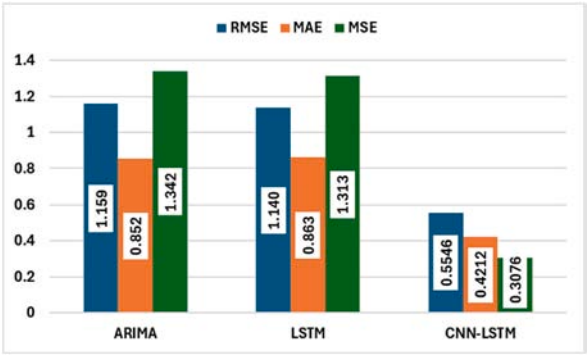

Fig. 18. Forecasting error values with time series models.

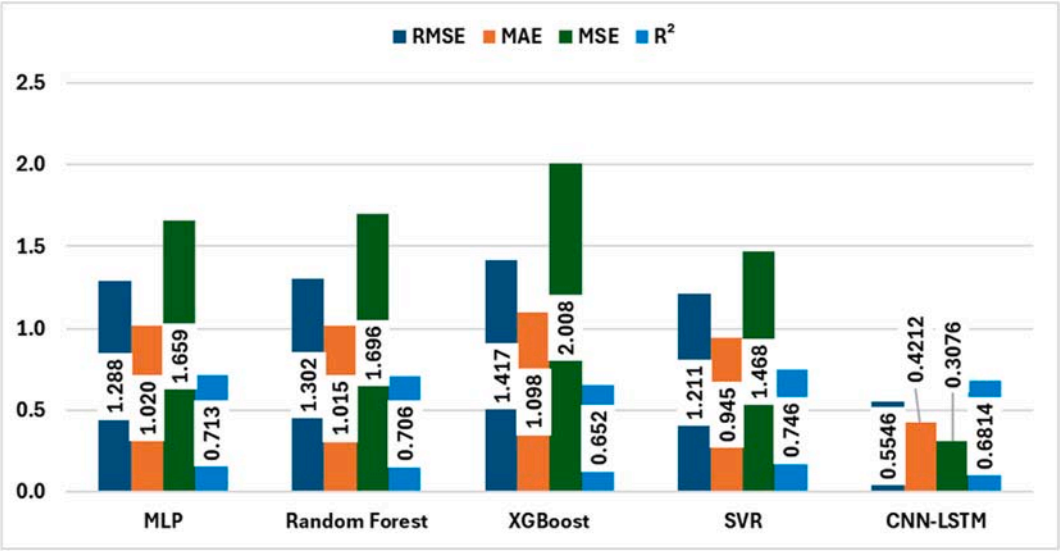

Fig. 19. Forecasting error values with regression models.

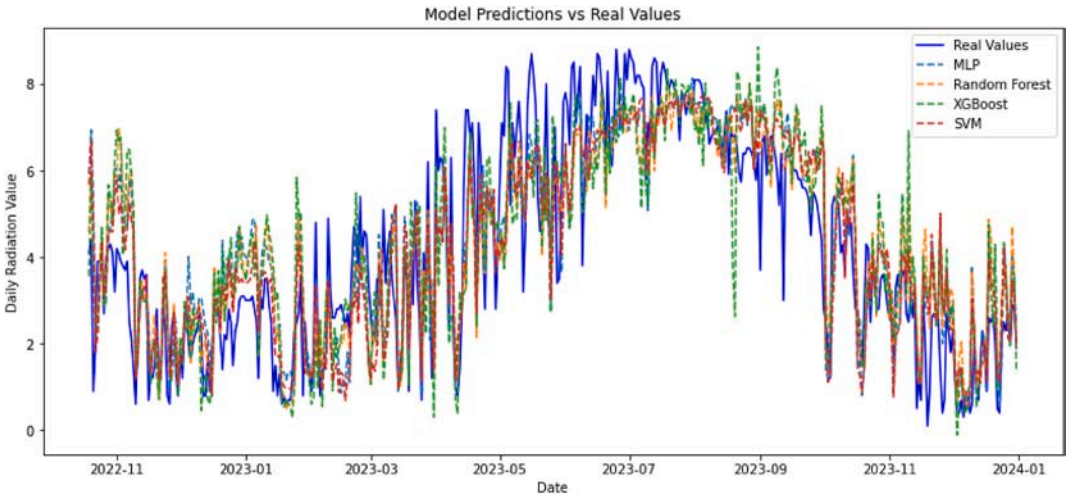

Fig. 20. Prediction with regression models.

This CNN-LSTM model is designed for time series forecasting. It starts with a Conv1D layer (64 filters, kernel size = 3, ReLU activation) to extract local patterns, followed by two stacked LSTM layers (100 and 50 units) to capture temporal dependencies. A Dense layer (50 units, ReLU activation) enhances feature learning, and a Dropout layer (0.2) prevents overfitting. The final Dense layer outputs the prediction. The model is compiled with the Adam optimizer (learning rate = 0.001) and uses MSE as the loss function with MAE as a performance metric. CNN-LSTM predictions results can be seen in Fig. 21.

The Optimized CNN-LSTM model integrates Conv1D for feature extraction and LSTM layers for capturing temporal dependencies in time series data. The Conv1D layer (64 filters, kernel size = 3, ReLU activation) detects local patterns before passing the sequence to LSTMs. Two stacked LSTM layers (optimized units, second LSTM with half the first layer's units) process long-term dependencies, followed by a Dropout layer (tuned dropout rate) to prevent overfitting. Finally, a Dense layer outputs the prediction. Hyperparameters such as LSTM units, dropout rate, and CNN filter size were optimized using PSO, GWO, and SFOA, leading to an improved hybrid model.

The combination of deep learning and optimization methods seems to provide better predictions. The Hybrid CNN-LSTM with optimization algorithms further improved the best forecasting. This is due to the ability of deep learning models to learn more complex relationships and patterns in meteorological datasets. While obtaining this result, the maximum iteration value for optimization algorithms was taken as 3. The learning rate range was chosen between 0.0001 and 0.01 and the number of neurons was chosen between 50 and 200. The dropout rate was chosen between 0.2 and 0.5.

The computational complexity of metaheuristic algorithms is a crucial aspect when evaluating their efficiency, particularly for

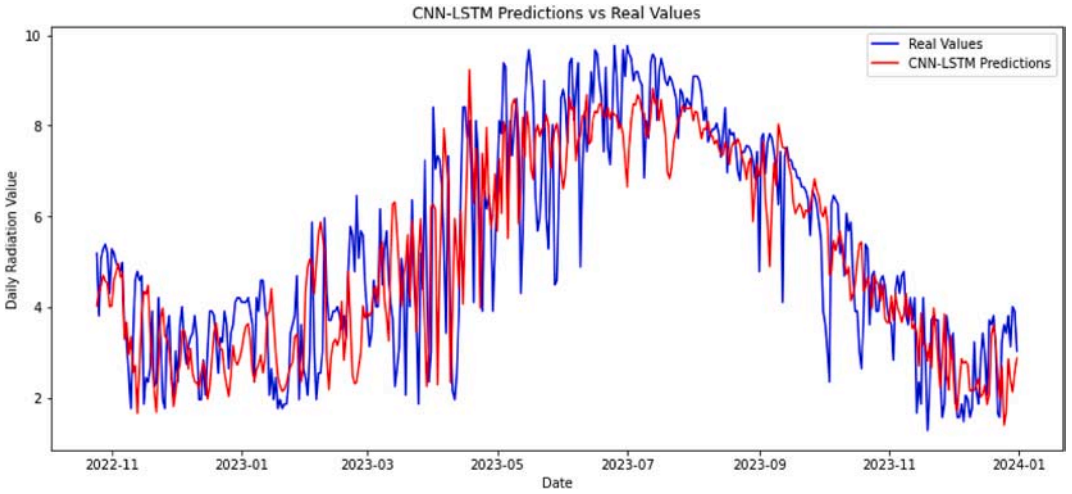

Fig. 21. Forecasting with CNN-LSTM.

high-dimensional optimization tasks such as deep learning hyperparameter tuning. Optimization algorithms typically require multiple iterations to converge, leading to increased computational time and resource utilization. For instance, optimization algorithms often involve swarm-based searches that scale with the number of particles and iterations, which can significantly increase runtime. PSO, GWO, and SFOA all exhibit a similar theoretical time complexity of $\mathcal{O}(N \cdot D \cdot T)$, where N denotes the population size, D the dimensionality of the search space, and T the number of iterations. In PSO, this complexity arises from updating each particle's position and velocity at every iteration, as well as evaluating the fitness function. GWO, while lacking velocity vectors, requires sorting and position updates based on three leader wolves (alpha, beta, and delta), resulting in comparable iteration-wise computational load. SFOA, also follows this general complexity form, but introduces a dynamic mechanism that switches between five-dimensional and unidimensional search strategies depending on the problem size, which can slightly improve efficiency in high-dimensional settings. While all three algorithms share the same asymptotic complexity, the practical computational cost can vary due to differences in update equations and behavioral strategies specific to each algorithm [62,63,71].

Implementing these algorithms alongside CNN-LSTM requires careful coordination between the optimization and training processes, especially when dealing with a large hyperparameter space. The computational complexity of a CNN-LSTM model arises from the combination of convolutional and recurrent components, each contributing distinct processing costs. Therefore, although CNN-LSTM offers strong spatiotemporal modeling capabilities, it can be computationally intensive, particularly for long sequences or large numbers of filters and hidden units, making optimization techniques essential for practical deployment. Although using three different optimization methods together may seem computationally expensive, the processing cost was minimized by keeping the population size at 10 and the maximum number of iterations at 3 for all optimization methods as seen in Table 2. This ensured that both time and computational costs remained low. Additionally, the model was trained for 50 epochs with a batch size of 16 to balance performance and efficiency. While all three algorithms share the same asymptotic complexity, practical differences in update rules and internal mechanics can lead to variations in runtime. This approach, although computationally intensive, is justified by the significant gains in model performance and generalization through effective hyperparameter selection. Hybrid Optimized CNN-LSTM prediction results can be seen in Fig. 22.

The Hybrid model outperforms all others across all metrics. PSO, GWO, and SFOA improve upon CNN-LSTM, but Hybrid is the best. The biggest improvement is in R^2, showing the Hybrid model explains significantly more variance. The Hybrid model achieves the lowest MAE, indicating that it produces the smallest average deviation from actual values. This suggests an improved generalization capability and better overall predictive performance compared to individual optimization methods.

The Hybrid model significantly reduces MSE, showing a major reduction in large deviations. When Table 3 is examined, the improvement rates of the models can be seen. The fact that the MSE improvement (21.4 %) is greater than MAE improvement (14.3 %) suggests that the Hybrid model effectively minimizes extreme prediction errors. The low RMSE in the Hybrid model shows that it does not make large estimation errors. The Hybrid model achieves the highest R^2, suggesting that it explains approximately 74.94 % of the variance in the data. Reducing MAE and MSE together suggests improved robustness and reduced bias in time-series forecasting.

Table 2
Comparison of PSO, GWO, and SFOA settings.

	PSO	GWO	SFOA
Population Size	10	10	10
Max Iterations	3	3	3
Parameters	w = 0.5 c1 = 1.5, c2 = 1.5	a (dynamic)	GP = 0.7

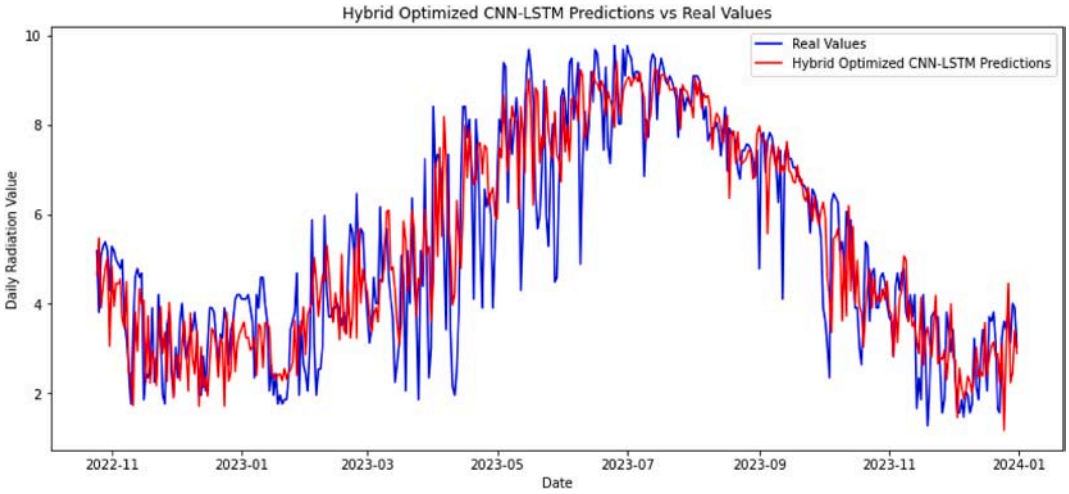

Fig. 22. Forecasting with hybrid (ensemble) optimized CNN-LSTM.

Table 3
Error measures and improvement rates.

	MAE	Improvement rate	MSE	Improvement rate	RMSE	Improvement rate	R^2	Improvement rate
CNN-LSTM	0.4212	baseline	0.3076	baseline	0.5546	baseline	0.6814	baseline
PSO	0.3938	6.5 %	0.2905	5.6 %	0.539	2.8 %	0.6991	2.6 %
GWO	0.3947	6.3 %	0.2905	5.6 %	0.5284	4.7 %	0.7109	4.3 %
SFOA	0.3951	6.2 %	0.2812	8.6 %	0.5303	4.4 %	0.7087	4.0 %
Hybrid	0.3611	**14.3 %**	0.2419	**21.4 %**	0.4919	**11.3 %**	0.7494	**10.0 %**

When examining the results of the models used to predict solar radiation values in Muş province, it is evident that successful predictions have been made, as shown in Fig. 23. The predicted values align well with the actual measurements, indicating that the models have made accurate forecasts. The results confirm the utility of Hybrid (Ensemble) in improving deep learning models. These techniques systematically explore the hyperparameter space, ensuring better convergence and model performance. While LSTM performs better, the addition of CNN and optimization techniques further enhances its predictive power by fine-tuning hyperparameters and improving generalization. Hybrid Optimized, PSO, GWO, and SFOA CNN-LSTM prediction results can be seen in Fig. 24.

When Fig. 25 is examined, the models can be evaluated based on error metrics. According to the R^2 analysis, the Hybrid model has the highest median value and stands out as the most successful model in explaining the target variable. While the GWO and SFOA methods also provide high accuracy levels, CNN-LSTM has the lowest R^2 values. This situation shows that the basic model, which is not supported by optimization algorithms, has a limited explanatory power. When the RMSE results are examined, it is observed that the Hybrid model minimizes the prediction errors on a square root basis. This model produces more consistent results by minimizing the

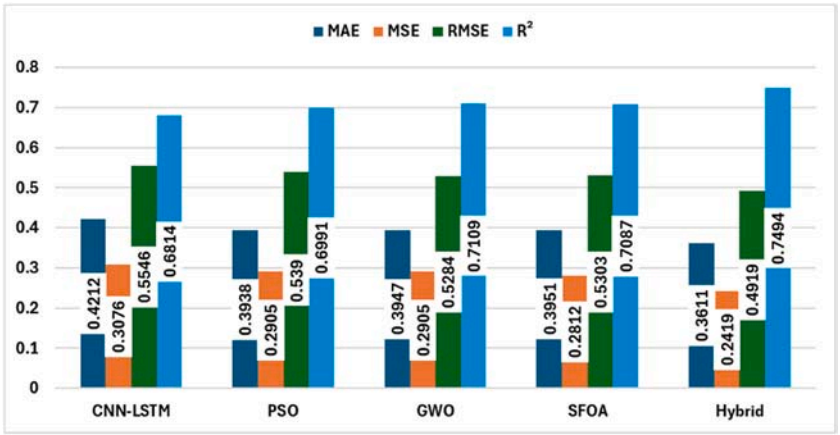

Fig. 23. CNN-LSTM and optimized models.

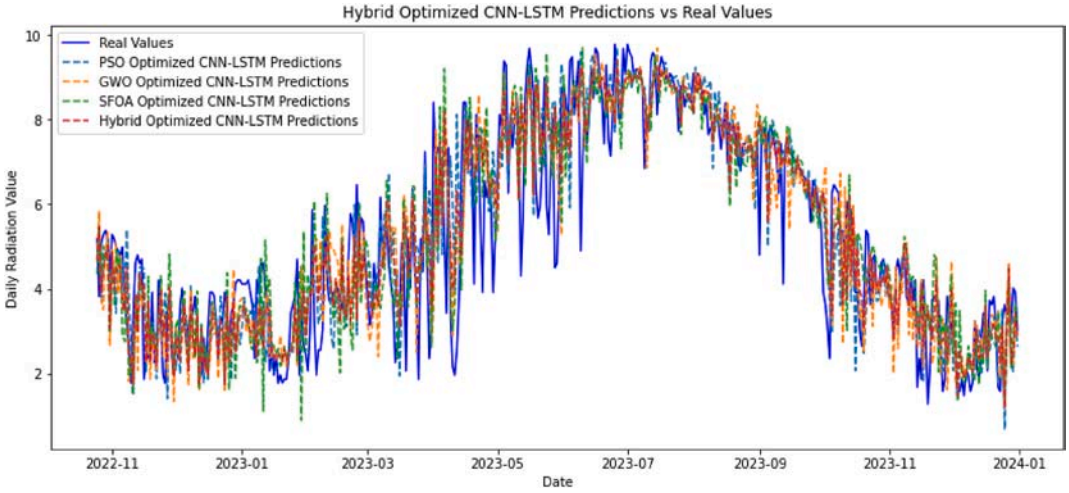

Fig. 24. Forecasting with hybrid optimized CNN-LSTM (ensemble), PSO, GWO, and SFOA

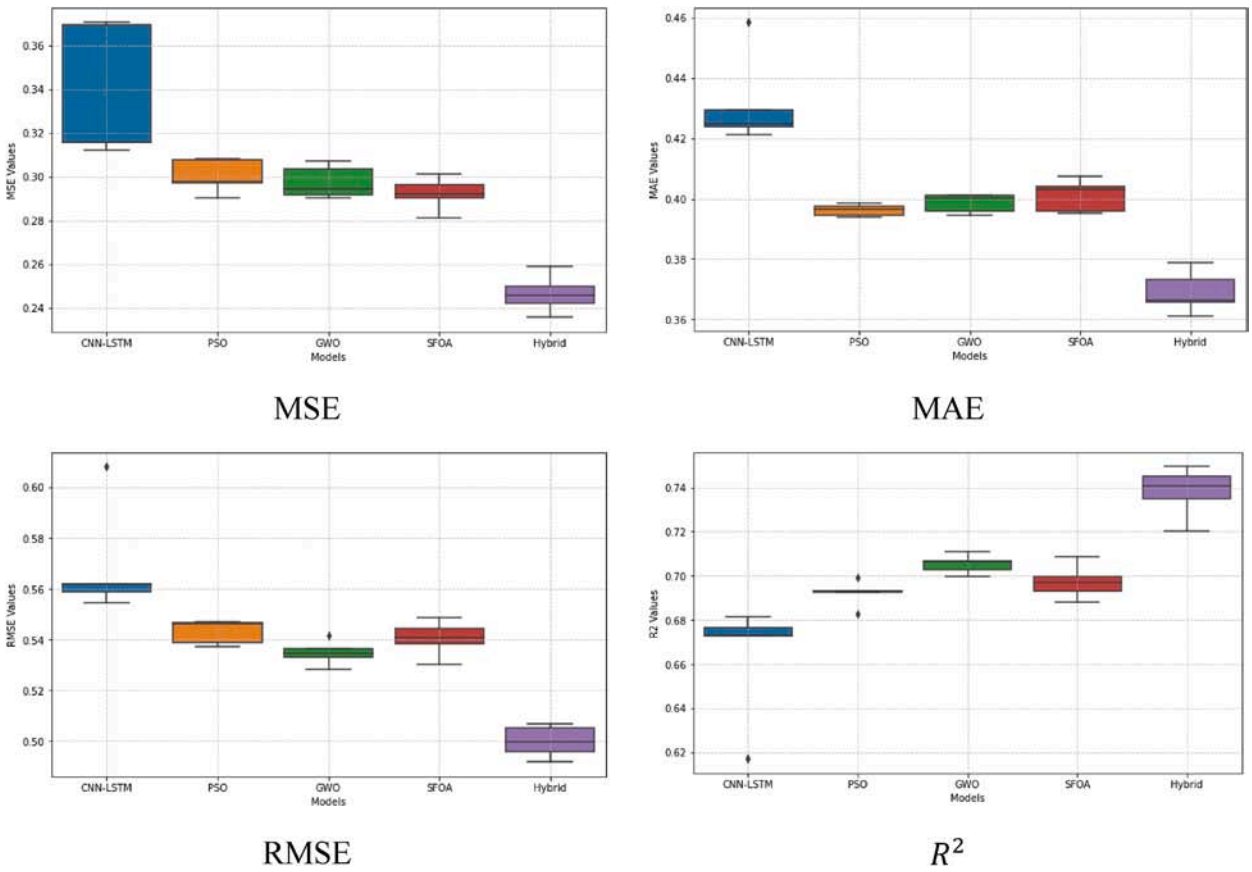

Fig. 25. Boxplot of error metrics.

magnitude of the deviations. While the GWO and SFOA methods show a performance close to Hybrid, the CNN-LSTM model stands out with high deviations. In terms of MAE, the Hybrid model stands out by making predictions with the lowest deviations and produces smaller errors in each example. Other methods exhibited relatively higher absolute errors. The fact that CNN-LSTM has the highest error values here reinforces the weakness of the basic model. Finally, in the MSE (Mean Square Error) analysis, the Hybrid method again reached the lowest values and proved that it is resistant to large error values. SFOA and GWO also produced successful results according to this metric, while CNN-LSTM had the highest error squares.

When all these findings are evaluated together, it is clear that the Hybrid approach is superior in terms of all metrics and is the most reliable model in terms of performance. In addition, it can be concluded that optimization-based methods, especially when integrated with classical deep learning models, significantly increase prediction performance.

The error distributions of the models are shown in Fig. 26. Whether there is a significant difference between CNN-LSTM and Hybrid prediction results can be seen with error rate, standard deviation, Paired T-Test and Wilcoxon Test. Hybrid model has a lower error margin in terms of mean error. The standard deviation value (Hybrid Optimized CNN-LSTM: 11.7850, CNN-LSTM: 13.5794) is also smaller in this model. This shows that the model produces more stable predictions. CNN-LSTM model exhibits lower performance with a higher mean error and standard deviation value. Paired T-test tests whether there is a significant difference between the means of two different measurements made on the same group. Paired T-test shows that the performance of the two models is significantly different. Negative t-statistic (−4.3631) shows that the errors of hybrid model are systematically lower than CNN-LSTM model. The p-value of 0.0 reveals that the probability of this difference occurring randomly is almost zero. The Wilcoxon test is a nonparametric test, meaning it does not require distribution assumptions. The w-statistic value is 36054 and the p-value is 0.0. The results show that the error distributions of the two methods are also significantly different. This supports that the Hybrid model consistently performs better. As a result, the hybrid model provides a statistically significant improvement by reducing error rates and producing more stable estimates. The proposed method consistently outperforms the baseline models from a statistical perspective, so energy planners and engineers can adopt the model with greater confidence, knowing that it is reliably better.

When evaluating all methods together, Hybrid CNN-LSTM shows the lowest results in MSE, MAE and RMSE, indicating its strong performance in handling both large and small errors. Additionally, the ability that CNN brings to LSTM can be seen in the results. LSTM's ability to learn long-term dependencies in time series offers an advantage over linear models like ARIMA. This demonstrates the powerful performance of deep learning-based models. MLP and SVR exhibit similar performance in terms of MSE, MAE, and RMSE, but they have higher errors compared to time series models, indicating that they perform slightly worse in both absolute and large errors. Random Forest and XGBoost, have the highest values across all three error metrics, suggesting that their predictions are more inaccurate than those of the other models. Ensemble models are generally strong in complex datasets, but in this case, XGBoost has shown lower performance compared to the other models.

5. Conclusions

In this study, an evaluation of classical and newly proposed methods for estimating solar radiation based on historical data is presented in order to assess the solar energy potential in the Muş Province located in the Eastern Anatolia Region of Türkiye. The methods employed include time series approaches such as ARIMA and LSTM, as well as regression techniques like MLP, Random Forest, XGBoost, and SVR, all of which have yielded satisfactory prediction results. Additionally, predictions were made with CNN-LSTM. Addition to these, CNN-LSTM predictions improved further with PSO, GWO and SFOA for hyperparameter tuning, delivering the better results. Finally, a Hybrid prediction model was developed by averaging the results obtained with the optimization methods using an ensemble approach. Hybrid optimization techniques improve the predictive accuracy of deep learning models by leveraging the strengths of multiple optimization strategies. The study highlights the superiority of Hybrid CNN-LSTM models in both predictive accuracy and robustness. The optimized model achieve the most significant improvements, showcasing the potential of ensemble methodologies that combine optimization algorithms with deep learning for solar radiation. This study is the first to apply

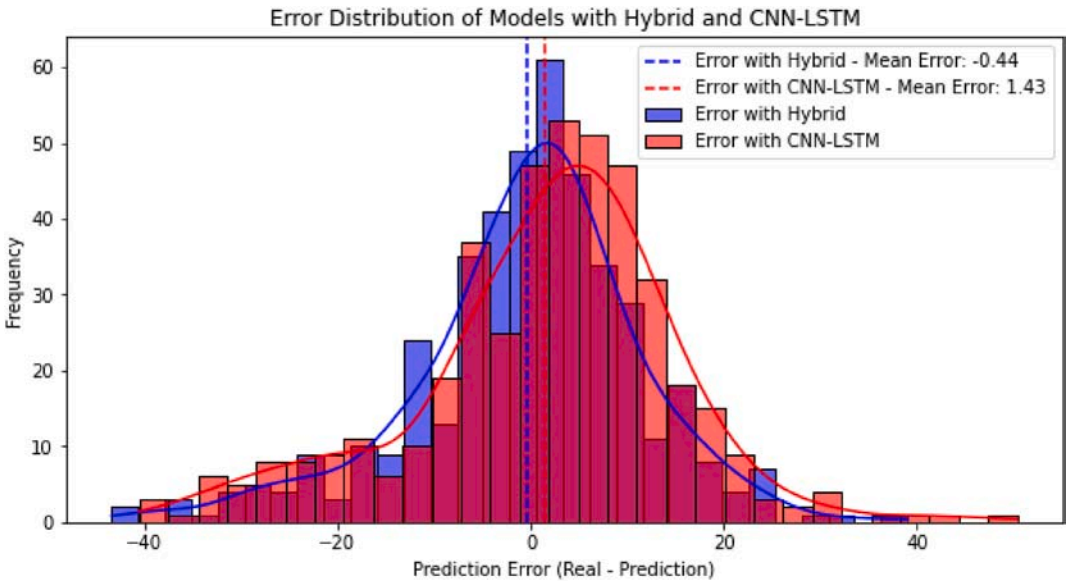

Fig. 26. Error distribution of models.

these methods for solar radiation forecasting in Muş, a region with significant potential for solar energy production. The findings highlight the region's solar energy potential, offering supportive insights for regional analyses. The application of deep learning methods proves crucial, particularly for improving prediction accuracy by considering the region's specific meteorological conditions. Furthermore, the study serves as a valuable guide for future research, detailing limitations and the scope of datasets used.

The findings of this study will contribute to the development of local energy policies and serve as a regional reference for future renewable energy projects. The accuracy of solar energy predictions will aid in effective decision-making for energy production planning, energy management, and environmental sustainability, thereby contributing to economic growth and societal welfare. This will support optimizing energy investments and facilitate the more efficient use of clean energy sources. By enhancing the accuracy of solar radiation predictions, it will also support the development of renewable energy strategies and promote the preference and use of clean energy in the context of climate crisis and natural disasters. In conclusion, this study can be regarded as a valuable resource for assessing solar energy potential and planning future energy strategies, as well as serving as a guide for future research.

Future studies could incorporate meteorological data such as solar irradiance intensity and cloud movement, adding new dimensions to the analysis. With more comprehensive datasets and advanced machine learning techniques, future research can further enhance understanding in this field. A further comparative study with Transformer-based models or hybridizing CNN-LSTM with attention mechanisms could provide additional insights into improving solar radiation forecasting models. Additionally, identifying regions within Muş with high energy potential or suitable locations for solar power plant installations could enable targeted measurements and provide deeper insights.

CRediT authorship contribution statement

İrem Fatma Şener: Writing – original draft, Software, Methodology, Formal analysis, Data curation, Conceptualization. **İhsan Tuğal:** Writing – review & editing, Writing – original draft, Validation, Supervision, Software, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We would like to thank Muş Meteorology Provincial Directorate for sharing meteorological data with us.

Data availability

Data will be made available on request.

References

[1] E. Chodakowska, J. Nazarko, Ł. Nazarko, et al., ARIMA models in solar radiation forecasting in different geographic locations, Energies 16 (2023), https://doi.org/10.3390/en16135029. Epub ahead of print.
[2] C. Voyant, G. Notton, S. Kalogirou, et al., Machine learning methods for solar radiation forecasting: a review, Renew. Energy 105 (2017), https://doi.org/10.1016/j.renene.2016.12.095. Epub ahead of print.
[3] Türkiye ministry of energy and natural resources. Access Date:10.October.2024, https://enerji.gov.tr/bilgi-merkezi-enerji-gunes.
[4] A. Bingol, O. Arslan, Effective utilization of solar energy in sustainable agricultural irrigation a provincial example from Turkiye, Therm. Sci. 27 (2023) 3373–3384.
[5] M. Khashei, M. Bijari, A novel hybridization of artificial neural networks and ARIMA models for time series forecasting, Appl. Soft Comput. 11 (2011) 2664–2675.
[6] P.J. Brockwell, R.A. Davis, Introduction to Time Series and Forecasting, Springer International Publishing, Cham, 2016, https://doi.org/10.1007/978-3-319-29854-2. Epub ahead of print.
[7] I. Tugal, F. Sevgin, Analysis and forecasting of temperature using time series forecasting methods a case study of Mus, Therm. Sci. 27 (2023) 3081–3088.
[8] M.H. Alsharif, M.K. Younes, J. Kim, Time series ARIMA model for prediction of daily and monthly average global solar radiation: the case study of Seoul, South Korea, Symmetry (Basel) 11 (2019) 240.
[9] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780.
[10] Y. Ning, H. Kazemi, P. Tahmasebi, A comparative machine learning study for time series oil production forecasting: ARIMA, LSTM, and prophet, Comput. Geosci. 164 (2022) 105126.
[11] U.M. Sirisha, M.C. Belavagi, G. Attigeri, Profit prediction using ARIMA, SARIMA and LSTM models in time series forecasting: a comparison, IEEE Access 10 (2022) 124715–124727.
[12] M. Vakili, S.-R. Sabbagh-Yazdi, K. Kalhor, et al., Using artificial neural networks for prediction of global solar radiation in Tehran considering particulate matter air pollution, Energy Proc. 74 (2015) 1205–1212.
[13] B. Babar, L.T. Luppino, T. Boström, et al., Random forest regression for improved mapping of solar irradiance at high latitudes, Sol. Energy 198 (2020) 81–92.
[14] Bamisile O, Ejiyi CJ, Osei-Mensah E, et al. Long-term prediction of solar radiation using XGboost, LSTM, and machine learning algorithms. In: 2022 4th Asia Energy and Electrical Engineering Symposium (AEEES). IEEE, pp. 214–218.
[15] Z. Ramedani, M. Omid, A. Keyhani, et al., Potential of radial basis function based support vector regression for global solar radiation prediction, Renew. Sustain. Energy Rev. 39 (2014) 1005–1011.
[16] B. Belmahdi, M. Louzazni, A. El Bouardi, Comparative optimization of global solar radiation forecasting using machine learning and time series models, Environ. Sci. Pollut. Res. 29 (2022) 14871–14888.

[17] M. Aslam, J.M. Lee, H.S. Kim, et al., Deep learning models for long-term solar radiation forecasting considering microgrid installation: a comparative study, Energies 13 (2019), https://doi.org/10.3390/en13010147. Epub ahead of print.
[18] R. Srivastava, A.N. Tiwari, V.K. Giri, Solar radiation forecasting using MARS, CART, M5, and random forest model: a case study for India, Heliyon 5 (2019) e02692, https://doi.org/10.1016/j.heliyon.2019.e02692. Epub ahead of print.
[19] A.G. Kaplan, Y.A. Kaplan, Using of the weibull distribution in developing global solar radiation forecasting models, Environ. Prog. Sustain. Energy (2024), https://doi.org/10.1002/ep.14380. Epub ahead of print.
[20] Y. Djeldjeli, L. Taouaf, S. Alqahtani, et al., Enhancing solar power forecasting with machine learning using principal component analysis and diverse statistical indicators, Case Stud. Therm. Eng. 61 (2024) 104924.
[21] R. Ahmed, V. Sreeram, Y. Mishra, et al., A review and evaluation of the state-of-the-art in PV solar power forecasting: techniques and optimization, Renew. Sustain. Energy Rev. 124 (2020) 109792.
[22] J.M. Álvarez-Alvarado, J.G. Ríos-Moreno, S.A. Obregón-Biosca, et al., Hybrid techniques to predict solar radiation using support vector machine and search optimization algorithms: a review, Appl. Sci. 11 (2021) 1044.
[23] E.S. Solano, P. Dehghanian, C.M. Affonso, Solar radiation forecasting using machine learning and ensemble feature selection, Energies 15 (2022), https://doi.org/10.3390/en15197049. Epub ahead of print.
[24] N. Krishnan, K. Ravi Kumar, S.A. R, Solar radiation forecasting using gradient boosting based ensemble learning model for various climatic zones, Sustain Energy, Grids Networks 38 (2024), https://doi.org/10.1016/j.segan.2024.101312. Epub ahead of print.
[25] T. Peng, C. Zhang, J. Zhou, et al., An integrated framework of Bi-directional long-short term memory (BiLSTM) based on sine cosine algorithm for hourly solar radiation forecasting, Energy 221 (2021), https://doi.org/10.1016/j.energy.2021.119887. Epub ahead of print.
[26] N. Krishnan, K.R. Kumar, C.S. Inda, How solar radiation forecasting impacts the utilization of solar energy: a critical review, J. Clean. Prod. 388 (2023), https://doi.org/10.1016/j.jclepro.2023.135860. Epub ahead of print.
[27] V.Z. Antonopoulos, D.M. Papamichail, V.G. Aschonitis, et al., Solar radiation estimation methods using ANN and empirical models, Comput. Electron. Agric. 160 (2019) 160–167.
[28] A.R. Pazikadin, D. Rifai, K. Ali, et al., Solar irradiance measurement instrumentation and power solar generation forecasting based on artificial neural networks (ANN): a review of five years research trend, Sci. Total Environ. 715 (2020) 136848.
[29] Z. Allal, H.N. Noura, K. Chahine, Machine learning algorithms for solar irradiance prediction: a recent comparative study, e-Prime - Adv Electr Eng Electron Energy 7 (2024), https://doi.org/10.1016/j.prime.2024.100453. Epub ahead of print.
[30] S.A. Haider, M. Sajid, H. Sajid, et al., Deep learning and statistical methods for short- and long-term solar irradiance forecasting for islamabad, Renew. Energy 198 (2022) 51–60.
[31] Y.-Y. Hong, J.J.F. Martinez, A.C. Fajardo, Day-ahead solar irradiation forecasting utilizing Gramian angular field and convolutional long short-term memory, IEEE Access 8 (2020) 18741–18753.
[32] M. Sakib, T. Siddiqui, S. Mustajab, et al., An ensemble deep learning framework for energy demand forecasting using genetic algorithm-based feature selection, PLoS One 20 (2025) e0310465.
[33] M. Chahardoli, N. Osati Eraghi, S. Nazari, An energy consumption prediction approach in smart cities by CNN-LSTM network improved with game theory and namib beetle optimization (NBO) algorithm, J. Supercomput. 81 (2025) 403.
[34] P.P. Phyo, Y.-C. Byun, Hybrid ensemble deep learning-based approach for time series energy prediction, Symmetry (Basel) 13 (2021) 1942.
[35] S. Khan, T. Mazhar, M.A. Khan, et al., Comparative analysis of deep neural network architectures for renewable energy forecasting: enhancing accuracy with meteorological and time-based features, Discov Sustain 5 (2024) 533.
[36] N.E. Benti, M.D. Chaka, A.G. Semie, Forecasting renewable energy generation with machine learning and deep learning: current advances and future prospects, Sustainability 15 (2023) 7087.
[37] J. Cao, Z. Li, J. Li, Financial time series forecasting model based on CEEMDAN and LSTM, Phys A Stat Mech its Appl 519 (2019) 127–139.
[38] Dadhich M, Doshi R, Rao SS, et al. Estimating and Predicting Models Using Stochastic Time Series ARIMA Modeling in Emergent Economy. pp. 295–305.
[39] E. Akarslan, F.O. Hocaoglu, A novel adaptive approach for hourly solar radiation forecasting, Renew. Energy 87 (2016), https://doi.org/10.1016/j.renene.2015.10.063. Epub ahead of print.
[40] M.Y. Anwar, J.A. Lewnard, S. Parikh, et al., Time series analysis of malaria in Afghanistan: using ARIMA models to predict future trends in incidence, Malar. J. 15 (2016) 566.
[41] W. Xu, L. Ning, Y. Luo, Wind speed forecast based on post-processing of numerical weather predictions using a gradient boosting decision tree algorithm, Atmosphere (Basel) 11 (2020) 738.
[42] M. Geurts, G.E.P. Box, G.M. Jenkins, Time series analysis: forecasting and control, J. Mark. Res. 14 (1977) 269.
[43] Younes MK, Nopiah ZM, Basri NEA, et al. Medium term municipal solid waste generation prediction by autoregressive integrated moving average. In: AIP Conference Proceedings, pp. 427–435.
[44] D. Ömer Faruk, A hybrid neural network and ARIMA model for water quality time series prediction, Eng. Appl. Artif. Intell. 23 (2010) 586–594.
[45] Kumar S. Time series analysis. In: Python for Accounting and Finance. Cham: Springer Nature Switzerland, pp. 381–409.
[46] I.E. Livieris, E. Pintelas, P. Pintelas, A CNN–LSTM model for gold price time-series forecasting, Neural Comput. Appl. 32 (2020) 17351–17360.
[47] M. Karbasi, M. Ali, S.M. Bateni, et al., Multi-step ahead forecasting of electrical conductivity in Rivers by using a hybrid convolutional neural network-long short-term memory (CNN-LSTM) model enhanced by Boruta-XGBoost feature selection algorithm, Sci. Rep. 14 (2024) 15051.
[48] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.
[49] C. Feng, J. Zhang, SolarNet: a sky image-based deep convolutional neural network for intra-hour solar forecasting, Sol. Energy 204 (2020) 71–78.
[50] A. Moradzadeh, A. Mansour-Saatloo, B. Mohammadi-Ivatloo, et al., Performance evaluation of two machine learning techniques in heating and cooling loads forecasting of residential buildings, Appl. Sci. 10 (2020) 3829.
[51] E. Eker, Ş. Atar, F. Şevgin, et al., Optimization of non-linear problems using salp swarm algorithm and solving the energy efficiency problem of buildings with salp swarm algorithm-Based multi-layer perceptron algorithm, ELECTRICA 24 (2024) 436–449.
[52] L. Wang, O. Kisi, M. Zounemat-Kermani, et al., Solar radiation prediction using different techniques: model evaluation and comparison, Renew. Sustain. Energy Rev. 61 (2016) 384–397.
[53] Long F, Xu M, Li Y, et al. XiaoA: a robot editor for popularity prediction of online news based on ensemble learning. In: Intelligence Science vol. II, pp. 340–350.
[54] L. Breiman, Random forests, Mach. Learn. 45 (2001), https://doi.org/10.1023/A:1010933404324. Epub ahead of print.
[55] M. Schonlau, R.Y. Zou, The random forest algorithm for statistical learning, Stata J Promot Commun Stat Stata 20 (2020) 3–29.
[56] Sohal JS, Hegde V, Smitha B. Solar Power Probabilistic Prediction Using Random Forest Regressor. pp. 123–135.
[57] Tianqi C, Tong H. Xgboost: Extreme Gradient Boosting. R Packag.
[58] J. Fan, X. Wang, L. Wu, et al., Comparison of support vector machine and extreme gradient boosting for predicting daily global solar radiation using temperature and precipitation in humid subtropical climates: a case study in China, Energy Convers. Manag. 164 (2018) 102–111.
[59] Chen T, Guestrin C. XGBoost. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA: ACM, pp. 785–794.
[60] R. Piraei, S.H. Afzali, M. Niazkar, Assessment of XGBoost to estimate total sediment loads in Rivers, Water Resour. Manag. 37 (2023) 5289–5306.
[61] S. Belaid, A. Mellit, Prediction of daily and mean monthly global solar radiation using support vector machine in an arid climate, Energy Convers. Manag. 118 (2016) 105–118.
[62] C. Zhong, G. Li, Z. Meng, et al., Starfish optimization algorithm (SFOA): a bio-inspired metaheuristic algorithm for global optimization compared with 100 optimizers, Neural Comput. Appl. (2024), https://doi.org/10.1007/s00521-024-10694-1. Epub ahead of print.
[63] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks. IEEE, pp. 1942–1948.

[64] H. Hauduc, M.B. Neumann, D. Muschalla, et al., Efficiency criteria for environmental model quality assessment: a review and its application to wastewater treatment, Environ. Model. Software 68 (2015) 196–204.
[65] Y. Xu, C. Hu, Q. Wu, et al., Research on particle swarm optimization in LSTM neural networks for rainfall-runoff simulation, J. Hydrol. 608 (2022) 127553.
[66] A.H. Elsheikh, M. Abd Elaziz, Review on applications of particle swarm optimization in solar energy systems, Int. J. Environ. Sci. Technol. 16 (2019) 1159–1170.
[67] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Software 69 (2014) 46–61.
[68] Liang S, Zhu B, Zhang Y, et al. A double channel CNN-LSTM model for text classification. In: 2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). IEEE, pp. 1316–1321.
[69] Abdullah M, Ahmad M, Han D. Facial expression recognition in videos: an CNN-LSTM based model for video classification. In: 2020 International Conference on Electronics, Information, and Communication (ICEIC). IEEE, pp. 1–3.
[70] W. Lu, J. Li, Y. Li, et al., A CNN-LSTM-Based model to forecast stock prices, Complexity 2020 (2020) 1–10.
[71] M.H. Nadimi-Shahraki, S. Taghian, S. Mirjalili, An improved grey wolf optimizer for solving engineering problems, Expert Syst. Appl. 166 (2021) 113917.