



## OPEN Explainable AI for intelligent green energy forecasting: deep learning with iHow optimization algorithm (iHOW)

Mahmoud Shabrawy<sup>1✉</sup>, Khaled Sh. Gaber<sup>1</sup>, Marwa M. Eid<sup>3,5</sup>, Amel Ali Alhussan<sup>4</sup>,  
Doaa Sami Khafaga<sup>4</sup> & El-Sayed M. El-kenawy<sup>2,6✉</sup>

Forecasting green energy with high accuracy is essential for effective energy management and ensuring the reliability of power grids. This is particularly important given the fluctuating nature of renewable sources such as solar and wind power, which require careful regulation to provide an uninterrupted electricity supply. To improve forecasting accuracy, this study integrates cutting-edge Dynamic Temporal Convolutional Networks (DTCNs), practical feature selection methods, and optimization through a novel metaheuristic algorithm. In the initial evaluation, our model achieved a Mean Squared Error (MSE) of 0.0845 and an R-squared ( $R^2$ ) score of 0.7265. These results indicate that the model was able to learn and replicate some of the underlying patterns in the training data. To enhance model performance, a feature selection process was applied to eliminate irrelevant information. This refinement reduced the MSE from 0.6988 to 0.0022 and improved the  $R^2$  score to 0.9005, thereby clarifying the relationship between input variables and green energy generation. Further performance improvements were obtained using a metaheuristic optimizer, the iHow Optimization Algorithm (iHOW). This approach led to a significant reduction in MSE to  $1.14 \times 10^{-5}$  and an increase in  $R^2$  to 0.9804. The combined use of feature selection, advanced optimization, and deep learning contributes substantially to forecasting precision. This approach offers a practical tool for energy professionals, supporting more reliable and efficient renewable energy production and contributing to a sustainable future.

**Keywords** Deep learning, Green energy forecasting, Metaheuristic optimization (iHOW), Feature selection, Renewable energy prediction

Emphasizing sustainability around the globe has resulted in many more investments in solar, wind, hydroelectric and geothermal energy. These new energy sources are attractive because they significantly decrease greenhouse gas emissions and can be used to replace fossil fuels, known to harm the climate<sup>1</sup>. Yet, since the power produced by renewables can be irregular, this introduces new obstacles to those overseeing the power grid. A good example is that changes in weather conditions can make it hard to maintain a regular supply of solar and wind electricity. For this reason, proper management should be guided by careful preparations for storing energy and budget planning<sup>2</sup>. It is essential for those involved to have robust and accurate approaches for forecasting when adding renewables to the grid. Using predictive analytics, grid operators can adjust their energy storage plans and fulfill the continued energy needs. With correct predictions, operators can reduce the use of back-up energy, as those alternatives often harm the environment and are costlier. Achieving a balance between energy use and generation allows organizations to support the stability and sustainability of the energy system<sup>3</sup>. Yet, despite the critical importance of accurate green energy forecasting, traditional methods, including statistical and time-series approaches, often fall short due to their limited capacity to capture the complex,

<sup>1</sup>Computer Engineering and Control Systems Department, Faculty of Engineering, Mansoura University, Mansoura, Egypt. <sup>2</sup>Department of Communications and Electronics, Delta Higher Institute of Engineering and Technology, Mansoura, Egypt. <sup>3</sup>Faculty of Artificial Intelligence, Delta University for Science and Technology, Mansoura, Egypt. <sup>4</sup>Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia. <sup>5</sup>Jadara University Research Center, Jadara University, Irbid 21110, Jordan. <sup>6</sup>Applied Science Research Center, Applied Science Private University, Amman, 11931, Jordan. ✉email: mshabrawy@std.mans.edu.eg; sayed.kenawy@dhiet.edu.eg

nonlinear, and multiscale dependencies inherent in renewable energy data<sup>4</sup>. These classical techniques, such as autoregressive integrated moving average (ARIMA), exponential smoothing, and linear regression, are built on fixed mathematical assumptions and typically rely on linear relationships, which restrict their ability to adapt to rapidly changing environmental conditions and the diverse factors influencing energy production<sup>5–7</sup>. Such factors as air density, atmospheric pressure, temperature, local landscape, and wind speed influence wind power generation. All these various interactions between meteorological elements make it challenging to describe them statistically. Similarly, solar power depends on cloud cover, solar position, air temperature, dust collecting on panels and how they are mounted, so that it can cause real challenges for accurate weather forecasting. As these models do not account for the detailed and nonlinear ways these things influence one another, they are less helpful in balancing supply and demand and organizing energy storage, both vital duties in grid management and energy planning<sup>8,9</sup>. Over the years, AI and DL have shown to be strong replacements for these standard approaches, providing benefits in working with data that is difficult to analyze and contains hidden, complex connections without human help in designing features<sup>10</sup>. Classical methods are less suitable for handling high-dimensional, multivariate data, while DL algorithms manage it easily, making them well-suited for renewable energy systems<sup>11,12</sup>. Using Deep Learning, examples of popular networks for this task are Long Short-Term Memory Networks (LSTMs), Gated Recurrent Units (GRUs) and Temporal Convolutional Networks (TCNs). LSTMs, a type of RNN, were created to solve the vanishing gradient problem experienced by general RNNs, making them ideal for catching long-lasting connections in sequence<sup>13,14</sup>. Because of this feature, it becomes possible to predict future energy production from renewable sources using similar weather patterns, seasonal indicators and what was consumed in the past. Unlike RNNs, TCNs avoid using sequential bottlenecks by dilated convolutions and placing convolutional layers. This architecture allows TCNs to process a whole input sequence at once, encoding the sequence far more efficiently than deep layer networks. Additionally, using TCNs ensures stable updates and allows the network to remember both short-term and long-term phenomena, eliminating the requirement for special gates<sup>15</sup>. Moreover, these models can find useful representations in the raw data, so they do not require much manual work from experts to choose these features. As a result, the predictions become more accurate and the models can be used successfully for various energy tasks like forecasting wind and solar energy, predicting how much power the grid will need and optimizing its management. Even so, learning from deep models is tough because it requires substantial training data, lots of computing power and accurate settings adjustments. Therefore, many experts now depend on metaheuristic algorithms to get the most out of data and boost accuracy in predicting future events. Thanks to advances in software and AI, the technique of renewable energy forecasting is improving, making energy predictions more accurate, reliable and suitable for wider use<sup>16</sup>. Still, there are significant problems that remain despite the improvements made. Data from the environment is known to be both high-dimensional and full of noise, which often harms the quality of the built model and encourages overfitting. The feature selection process removes unnecessary data, simplifies the model, and improves performance. As a result, deep learning models can be applied more easily to real-world situations since they become more accurate, efficient and space-saving. Several methods for selecting features have been introduced, including statistical techniques, analysis using mutual information and embedded ones such as L1 regularization. To boost the performance of deep learning models in making predictions, scientists have started to use metaheuristic optimization algorithms that copy nature's processes to solve challenging problems and discover solutions that are very close to the best. Alternatives such as Particle Swarm Optimization (PSO)<sup>17</sup>, Genetic Algorithms (GA)<sup>18</sup>, and iHow Optimization Algorithm (iHOW) work better than traditional methods, since those methods may miss the best result by getting stuck in local minima. With the help of deep learning algorithms, one can adjust the settings, optimize the network and improve the resistance of the model. Even so, a difficult task is to connect selecting the right features with metaheuristic methods in forecasting green energy. Model architecture and optimization are examined separately by researchers, rather than examining their interrelationship. This paper focuses on narrowing this gap by creating a robust framework that links feature selection, metaheuristic optimization and advanced deep learning methods. The study aims to make Temporal Convolutional Networks (TCNs) perform better in forecasting time-series data, since they are efficient and effective at handling long dependencies in data.

The key contributions of this study are as follows:

- A novel framework for forecasting renewable energy demand is proposed, which integrates a custom metaheuristic algorithm—the iHow Optimization Algorithm (iHOW)—with advanced deep learning architectures such as Dynamic Temporal Convolutional Networks (DTCNs) to deliver highly accurate predictions of green energy output.
- A combination method using feature selection and finding the best hyperparameters was presented, making the model more understandable, accurate and straightforward.
- Thorough Evaluation: By running numerous experimental trials, it was shown that the proposed iHOW-optimized methods performed better than other high-level deep learning algorithms on different forecasting measurements with no need for hand-picked parameters.
- Scalability for High-Dimensional Time-Series Data: By using sophisticated feature selection approaches, it was possible to address the significant difficulties associated with high-dimensional, noisy data, which are typical of renewable energy systems. This allowed for flexible and scalable energy forecasting.
- Integration of Real-World Data: Large, real-world green energy datasets were used to verify the efficacy of the suggested framework and guarantee its usefulness in various energy scenarios.
- Establishing a strong basis for combining real-time data streams, multi-source data fusion, and hybrid ensemble learning paved the path for the development of intelligent energy forecasting systems of the future.

The rest of the paper is set up in this way. The second section gives a detailed overview of research in renewable energy forecasting, feature selection and metaheuristic optimization. Next, the data that will be used and how it is prepared for the model are carefully explained. Here, the authors specify the deep learning techniques and optimization methods they applied, including feature selection integration. After describing the setup and evaluation in the next section, a detailed analysis of the outcomes is performed, including a performance comparison and how effective the approach was. Finally, the paper summarizes the main points, discusses where they can be used, and suggests ways for future studies.

## Literature review

Using renewable energy resources in power systems plays a key role in providing electricity for the world and lessening climate change impacts. Hydrogen is among the renewables, and it can help a lot with the transition to clean energy systems because it is versatile. Hydrogen is essential in sustainable energy strategies because it is clean, serves as fuel and energy and is used for storage. Yet, because there are several methods for producing hydrogen and they have different prices, success rates and environmental impacts, comparing and perfecting them becomes challenging. Advances in various models, optimization and modern technologies are helping speed up the use of hydrogen globally as we move toward an energy transition. The literature review gathers and summarizes the most recent research and best methods associated with decision-making in hydrogen production, using existing infrastructure and their development worldwide. EVOA came about to enhance the controller used in wind power plants with DFIGs to ensure efficient use of electricity, depending on the varying wind speed<sup>19</sup>. Moreover, connecting PV systems to utility grids is seen as a sustainable power generation option, thanks partly to AI for optimizing the amount of power generated<sup>20</sup>. Techniques from reinforcement learning are being adopted for hydropower reservoirs to achieve higher efficiency and sustainability in hydrogen production systems<sup>21</sup>. At last, PSCs are considered bright in photovoltaics due to how efficiently and cleanly they generate electricity<sup>22</sup>, potentially offering a low-priced route to produce hydrogen by solar electrolysis. All in all, these achievements confirm that hydrogen's multifunctional role helps reduce carbon emissions and successfully merge renewables into large-scale systems<sup>23,24</sup>. Going further, future progress in hydrogen, materials and energy systems is necessary for building a sustainable global energy system. Because of increased attention on renewable energy, more research has been done on accurate forecasting, mainly ML and DL methods. Precise forecasting is required for easy grid operation, good energy management and incorporating renewable sources into old power grids. It reviews recent growth in renewable energy predictions, mentions the best aspects and gaps in popular models and describes the main barriers for further improvement. Forecasting renewable energy is now essential because more renewable sources are being added to the power grid. Although traditional statistical tools serve specific purposes, dealing with the complex and changing RES data remains difficult. Because of this, new methods based on ML and DL have been applied because they can learn from data that follows complex patterns. Recent studies have found that ML and DL models work more effectively and flexibly than the traditional methods found in the energy sector. The paper further reviewed obstacles in interpreting models, obtaining required data and establishing strong forecasting approaches to let RES help the electricity grid move towards sustainability<sup>25</sup>. A research study was conducted to apply deep learning techniques for estimating green energy output in Asia. Authors created the Green-electrical Production Ensemble (GP-Ensemble) using CNNs, GRUs and FNNs to increase the accuracy of their forecasts. GP-Ensemble performed better than single models, including GRU, FNN and CNN and also outperformed classic ensembles, obtaining MSE of 0.0631, MAE of 0.1754 and RMSE of 0.2383. This method highlights that hybrid deep learning can improve forecasting renewable energy, providing key ideas for planning and managing power grids in fast-developing regions<sup>26</sup>. Recent studies have reinforced the value of hybrid deep learning and optimization models in renewable energy forecasting. Li et al.<sup>27</sup> proposed a hybrid model that combines wavelet packet decomposition (WPD) with long short-term memory (LSTM) networks for short-term photovoltaic (PV) power forecasting. The method decomposes PV time series into subcomponents and forecasts them using separate LSTMs, achieving improved accuracy and stability over standalone deep learning models. Mubarak et al.<sup>28</sup> introduced an LSTM-Attention model with time encoding to forecast one-hour-ahead active and reactive household power consumption. Their model emphasizes interpretability using SHAP analysis and demonstrates superior performance over classical models, suggesting its applicability for intelligent power management systems.

From the perspective of hybrid optimization, Adegboye et al.<sup>29</sup> developed a Worst Moth Disruption Strategy (WMFO) to improve the original Moth Fly Optimization algorithm. When integrated with a multi-layer perceptron (MLP) for CO<sub>2</sub> emission prediction, the WMFO-MLP model achieved outstanding accuracy and robustness, validated through statistical testing.

Similarly, Almsallti et al.<sup>30</sup> proposed the Red-Billed Blue Magpie Optimizer (RBMO) to enhance the performance of extreme learning machines (ELMs) in predicting carbon emissions. Their RBMO-ELM model outperformed traditional hybrid algorithms across benchmark functions and real-world datasets, highlighting its potential in sustainability modeling. In the area of marine energy, Neshat et al.<sup>31</sup> developed a Meta Wave Learner—a deep gradient boosting framework combining multiple convolutional surrogate models—to predict wave energy output from farms across Australia. The method demonstrated high accuracy and transferability across multiple geographic locations. Finally, Beshal et al.<sup>32</sup> proposed a novel Polar Lights Salp Cooperative Optimizer (PLSCO), integrating elements of PLO, CSO, and SSA, for predictive maintenance in manufacturing. When combined with ELM, their PLSCO-based framework achieved 95.4% prediction accuracy and delivered interpretable insights through feature importance analysis. These studies collectively reinforce the significance of hybrid optimization-deep learning frameworks for energy prediction tasks and further support the design choices made in the proposed iHOW-DTCN system (Table 1). A study focused on microgrid energy management investigated how multiheaded convolutional LSTM and particle swarm optimization (MHCLSTM-PSO) could forecast both wind speed and solar irradiation. The authors say this approach combines deep learning and community management,

References	Objective	Methodology	Key findings
19	Optimize wind power plant performance.	Energy Valley Optimizer Approach (EVOA) with adaptive fuzzy logic controllers.	Improved control of grid-tied doubly fed induction generators (DFIGs), enhancing system stability and efficiency.
20	Integrate photovoltaic systems into power grids.	AI-enhanced integration of PV systems.	Demonstrated sustainable power generation with optimized energy output.
21	Optimize hydropower reservoir operations.	Reinforcement learning (RL) for operational optimization.	Improved efficiency and reliability of hydropower systems, supporting hydrogen production.
27	Improve PV power forecasting accuracy.	Hybrid WPD-LSTM model for short-term PV power forecasting.	Demonstrated superior performance compared to LSTM, RNN, GRU, and MLP models using real PV data from Alice Springs.
29	Enhance carbon emission prediction.	WMFO-MLP model combining Worst Moth Fly Optimization with Multi-Layer Perceptron.	Achieved 97.8% accuracy and identified key features like oil efficiency and economic growth in emission forecasting.
28	Forecast active/reactive household power.	LSTM-Attention model with explicit time encoding and SHAP analysis.	Outperformed traditional models with high accuracy; improved interpretability for demand response strategies.
30	Predict CO <sub>2</sub> emissions in industrializing economies.	RBMO-ELM combining red-billed magpie optimizer and Extreme Learning Machine.	Achieved superior R <sup>2</sup> and convergence; identified GDP and globalization indicators as key factors.
31	Forecast wave energy output in coastal regions.	Meta Wave Learner using deep gradient boosting and convolutional surrogates.	Demonstrated 84–90% accuracy across multiple Australian wave farms, improving robustness in marine energy forecasting.
32	Improve predictive maintenance in manufacturing.	PLSCO hybrid optimizer with Extreme Learning Machine (ELM).	Achieved 95.4% accuracy; highlighted torque and tool wear as dominant indicators of machine failure.
22	Improve photovoltaic efficiency.	High-efficiency perovskite solar cells (PSCs).	Demonstrated cost-effective hydrogen production via solar-powered electrolysis.
23,24	Analyze hydrogen's role in energy systems.	Comprehensive analysis of hydrogen as an energy carrier.	Highlighted hydrogen's critical role in decarbonization and renewable energy integration.
25	Improve renewable energy forecasting.	ML and DL methods for time-series analysis.	Identified challenges in data availability, model interpretability, and uncertainty handling.
26	Enhance energy forecasting accuracy.	GP-Ensemble model integrating CNN, GRU, and FNN.	Outperformed traditional methods in MSE, MAE, and RMSE, demonstrating superior forecasting performance.
33	Optimize microgrid energy management.	Multiheaded convolutional LSTM-PSO model for wind speed and solar irradiance forecasting.	Achieved 93.54% accuracy, outperforming conventional methods.
34	Improve PV-hydrogen system efficiency.	MLP and Random Forest for hydrogen production forecasting.	Demonstrated effective hydrogen production forecasting with high R <sup>2</sup> values.

**Table 1.** Summary of key research areas in hydrogen and renewable energy systems.

resulting in better forecast accuracy. Our model was much more accurate than conventional approaches such as CNN (72.52%), LSTM (78.16%) and CLSTM (85.56%). It emphasizes that combining distributed energy resources and new forecasting approaches adds to the efficiency and sustainability of current energy systems<sup>33</sup>. A close examination of PV systems in Morocco considered the way solar energy, environmental temperature and green hydrogen go together. Researchers examined monocrystalline, polycrystalline and amorphous PV panels for nine years and found that their production performance varied widely. Energy assessments found that polycrystalline panels gave the most excellent hydrogen yield, and amorphous panels tended to exhibit the most unstable efficiency. The results indicated that both SVR, Random Forest and MLP worked effectively to predict hydrogen production, with MLP obtaining the best R<sup>2</sup> measure of 0.9430 for monocrystalline panels. As a result, weather and environmental impacts greatly affect the PV-hydrogen system's performance, and new adaptive models are necessary to help maximize energy harvest under changing circumstances<sup>34</sup>.

### Research gap

The literature highlights a sharp growth in studies that look at renewable energy forecasting using ML, DL and hybrid models. LSTM, GRU and TCN have provided excellent results in modeling and representing information from time series over a long period. Studies such as those by<sup>25,26</sup> report that DL hybrid models are more accurate and reliable than statistics-based methods. Even so, these contributions cover a lot, but they still leave some gaps. Firstly, even though GP-Ensemble and MHCLSTM-PSO increase accuracy, they seldom have built-in strategies to automatically change model settings and select the best features simultaneously. A majority of the frameworks either concentrate on creating a mathematical model or working on optimization, but they are rarely designed to achieve both. The result is that models are not as efficient, versatile or adaptable to the frequently high-dimensional and noisy data used in green energy forecasting. Traditional ways of forecasting and ML/DL approaches do not clearly explain their results. As pointed out in<sup>34</sup>, although MLP and Random Forest perform well predicting hydrogen production, they fail to explain which model parameters are the most important. Trust and proper use depend on explanation for applications in critical sectors such as energy infrastructure, but many modern approaches often neglect it. Also, we see that there is underuse of cognitive-based metaheuristic algorithms. Many studies<sup>17,18</sup> that use PSO and GA algorithms for hyperparameter tuning tend to end up at local optima soon and have trouble doing both exploitation and exploration simultaneously. Despite the growing interest in metaheuristic optimization, few studies have investigated the iHow Optimization Algorithm (iHOW), a cognitively inspired method that leverages learning mechanisms and knowledge-driven search to enhance solution quality and convergence performance. So far, RL has achieved good results in improving hydro operations<sup>21</sup>, but joining it with performant DL methods for prediction is still not common. Green hydrogen research also suggests interest in cleaner energy solutions, though no strong models are available to help forecast it anywhere consistently. There is a missing link between popular forecasting methods and actual multivariate

and seasonal data. Most studies are based on limited or fake datasets. Still, we need environmental forecasting tools that can cope with all kinds of information (e.g., solar irradiance, wind speed, temperature) and deal with changes in weather over time. This work aims to address these issues by suggesting a single forecasting approach that brings together:

- Another element is advanced deep learning architecture (Dynamic Temporal Convolutional Networks, DTCN).
- This study uses iHOW for optimization, a cognitively inspired algorithm that employs learning and knowledge-based search strategies.
- Feature selection as a way to lower the number of dimensions.
- Understanding AI with the help of SHAP analysis. Real-world green energy data were also used in testing the models.

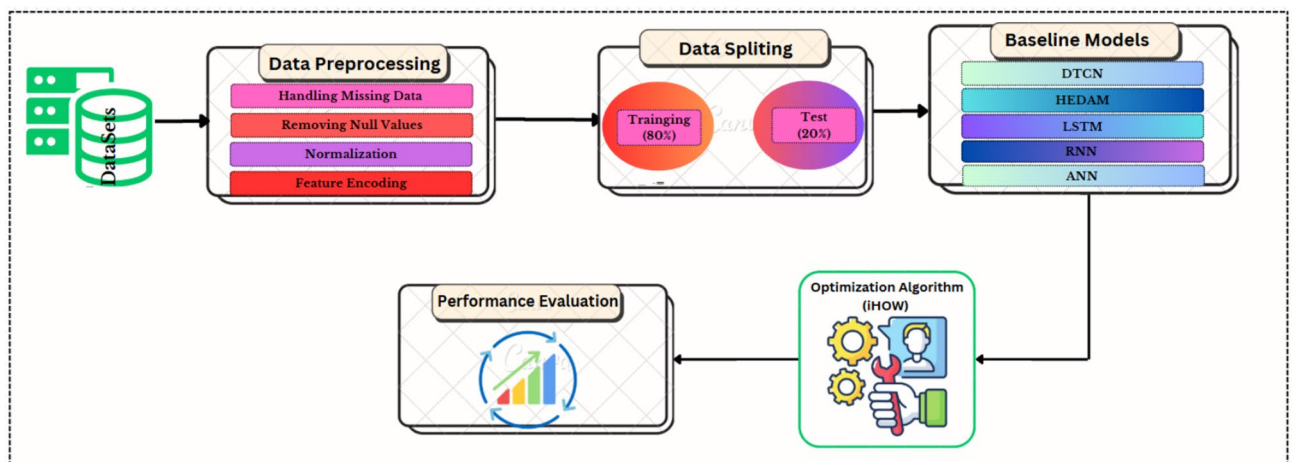
Using these components together, the proposed system addresses known scalability, interpretation and optimization challenges, making it new and suitable for green energy forecasting.

## Materials and methods

To carry out this study, a framework was designed to step-by-step examine the integration of deep learning and metaheuristic optimization for improving green energy forecasts. This method requires effective data preprocessing, creating practical features and enhancing the model for predictive analysis to achieve better accuracy, stability and application of the model. The framework illustrated in Fig. 1 involves preprocessing, splitting the data, training the baseline model, evaluating its performance and then using the iHOW algorithm to optimize its hyperparameters. All the pieces are created to resolve the specific challenges seen in data with many dimensions and time stamps standard in renewable energy, making the forecasts reliable and trustworthy. To begin the process effectively, you must handle missing data, remove null entries, adjust the scales of different features and assign numbers to various categories. Doing this helps reduce errors caused by too much data, reduces fitting only to the data you have and increases the ability of your model to solve new problems. Preprocessed data is generally split so that 80% becomes the training set and 20% is the testing set. Subsequently, the baseline deep learning models Dynamic Temporal Convolutional Networks (DTCN), Hybrid Encoder-Decoder Attention Model (HEDAM), Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN) and Artificial Neural Networks (ANN), are trained using the dataset that was prepared. Using these models, one can initially judge how effective the optimization strategies are. Next, the iHOW algorithm is used to change the model parameters, boosting its accuracy and stability.

## Dataset description

This study utilizes the Green Energy Demand dataset as the primary data source for training and validating the proposed machine learning models. The dataset spans a ten-year period, from 2008 to 2018, and contains hourly energy usage records. This high-resolution temporal data enables detailed observation of demand fluctuations across different seasons, regions, and activity types. The dataset includes variables such as regional identifiers and energy types, which reflect diverse climatic conditions and consumption behaviors. This comprehensive coverage is crucial for identifying regular patterns in hourly and daily usage, as well as for capturing long-term developments in renewable energy adoption. The dataset is structured as a time series, where each record represents an hourly energy demand value—typically measured in megawatt-hours (MWh) or gigawatt-hours (GWh). Given the continuous nature of the target variable, the forecasting task is framed as a regression problem. To ensure proper model evaluation, the dataset is split into training and testing subsets. Specifically, the models are trained using historical data from 2008 to 2018 and tested on unseen data from 2019 to 2021. This temporal

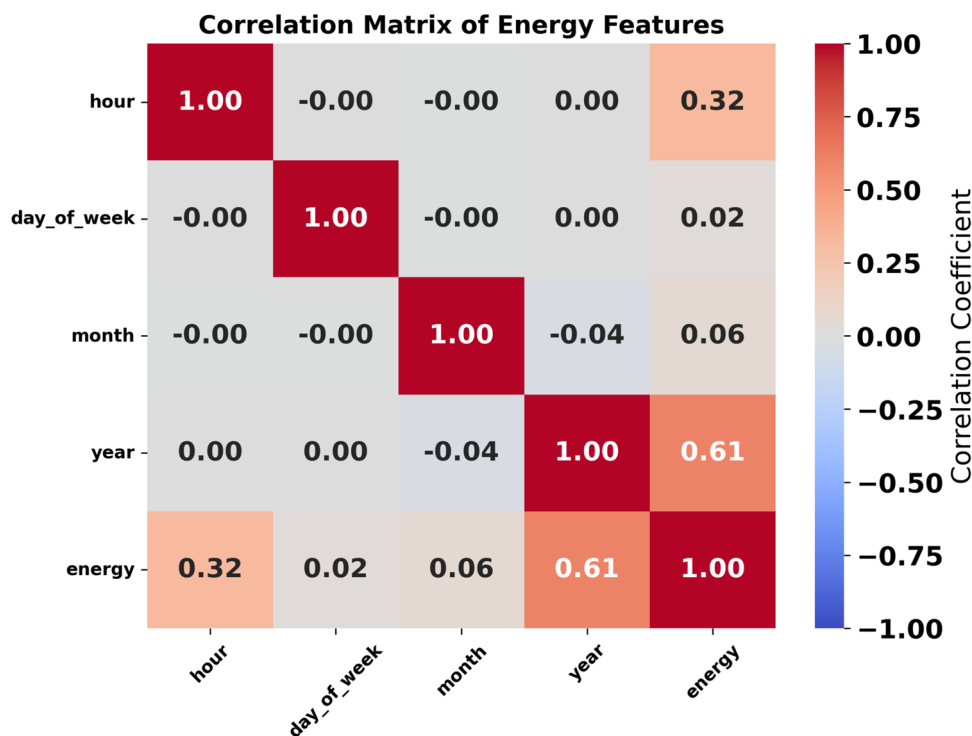


**Fig. 1.** Methodological framework for green energy demand forecasting, including data preprocessing, model training, and optimization.

separation mimics real-world forecasting scenarios, where models must generalize to future, unseen conditions. The complete dataset comprises approximately 105,000 hourly entries per region, totaling over 1 million observations across all geographic zones and energy types. Each sample includes a timestamp, region identifier, energy demand value, and energy source classification (e.g., wind, solar, hydro). To enhance the model's learning capacity, several engineered features were introduced, such as lagged variables (`lag_1`, `lag_2`, etc.), rolling window statistics (e.g., 24-hour moving average and standard deviation), and periodic features like `sin_hour` and `cos_hour` to capture daily seasonality. Additional categorical transformations included calendar-based indicators (e.g., `day_of_week`, `month`, `season`) to reflect external temporal drivers. After preprocessing, the final input space consisted of 28 features. Missing values were addressed using forward-fill imputation, and all continuous attributes were normalized via Min-Max scaling to the [0, 1] range. Low-variance features and those with high collinearity were excluded using a combination of variance thresholding and correlation filtering. This robust preprocessing pipeline ensured that only informative, non-redundant signals were passed into the model. An exploratory data analysis (EDA) was conducted to examine the key characteristics of the dataset. This process included visualizations and statistical summaries that revealed temporal patterns and feature interactions. Correlation analysis was applied to quantify the linear relationships between temporal features—such as hour of the day, day of the week, month, and year—and energy demand. As shown in the correlation matrix (Fig. 2), energy consumption is most strongly correlated with the hour and year variables, suggesting that both diurnal cycles and long-term trends significantly influence energy usage. In contrast, weaker correlations were observed with the day of the week and month, indicating these variables have a lesser impact.

Understanding the underlying components of a time series is crucial for effective modeling, forecasting, and interpretation of temporal data. Figure 3 illustrates the seasonal decomposition of energy consumption over time, allowing for a detailed inspection of the constituent elements that influence the observed values. The original time series, labeled **Observed**, is depicted in the top panel and reflects the total measured energy values over a span of roughly ten years. This plot serves as the foundation for further decomposition and analysis. Beneath the observed data, the second panel presents the **Trend** component, which captures the long-term progression in energy consumption. This trend reveals gradual growth over time, potentially driven by macroeconomic factors, technological development, or population growth, and helps in isolating the non-random structural changes in the series. The third subplot illustrates the **Seasonality** component. This segment displays regular and repeating patterns that occur over a fixed period (e.g., monthly or yearly), indicative of cyclical usage behavior such as increased energy demand during specific seasons. Identifying and quantifying seasonality is critical in improving forecast accuracy and understanding recurrent phenomena. Finally, the **Residuals** component, shown in the bottom panel, captures the irregular or random variation in the data that cannot be explained by the trend or seasonal patterns. These residuals represent noise or anomalies and are crucial for diagnosing model adequacy and identifying outlier behavior.

In summary, the exploratory analysis underscores the temporal complexity of energy consumption data. The findings guide the feature engineering process and inform the selection of machine learning algorithms capable of capturing long-term and short-term dependencies inherent in green energy demand.



**Fig. 2.** Correlation matrix of energy features.

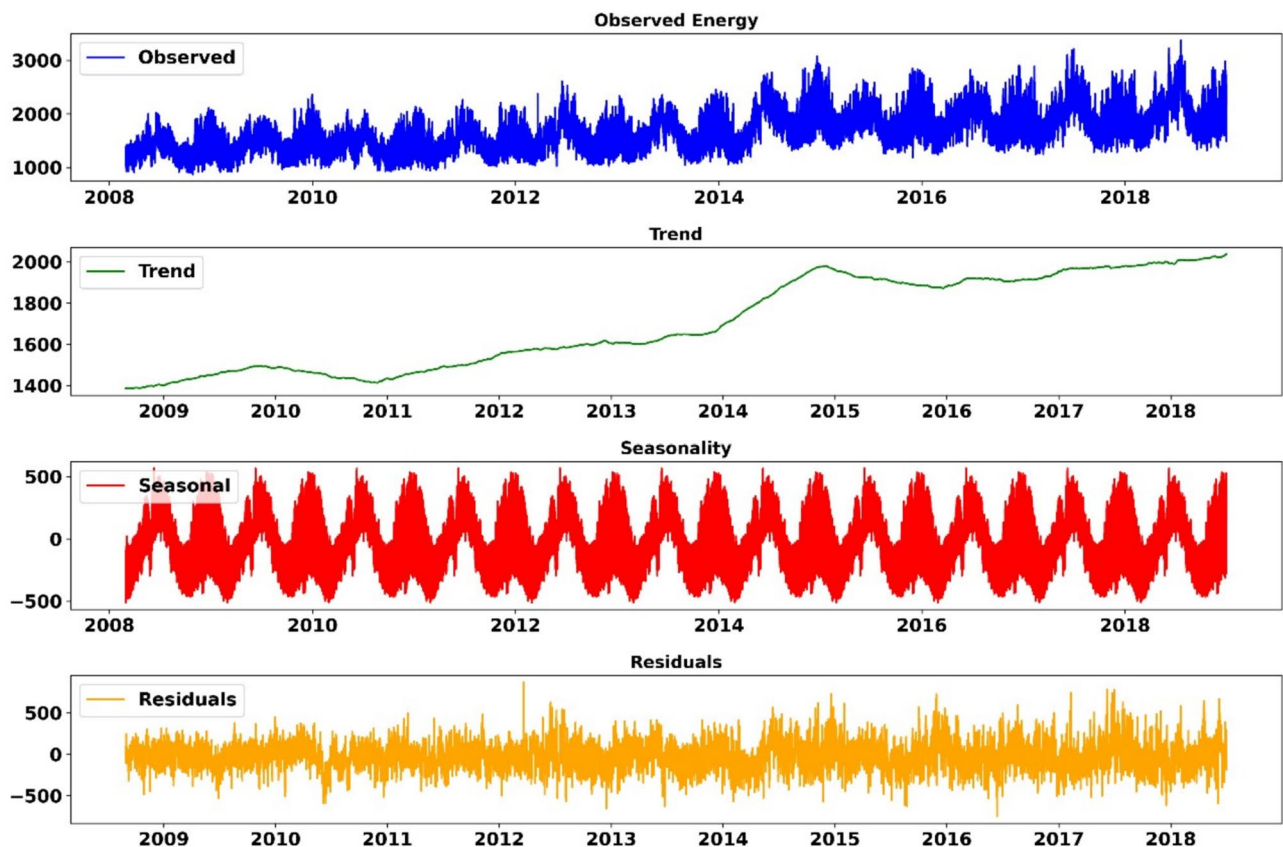


Fig. 3. Seasonal decomposition of energy time series into observed, trend, seasonal, and residual components.

### Data preprocessing

Adequate data preparation must be carried out during the analytical process for a forecasting model to be accurate and reliable. Data missing values can occur for several reasons, so handling them is essential before further analysis. If some values are missing, this may introduce issues and lessen the quality of training data. Any NaN (Not a Number) and infinite values found were removed throughout the study to keep the training process consistent. The cleaning examined the information row by row and column by column to prevent incomplete and imperfect data.

Besides data cleaning, using enhanced features helps to boost the performance of deep learning algorithms. Time-related features were created from the timestamps to highlight the repeated and periodic energy use. The hour, day, month, year and week are all engineered to record how energy needs vary based on the time and day of the week. Daily features allow you to reflect on the morning and evening rush in the houses. With weekly and monthly features, you capture trends linked to working hours, weather throughout the year and seasons in various industries.

In addition to basic temporal decomposition, a structured feature engineering process was applied to extract meaningful patterns from the raw `datetime` and `energy` columns. Specifically, lag features (`lag_1` through `lag_24`) were generated to capture the hourly temporal dependence of energy usage. Furthermore, rolling statistics such as `rolling_mean_24h`, `rolling_std_24h`, `rolling_skewness`, and `rolling_kurtosis` were calculated over a 24-hour window to reflect local trend, volatility, and distributional shape. Additional variables like `cumulative_energy`, `smoothed_energy`, and `energy_diff` were derived to capture the long-term buildup and short-term fluctuations in energy consumption. Cyclical features like `sin_annual` and `cos_annual` were added to preserve the periodicity of seasonal patterns. These engineered features enriched the dataset with contextual signals, boosting model interpretability and predictive performance.

Following this step, feature selection was conducted based on correlation filtering and SHAP value analysis. Features that exhibited either high redundancy or low predictive contribution were excluded. The final set of selected features included: `energy`, `lag_1`, `lag_2`, `lag_3`, `lag_22`, `lag_23`, `lag_24`, `rolling_mean_24h`, and `cumulative_energy`. This refined set was used in model training and SHAP explainability analysis.

The input features were normalized to get the best results and avoid variation during modeling. The impact of numerically prominent features can bias the learning process and go much more slowly in some parts of the model. To ensure all the features' scales were equal, Min-Max scaling was used, and each value was changed to fall between 0 and 1, as stated below:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

When  $X$  is the original feature value,  $X_{min}$  is the smallest number seen for that feature, and  $X_{max}$  is the most significant number we have for it. Merging small datasets into a larger set ensures the data is not skewed and decreases the danger of unstable computations, fast growth of gradients, and excessively fitting the model.

The change in energy demand throughout the year was given particular attention to boost the accuracy of the forecasts. Energy usage tends to fluctuate based on weather changes, shifts in the economy and the customs of societies. Residential energy use surges in winter and summer when people require heat or air conditioning. In contrast, the industry's energy needs might change depending on when industrial products are produced. Seasonal components were examined in the data to analyze long-term trends, and valuable features were incorporated to help the models predict these changes every year. As a result, models can see short-term and long-term movements in the market, leading to better and more reliable predictions.

To ensure the data was evaluated reliably, it was split into training and testing sections. Models were trained using data from 2008 and tested or validated against data from 2019 to 2021. Having to switch to new periods in the simulation models the process of testing forecasts for future times in real life. The engineered temporal features in the processed and enriched dataset help define the deep learning approaches and metaheuristic optimization models used later on. Thanks to this strategy, the models are trained using semantically enriched and informative inputs, improving the accuracy and reliability of their green energy predictions. To build a comprehensive and robust feature space for predictive modeling, both original and derived features were considered. The original dataset included temporal and contextual variables such as hour, day of the week, month, and energy consumption. Additional features were created through various feature engineering techniques including lag features, rolling statistics, cumulative metrics, and cyclic transformations to capture seasonality. Table 2 summarizes all features that were generated prior to the feature selection process.

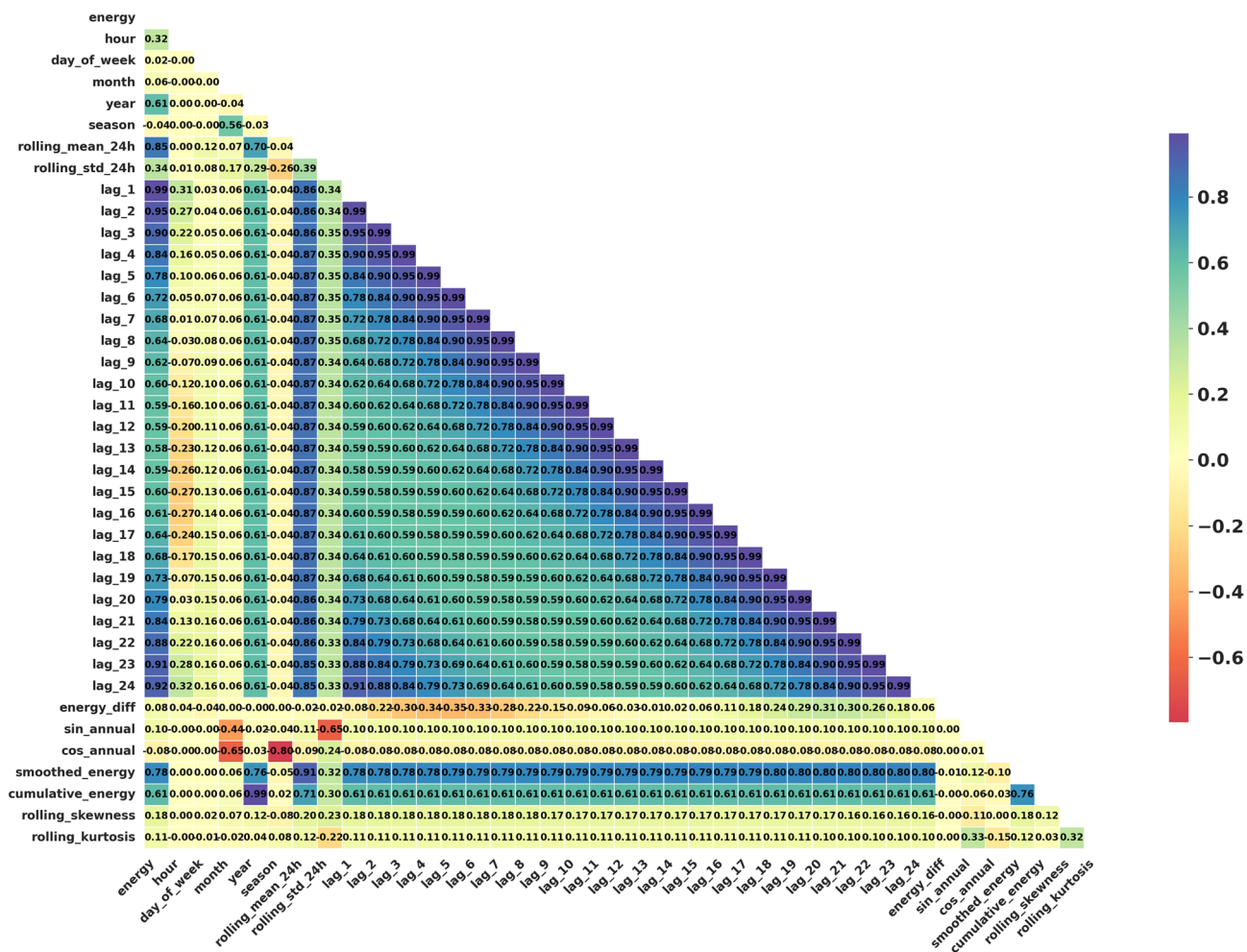
Understanding the interrelationships between features is a foundational step in the development of robust predictive models, particularly in time-series forecasting tasks. To this end, an enhanced feature correlation heatmap was constructed to evaluate the linear correlations among the engineered features within the dataset. This visualization enables the identification of potential redundancies, multicollinearity, and the degree of information overlap between features. As depicted in Fig. 4, the heatmap encompasses both raw features (e.g., hour, day of week, and seasonal indicators) and derived features (e.g., lag values, rolling statistics, and trigonometric seasonal encodings). Notably, a strong autocorrelation structure is observable among the lagged energy values (lag\_1 through lag\_24), suggesting significant temporal dependency in energy consumption data. Conversely, variables such as “month” and “day\_of\_week” exhibit minimal linear correlation with the target variable “energy”, which may warrant further investigation regarding their inclusion in downstream modeling tasks. By leveraging this correlation analysis, practitioners can refine their feature selection strategy to improve model generalizability and mitigate issues such as overfitting or multicollinearity. The enhanced heatmap thus serves as both a diagnostic and explanatory tool in the broader feature engineering pipeline.

To mitigate the risk of data leakage during temporal feature engineering, we strictly adhered to causal processing rules. All lag-based features (e.g., lag\_1, lag\_24), rolling statistics (e.g., rolling\_mean\_24h, rolling\_std\_24h), and cumulative measures were computed using only past data up to each time point. No future information was included when generating features for either the training or validation set. Furthermore, the cross-validation framework was designed to preserve temporal order in each fold, ensuring that features used during training were not influenced by values from the corresponding validation or test windows. This rigorous handling of time-aware features eliminates any leakage risk and ensures the integrity of the forecasting evaluation.

Feature name	Description
energy	Target variable (energy consumption)
hour	Hour of the day (0–23)
day_of_week	Day of the week (0–6)
month	Month of the year (1–12)
year	Calendar year
season	Categorical season indicator (Winter, Spring, etc.)
lag_1 to lag_24	Energy consumption at previous 1 to 24 hours
rolling_mean_24h	24-hour rolling mean of energy
rolling_std_24h	24-hour rolling standard deviation
energy_diff	Difference in energy from previous hour
cumulative_energy	Cumulative energy consumption
smoothed_energy	Smoothed series using exponential smoothing
sin_annual, cos_annual	Sine and cosine seasonal encodings for yearly seasonality
rolling_skewness	Rolling skewness over time
rolling_kurtosis	Rolling kurtosis over time

**Table 2.** Complete set of features after preprocessing and feature engineering.

### Enhanced Feature Correlation Heatmap



**Fig. 4.** Enhanced feature correlation heatmap showing the Pearson correlation coefficients among raw and engineered features.

In order to better understand the relative importance of the input variables in shaping the model's predictions, we employed the SHAP (SHapley Additive exPlanations) framework, which provides a unified measure of feature contribution to the output. SHAP values decompose each prediction into the sum of the contributions of individual features, thus offering both global and local interpretability of the model's behavior. Figure 5 illustrates the SHAP summary plot, where the features are ranked by their overall importance and the distribution of their impacts is visualized. The horizontal spread of each feature indicates the magnitude and variability of its influence on the model output, while the color gradient represents the original feature values, ranging from low (blue) to high (red). Such a representation allows us to identify not only which features dominate the predictive process, but also the directional effect that high or low feature values exert on the target variable.

#### Deep learning models

Deep learning successfully identifies the complicated features in time-series data, which is why it performs well in forecasting renewable energy. Here, I will lay out the deep learning architectures used in this study, along with their main features, good sides, and why they were chosen. The models are meant to handle the problem of learning for a long-term period, representing features in motion and working with sequence tasks. These issues are vital for accurate forecasting of energy demand. In this study, the baseline models used for deep learning are Dynamic Temporal Convolutional Networks (DTCNs), Hybrid Encoder-Decoder Attention Models (HEDAM), Long Short-Term Memory Networks (LSTMs), Recurrent Neural Networks (RNNs) and Artificial Neural Networks (ANNs). These models can be helpful in predicting future values in time series, thanks to how they organize time-specific data and represent various features.

- *Dynamic Temporal Convolutional Networks (DTCNs)* They are an effective competitor to traditional recurrent models, because they perform time-series prediction using convolutions and focus on high performance and considering future events. While regular RNNs and LSTMs consider one section at a time, DTCNs use 1D

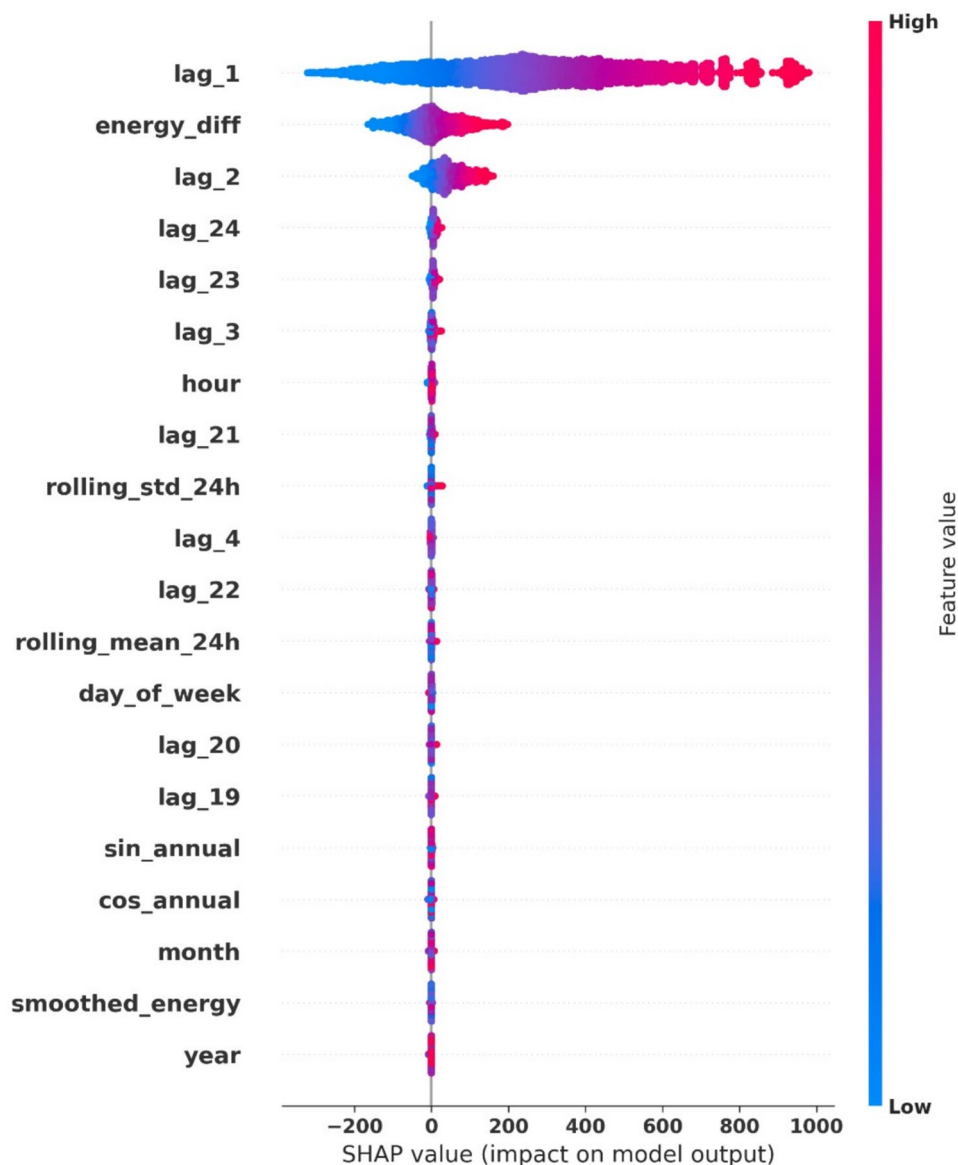


Fig. 5. SHAP summary plot showing the impact of features on model output.

convolutions with dilation to take in an entire sequence and recognize both short and long changes within it. Because of the design, many tasks can be done simultaneously, which cuts down training time. Additionally, because they are non-recurrent, DTCNs do not face much risk of either gradients reducing or growing too large<sup>35</sup>. As these networks are built for high bandwidth data, they can be handy for forecasting green energy, as capturing the many changing energy patterns over time is vital for accurate results.

- **Hybrid Encoder-Decoder Attention Models (HEDAMs)** Hybrid Encoder-Decoder Attention Models (HEDAMs) integrate encoder-decoder architectures with attention mechanisms specifically designed for sequential data processing. In this structure, the encoder transforms the input sequence into a latent representation, while the decoder generates the output sequence by selectively attending to relevant encoded information at each time step. The incorporation of attention allows the model to dynamically prioritize informative temporal dependencies, enhancing its capacity to model long-range contextual relationships. Owing to these characteristics, HEDAMs are particularly effective in time-series forecasting tasks, where capturing complex, non-linear temporal dynamics is essential—such as in the context of renewable energy demand prediction<sup>36</sup>.
- **Long Short-Term Memory Networks (LSTMs)** These models are optimized RNNs to deal with the problem called the “vanishing gradient” problem that affects most standard RNNs. LSTMs selectively control the information by using an input, a forget and an output gate. Since LSTMs can remember long sequences, they are suited for problems where past events affect the results for a considerable time, such as forecasting energy demand related to the weather. Many people use LSTMs in natural language, speech recognition and making financial forecasts, thanks to their excellent memory skills<sup>37</sup>.

- *Recurrent Neural Networks (RNNs)* They are created to structure sequence models and allow processing of sequences of any length, while saving context for past time steps in a hidden state. Traditional RNNs have difficulty learning from data that includes long-term relevant patterns due to the vanishing gradient problem. RNNs still have value because they have a simple design and can deal with sequences of any length. They are applied in processing speech, creating language models and analyzing financial time-series<sup>38</sup>.
- *Artificial Neural Networks (ANNs)* They are simple and include layers of neurons working together to change input data using certain functions. Many ANNs have layers where each neuron in one layer is connected to every neuron in the next layer. ANNs lack the explicit memory stored in LSTMs and RNNs, but they can discover complex nonlinear relationships in data by having many layers. Because ANNs are efficient and straightforward, machine learning practitioners can compare them with other more sophisticated architectures<sup>39</sup>.

Since Dynamic Temporal Convolutional Network (DTCN) performed well in capturing long-term relationships among signals and complex features, this study selected it as the primary model to optimize hyperparameters. Because sequential data is handled via temporal convolutions in DTCNs, they can process this type of data as efficiently as LSTMs or RNNs and offer greater speed and scalability in training. This method best forecasts energy since short-term and long-term changes should be considered for accurate predictions. DTCNs work well at time-series tasks because they reduce computational complexity, use more parallelism and ensure improved gradient flow, compared to traditional recurrent methods. These features suggest that metaheuristic algorithms could enhance DTCNs' performance by choosing suitable hyperparameters for improved forecasting and smooth calculations. As highlighted, DTCNs are the leading choice for the optimization experiments because they are expected to reach the highest level of accuracy in predicting green energy.

#### *Proposed iHow optimization algorithm (iHOW)*

The iHow Optimization Algorithm (iHOW) is a novel human-inspired metaheuristic optimization technique designed to enhance the performance of complex models such as deep learning networks. Unlike nature-based algorithms that emulate animal or physical behavior, iHOW draws on cognitive developmental principles that model how humans process information, learn, and refine knowledge over time. Its design addresses hyperparameter tuning challenges by incorporating a multi-layered reasoning and learning structure to balance global exploration with local exploitation<sup>40</sup>.

Unlike traditional metaheuristics such as PSO, HHO, and GWO that rely on biologically inspired behaviors, iHOW models cognitive and social learning dynamics rooted in human decision-making. The algorithm emphasizes reflective learning, progressive memory updates, and interaction-based knowledge refinement—key aspects often absent in single-phase search strategies.

**Cognitive Architecture of iHOW** The core innovation of iHOW lies in its five-stage cognitive learning structure:

1. *Data Collection:* Solutions collect raw environmental data, simulating human sensory perception. Initial positions are initialized randomly across the search space to represent diverse experiential input.
2. *Learning and Asking:* Solutions modify their behavior by evaluating previous memory and population information, adjusting parameters via adaptive learning rates  $r_1, r_2, r_3$ . This stage models interactive learning, where individuals form updated opinions by questioning existing beliefs and incorporating shared experiences.
3. *Information Processing:* Insights are extracted from the learned data to generate informed movement strategies. Solutions synthesize short-term experiences into meaningful navigation patterns. This step mirrors the cognitive compression of knowledge—transforming raw data into strategic action plans.
4. *Knowledge Acquisition:* Knowledge is updated by integrating processed data with prior information. This stage ensures the retention of beneficial patterns and guides strategic repositioning in the solution space. It functions as a long-term memory formation mechanism, which is critical for search consistency and generalization.
5. *Expertise:* Final-stage solutions rely on cumulative expertise and refine their decisions through enhanced search rules, converging toward optimal results using collective memory and adaptive recombination. This stage formalizes the use of both historical and social memory, enabling the algorithm to shift decisively from exploration to exploitation.

**Mathematical Dynamics of iHOW** The algorithm applies dynamic coefficient updating and knowledge factors to achieve cognitive realism. Position updates are influenced by both the best global solutions and adaptive learning states:

$$X_{t+1} = X_t + \alpha \cdot (X_r - X_t) + \beta \cdot (X_p - X_t)$$

Where:

- $X_t$ : current solution
- $X_r$ : random peer solution
- $X_p$ : best-known solution
- $\alpha, \beta$ : learning rates adaptively tuned during search

This position update rule introduces dual reference dynamics—using both random peers and best-known exemplars to simulate diverse opinion learning. Unlike PSO which depends on inertia and velocity, iHOW’s mechanism is grounded in social learning theory, which encourages adaptive assimilation and divergence.

The knowledge coefficient  $K$  decays over time, promoting exploration in early stages and exploitation later:

$$K = 2 - 2 \cdot \left( \frac{\text{current iteration}}{\text{max iterations}} \right)$$

This nonlinear decay simulates the natural learning saturation process—where individuals gradually reduce openness to new inputs and consolidate their learned strategies. It ensures a smooth transition from stochastic discovery to deterministic refinement.

From a theoretical standpoint, iHOW differentiates itself by incorporating elements of cognitive psychology—such as memory reinforcement, reflective questioning, and knowledge integration—into its core operations. These properties are absent in traditional metaheuristics, which do not account for multi-phase learning dynamics. Furthermore, iHOW’s structure allows for guided diversity, making it less prone to premature convergence and better suited to problems with complex, dynamic landscapes.

The pseudocode in Algorithm 1 describes in detail how the iHOW algorithm functions at its five cognitive layers, from data gathering to reaching convergence. The representation offers clear operational clarity about processes and functions as a blueprinted approach for implementing hyperparameter tuning.

---

```

1: Initialize population size, learning rates  $(r_1, r_2, r_3, \dots)$ , knowledge factor  $K$ , and maximum
   iterations
2: Generate initial population with random positions
3: Step 1: Data Collection
4: for each individual do
5:   Collect raw data  $D$ 
6:   Store for future learning
7: end for
8: Step 2: Learning and Asking
9: for each individual do
10:  Update position using adaptive rates and past memory
11:  Store outcomes for evaluation
12: end for
13: Step 3: Information Processing
14: for each individual do
15:  Extract insights from past iterations
16:  Update population-level information
17: end for
18: Step 4: Knowledge Acquisition
19: for each individual do
20:  Integrate new insights with historical patterns
21:  Refine knowledge parameters  $KS$ 
22: end for
23: Step 5: Optimization
24: for each iteration do
25:   for each individual do
26:    Update position based on knowledge and best solution
27:    if current solution is better than best then
28:      Update  $X_{\text{best}}$ 
29:    end if
30:   end for
31: end for
32: Convergence
33: if maximum iterations reached then
34:  Output  $X_{\text{best}}$  as optimal solution
35: else
36:  Repeat search
37: end if

```

---

**Algorithm 1.** iHow Optimization Algorithm (iHOW)

The iHOW algorithm's cognitive framework enables flexible exploration, contextual learning, and efficient convergence. Its five-phase design draws from established principles in cognitive science, allowing individuals to learn adaptively, interact meaningfully, and converge intelligently. It demonstrates superior performance on benchmark problems by avoiding local minima and reducing hyperparameter sensitivity. Its five-phase design draws from established principles in cognitive science, allowing individuals to learn adaptively, interact meaningfully, and converge intelligently. The integration of iHOW with the DTCN model is motivated by the need for robust and adaptive hyperparameter tuning in complex, non-stationary time series forecasting. Deep Temporal Convolutional Networks (DTCNs) offer powerful temporal feature extraction, but their performance is highly sensitive to architectural and training hyperparameters such as filter sizes, dilation rates, learning rates, and batch sizes. iHOW addresses this by leveraging its cognitively inspired exploration-exploitation mechanism to discover optimal configurations that general optimizers often miss. Unlike conventional tuning methods or biologically inspired algorithms that may prematurely converge or overfit, iHOW's multi-stage reasoning and social learning structure enables more informed, diverse, and convergent search behavior. This synergy allows the forecasting framework to not only achieve higher predictive accuracy but also adapt to evolving data patterns, which is critical for energy systems influenced by seasonal and environmental fluctuations.

In this study, iHOW is applied to optimize hyperparameters in deep learning models, such as DTCNs, for forecasting green energy. The integration of multi-stage memory and decision-making results in robust, adaptive, and highly accurate optimization behavior.

#### *Benchmark algorithms*

To rigorously evaluate the effectiveness of the proposed iHow Optimization Algorithm (iHOW), we benchmark its performance against nine well-established metaheuristic algorithms. These optimizers are widely recognized for their adaptability and effectiveness in hyperparameter tuning and energy forecasting tasks. Each algorithm is inspired by distinct natural or physical processes and provides a different balance between exploration and exploitation in the search space.

- *Harris Hawks Optimization (HHO)*: Inspired by the cooperative hunting strategy of Harris hawks, HHO models intelligent prey pursuit through position updates based on energy and escape probability<sup>41</sup>. It dynamically balances exploration and exploitation to locate global optima efficiently.
- *Grey Wolf Optimizer (GWO)*: Mimicking the social hierarchy and hunting behavior of grey wolves, GWO updates candidate solutions using the positions of alpha, beta, and delta leaders, facilitating a guided and adaptive search process<sup>42</sup>.
- *Particle Swarm Optimization (PSO)*: A population-based optimizer that emulates the social behavior of flocks and swarms. Each particle adjusts its velocity and position using both personal experience and collective knowledge from the population<sup>43</sup>.
- *Whale Optimization Algorithm (WOA)*: Based on the bubble-net feeding behavior of humpback whales, WOA explores and exploits the search space using spiral and encircling mechanisms around promising solutions<sup>44</sup>.
- *Biogeography-Based Optimization (BBO)*: Rooted in the biogeographical distribution of species, BBO applies migration and mutation operators to share features among candidate solutions, improving population diversity and solution quality<sup>45</sup>.
- *JAYA Algorithm*: A parameter-free optimizer that simultaneously minimizes the distance from the best solution and maximizes the distance from the worst, ensuring steady convergence without requiring algorithm-specific parameters<sup>46</sup>.
- *Multi-Verse Optimizer (MVO)*: Inspired by cosmological concepts such as white holes, black holes, and wormholes, MVO enables candidate solutions to probabilistically explore new regions based on fitness and diversity metrics<sup>47</sup>.
- *Free Energy Perturbation Optimizer (FEP)*: Inspired by principles from statistical thermodynamics and molecular dynamics, FEP utilizes the concept of free energy differences between states to guide the search process. By perturbing candidate solutions and evaluating the change in a system's free energy, this optimizer navigates the solution space using thermodynamic probabilities. It is particularly effective in problems where maintaining equilibrium between exploitation and exploration is critical, such as in non-linear, high-dimensional optimization landscapes<sup>48</sup>.
- *Simulated Annealing Optimization (SAO)*: A single-solution metaheuristic based on the physical annealing process in metallurgy. SAO accepts suboptimal solutions early on with a decreasing probability to escape local optima, gradually refining toward the global minimum<sup>49</sup>.

While each of the benchmark algorithms leverages an elegant metaphor-biological, physical, or sociological—they predominantly employ single-layered or static mechanisms for search adaptation. For example, PSO relies on momentum-based position updates, GWO on leader-following strategies, and HHO on prey escape probability modeling. These mechanisms, while effective, do not incorporate learning accumulation or cognitive decision-making. In contrast, iHOW introduces a multi-phase cognitive learning framework inspired by human developmental cognition. It simulates sensory input, questioning, information synthesis, memory consolidation, and strategic expertise - stages that reflect how humans adapt their reasoning over time. This enables iHOW to dynamically shift between exploration and exploitation using memory-guided learning, adaptive recombination, and knowledge decay - capabilities absent in traditional metaheuristics. Thus, the inclusion of iHOW contributes not only an empirical improvement but also a theoretical advancement in the design of adaptive intelligent optimization algorithms.

## Evaluation metrics

Deep learning models and feature selection algorithms should be extensively measured to maintain the forecasting process's accuracy and effectiveness. This research uses several measures to analyze the models' prediction accuracy and the chosen features' value. It's necessary to consider these metrics to check and improve selected models, fix their parameters and verify the correctness of the forecasting model. The deep learning models used for green energy forecasting are checked in three ways: by observing errors, overall performance and how much the results differ from the actual values. They include measurements of the model's accuracy, fit to the data, and ability to predict similar results.

Table 3 lists the main metrics that measure the performance of deep learning models for green energy forecasting. Selecting the best features is an essential step in forecasting because it influences how complex the model is, how long it takes to train and how accurate it becomes. This table covers the main assessment metrics included in this study.

Table 4 highlights the main factors used to examine how effective various feature selection methods are in finding the main features for accurate predictions of green energy.

## Empirical results

This section reviews how deep learning models have been evaluated for their ability to predict green energy output. The study evaluates baseline performance, checks how choosing the right features changes the model, sees the impact of employing various optimized sets of features and comprehensively compares the optimized models that result from various metaheuristic approaches. The results are divided into four key subsections, each focusing on a significant part of the forecasting methodology.

### Baseline deep learning performance

Using baseline deep learning models gives a basic measure for understanding how well different feature selection and optimization techniques work. In this situation, five architectures were examined: Dynamic Temporal Convolutional Networks (DTCN), Hybrid Encoder-Decoder Attention Models (HEDAM), Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN) and Artificial Neural Network (ANN). Tests were carried out using the original features without optimization, setting a line for later improvement. Standard performance metrics are Mean Squared Error, Root Mean Squared Error, Mean Absolute Error, Mean Bias Error, correlation coefficient, Coefficient of Determination, Relative RMSE, Nash-Sutcliffe Efficiency and Willmott Index.

It is evident from Table 5 that DTCN surpasses all other models by having the lowest MSE, RMSE and MAE, and the best  $R^2$  and NSE scores. The strong results demonstrate that convolutional networks are valuable for noticing time-based changes in green energy data, so DTCNs are prime for additional improvement. The chart in Fig. 6 offers a view of how the deep learning models in this study measured up compared to one another. These models are the Dynamic Temporal Convolutional Network (DTCN), Hybrid Encoder-Decoder Attention Models (HEDAM), Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN) and Artificial Neural Network (ANN). The chart displays how these critical measures are spread across values, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Bias Error (MBE), the correlation coefficient ( $r$ ), coefficient of determination ( $R^2$ ) and Relative Root Mean Squared Error (RRMSE). Using a radar plot, the study shows how each forecasting model performs across the diverse evaluation metrics, highlighting their overall results in forecasting renewable energy capacity. A bar chart makes it easier to examine how robust, predictable and general the findings are, which determines the best forecasting model.

You can see in Fig. 7 how the key metrics are distributed for various deep learning models. You should calculate Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Bias Error (MBE), correlation coefficient ( $r$ ), coefficient of determination ( $R^2$ ), Relative Root Mean Squared Error (RRMSE), Nash-Sutcliffe Efficiency (NSE) and Willmott Index (WI). Each box plot in the graphs captures the central tendency, spread and outliers in each set of model results, showing the differences in performance. Representation is mainly helpful for finding models whose performance and accuracy are stable and those whose predictions show more changes. Thanks to the horizontal swarm plots, this figure adds information on how frequent and different the various results are, which refines our appreciation of the model's reliability and robustness.

### Feature selection results

Picking the right features is essential to simplify the model, improve how it is understood and increase the accuracy of predictions. The results for eleven different feature selection algorithms, which include biHOW, bHHO, bGWO, bPSO, bBA, bWAO, bBBO, bJAYA, bbMVO, bFEP and bSAO, are provided in Table 6. Significant numbers are Average Error, Average Select Size, Average Fitness, Best Fitness, Worst Fitness and Standard Deviation of Fitness.

The data presented in Table 6 confirm that biHOW achieved the most minor average error, highest fitness and lowest standard deviation, proving it is robust and reliable for selecting useful features. How metrics are distributed across several metaheuristic algorithms tells us much about their general efficiency and stability. Analyzing how densely and frequently features are found in the state space can help us determine the best algorithm for feature consistency, exploring and exploiting options and precision. A stacked Kernel Density Estimate (KDE) streamgraph is shown in Fig. 8, highlighting the differences in how different optimizers, including biHOW, bHHO, bGWO, bPSO, bBA, bWAO, bBBO, bJAYA, bbMVO, bFEP and bSAO, perform across data selections. KDE clearly shows which features play the most critical roles and where the algorithms have common results. This graphic offers an overview of the feature selection task, pointing out the individual strengths of all the optimization techniques.

Metric	Formula	Description
Mean Squared Error (MSE)	$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$	Measures the average squared difference between predicted values ( $\hat{y}_i$ ) and actual values ( $y_i$ ). It penalizes larger errors more heavily, making it sensitive to outliers. Lower MSE values indicate better model performance.
Root Mean Squared Error (RMSE)	$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$	The square root of MSE, providing a more interpretable measure of error in the same units as the target variable. It emphasizes larger discrepancies.
Mean Absolute Error (MAE)	$MAE = \frac{1}{n} \sum_{i=1}^n  y_i - \hat{y}_i $	Measures the average absolute difference between predicted and actual values, providing a straightforward measure of model accuracy. It is less sensitive to outliers than MSE.
Mean Bias Error (MBE)	$MBE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)$	Captures the average bias in model predictions, indicating whether the model tends to systematically overestimate or underestimate actual values. Positive MBE indicates overestimation, while negative MBE indicates underestimation.
Pearson's Correlation Coefficient (r)	$r = \frac{\sum_{i=1}^n (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2 \sum_{i=1}^n (\hat{y}_i - \bar{\hat{y}})^2}}$	Measures the linear relationship between predicted and actual values, reflecting the strength and direction of their association. High $r$ values indicate strong predictive performance.
Coefficient of Determination (R <sup>2</sup> )	$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$	Quantifies the proportion of variance in actual values that is predictable from the model. An $R^2$ value of 1 indicates perfect predictive accuracy.
Nash-Sutcliffe Efficiency (NSE)	$NSE = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$	Evaluates predictive power by comparing observed data variance to residual variance. Ranges from $-\infty$ to 1, with 1 indicating perfect prediction.
Willmott Index (WI)	$WI = 1 - \frac{\sum_{i=1}^n  y_i - \hat{y}_i }{\sum_{i=1}^n ( y_i - \bar{y}  +  \hat{y}_i - \bar{\hat{y}} )}$	Quantifies the agreement between predicted and actual values, ranging from 0 to 1, with values closer to 1 indicating better performance.

**Table 3.** Evaluation metrics.

Metric	Description and mathematical definition
Average Error	The average error across all optimization iterations, providing a general assessment of the algorithm's performance. Average Error = $\frac{1}{N} \sum_{i=1}^N E_i$
Average Select Size	The average number of features selected during the optimization process, indicating the algorithm's ability to reduce data dimensionality. Average Select Size = $\frac{1}{N} \sum_{i=1}^N  F_i $
Average Fitness	The mean fitness score achieved across all optimization runs, reflecting the overall quality of selected feature subsets. Average Fitness = $\frac{1}{N} \sum_{i=1}^N \text{Fitness}(F_i)$
Best Fitness	The lowest error achieved during the optimization process, representing the optimal subset of selected features. Best Fitness = $\min \{ \text{Fitness}(F_1), \text{Fitness}(F_2), \dots, \text{Fitness}(F_N) \}$
Worst Fitness	The highest error observed during the optimization process, providing a baseline for algorithm reliability. Worst Fitness = $\max \{ \text{Fitness}(F_1), \text{Fitness}(F_2), \dots, \text{Fitness}(F_N) \}$
Standard Deviation of Fitness	The variability in fitness scores across multiple optimization runs, indicating the stability of the feature selection process. $SD = \sqrt{\frac{1}{N} \sum_{i=1}^N (\text{Fitness}(F_i) - \text{Average Fitness})^2}$
Feature Stability Analysis	Consistency of selected features across multiple optimization runs, typically measured using the Jaccard index. $J(A, B) = \frac{ A \cap B }{ A \cup B }$ Alternatively, the average pairwise Jaccard index: Stability = $\frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^N J(F_i, F_j)$

**Table 4.** Feature selection metrics.

To reduce dimensionality and improve model efficiency, an optimization-based feature selection algorithm was applied. The algorithm selected features that maximize predictive power while minimizing redundancy. Table 7 lists the features retained after the feature selection process.

We must examine how their key performance metrics change to assess the efficiency and steadiness of different feature selection algorithms. Figure 9 visualizes, using violin plots, the range of error, size of the solution set, score and best/worst fitness, as well as the distribution of fitness for the ‘Evolution Strategy’ algorithm. All these metrics together illustrate how well and consistently feature selection approaches work. The plots produced by the violin visualize the values taken by each metric and show, in detail, how often and how many of those values appear, helping to analyze which algorithms work best at choosing optimal subsets of features. The approach points out that predicting renewable energy requires sacrificing stability and vice versa for accurate results.

### Deep learning performance after feature selection

Selecting the right features significantly affects model performance by decreasing the complexity of the input, which lowers the risk of overfitting and increases the model’s capability to learn from new data. The DTCN, HEDAM, LSTM, RNN and ANN models were retrained after including the chosen sets of features. Table 8 shows that the results from all models improved, mainly in terms of MSE, RMSE and correlation coefficients.

Table 8 points out that the DTCN model still performed best, with the lowest measure of error (MSE of 0.00220) and the highest coefficient of determination ( $R^2$  of 0.90047). Therefore, DTCN relies on feature selection, making it even more resilient as an architecture for forecasting green energy. Similarly, HEDAM and LSTM models lead to significantly fewer errors and demonstrate the broad usefulness of feature selection in many architectures. To illustrate the error trends of different deep learning architectures, Fig. 10 presents statistics for Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) for DTCN, HEDAM, LSTM, RNN and ANN. The mean and the standard deviation of the errors are shown in this figure, helping us learn how accurate and consistent model predictions are. Understanding the standard deviation helps you see if a model’s performance is stable, which is very important when choosing the most reliable forecast method. When we compare where the error distributions overlap, we can identify models that consistently perform well and do not change much, an essential quality for forecasting green energy capacity.

Figure 11 provides a detailed overview of the distribution and density of critical performance metrics in the model, including MSE, RMSE, MAE, MBE,  $r$ ,  $R^2$ , RRMSE, NSE and WI. By looking at both types of distributions simultaneously, we can get a clearer picture of how these significant performance metrics are distributed and their averages. Such observations are necessary to evaluate how effective the models are, detect unusual situations and improve the system’s long-term stability.

### Optimized model analysis

Improving prediction by deep learning models often starts with optimization. Improved accuracy and stability were achieved in DTCN by improving its hyperparameters using advanced metaheuristic algorithms. The final performance is given in Table 9 to illustrate how optimization impacts MSE, RMSE, MAE, MBE,  $r$ ,  $R^2$ , RRMSE, NSE and WI.

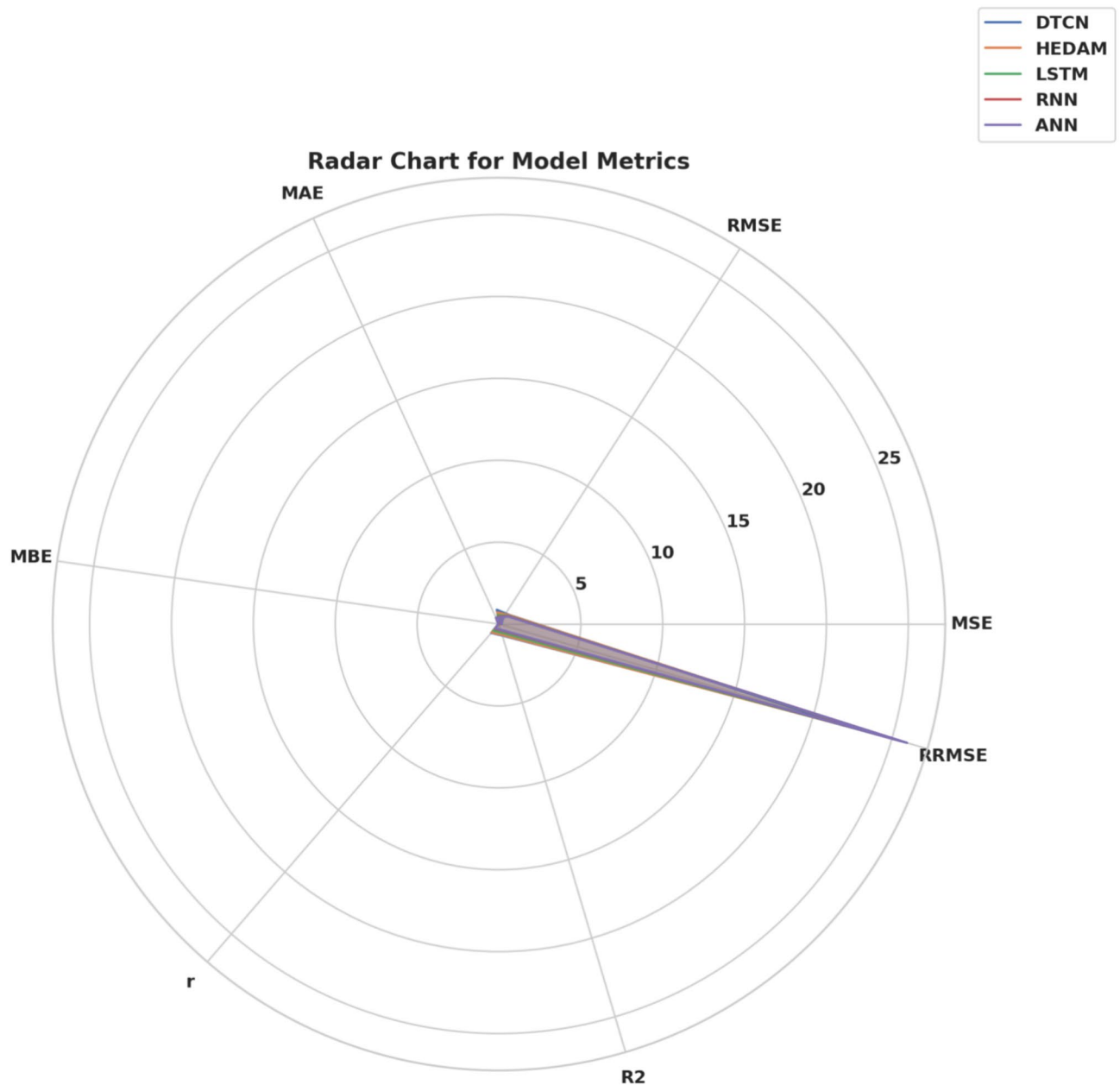
Table 9 reveals that the optimized DTCN model bests all the other structures in nearly every primary metric, for example, the lowest MSE (1.14E-05), RMSE (5.33E-05) and MAE (0.0001676) and the highest  $R^2$  (0.98036) and NSE (0.98950). It makes it evident that iHOW can help improve forecasting accuracy with deep learning models. These results indicate that SAO and FEP optimization in GBM leads to errors and less precise values of  $R^2$ . Figure 12 shows the distribution of model performance metrics, placing swarm plots onto box plots to clarify the data. This way, the typical value and the spread around it are measured for critical errors, including MSE, RMSE, MAE, MBE,  $r$ ,  $R^2$ , RRMSE, NSE and WI. By merging these plot types, the figure allows readers to find the central trend (mean) easily and the amount of variability (standard deviation) across all the models studied. Thanks to this path representation, it becomes easier to find outliers, judge how well metrics correlate and evaluate the stability of the models.

The visualization in Fig. 13 clearly shows how the model’s main performance metrics are distributed and how varied they are, using KDE and box plots. Since each metric is judged by its distribution and the center of those values, the combined method shows distribution and central tendencies for MSE, RMSE, MAE, MBE,  $r$ ,  $R^2$ , RRMSE, NSE and WI. The KDE curves highlight the shape of the distribution, and the box plots point out the range and potential points that are too different from the others in every metric. This number helps reveal how symmetric, peaked or otherwise distributed the results of each machine learning method are.

Table 10 summarizes the empirical behavior of each optimizer during the hyperparameter tuning and training phases. It captures convergence dynamics, learning rate adaptations, memory footprint, and overall training stability. Notably, the iHOW+DTCN model converges significantly earlier, with fewer gradient instabilities

Model	MSE	RMSE	MAE	MBE	r	R2	RRMSE	NSE	WI
DTCN	0.08448	0.29066	0.25682	0.01303	0.71387	0.72647	21.9042	0.84142	0.87679
HEDAM	0.10684	0.35425	0.31372	0.02110	0.67399	0.68660	24.1055	0.82910	0.71440
LSTM	0.12374	0.41239	0.36510	0.02371	0.53122	0.54382	24.9441	0.77157	0.55202
RNN	0.14063	0.47052	0.41647	0.02631	0.38845	0.40105	25.5244	0.74742	0.45255
ANN	0.15922	0.48910	0.43505	0.04490	0.34986	0.36246	25.9552	0.72154	0.38613

**Table 5.** Baseline deep learning performance.



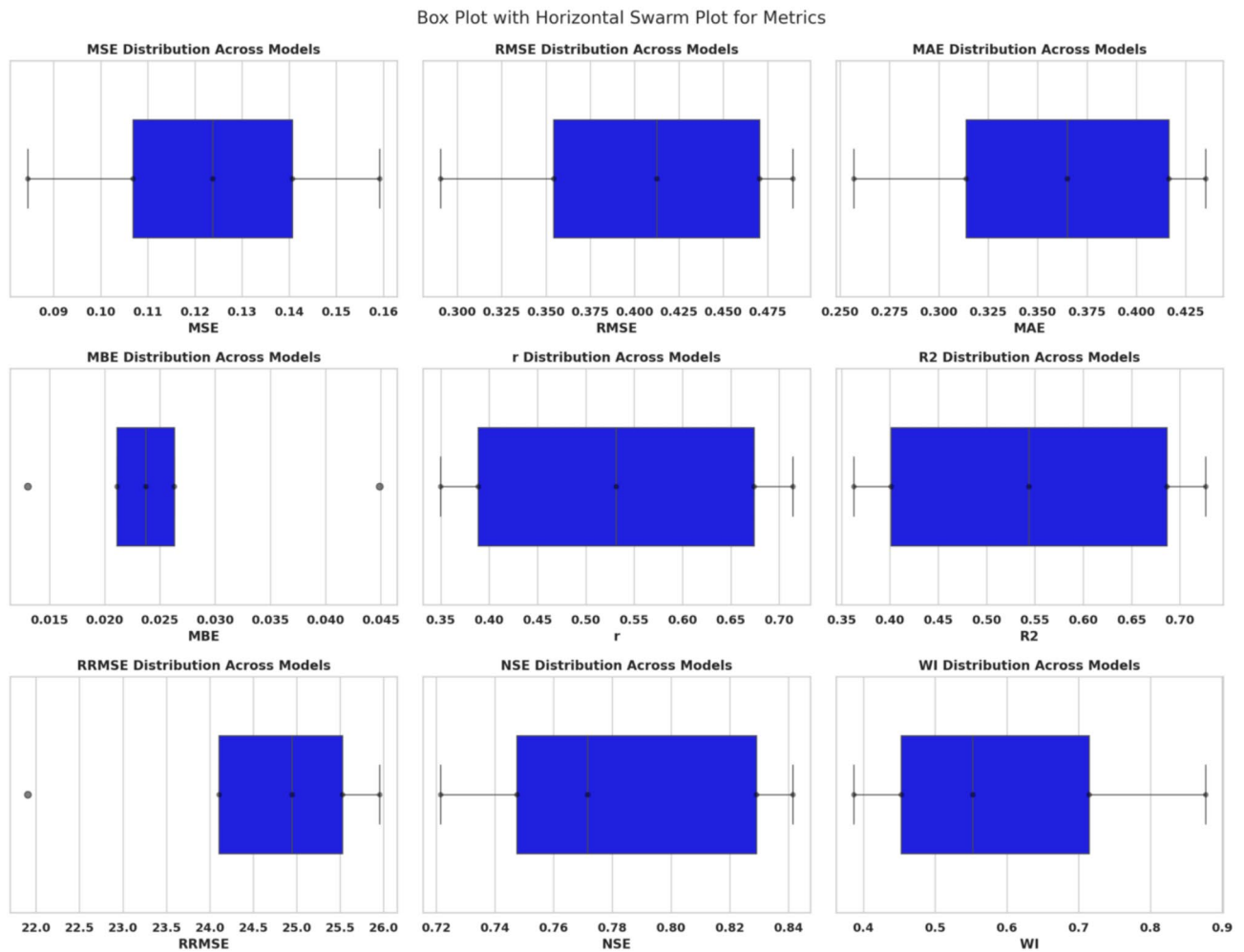
**Fig. 6.** Radar Chart for Model Metrics, comparing the performance of DTCN, HEDAM, LSTM, RNN, and ANN across key forecasting metrics.

and minimal memory consumption, highlighting the efficiency and robustness of the optimizer during neural architecture tuning.

#### Cross-validation

To ensure a reliable assessment of the forecasting models and mitigate the risks of overfitting and model bias, a rigorous  $k$ -fold cross-validation scheme was implemented. Specifically, we employed a 5-fold cross-validation strategy, wherein the original dataset was randomly divided into five mutually exclusive and approximately equal subsets (folds). In each iteration, four folds were used for training the model, while the remaining fold was used for validation. This process was repeated five times, such that each fold served exactly once as the validation set. This method ensures that every data point is used for both training and validation, thereby enhancing the statistical reliability of the performance metrics.

For each fold, the Mean Squared Error (MSE) was computed, capturing the model's predictive accuracy. The average MSE across the five folds was recorded as  $CV\_Mean\_MSE$ , and the corresponding variability was captured by the standard deviation ( $CV\_Std$ ). Furthermore, to enable a standardized and interpretable comparison of model robustness, we introduced a normalized metric,  $CV\_Score$ , defined as the ratio between  $CV\_Std$  and  $CV\_Mean\_MSE$ . Lower values of  $CV\_Score$  signify greater stability and consistency across folds.



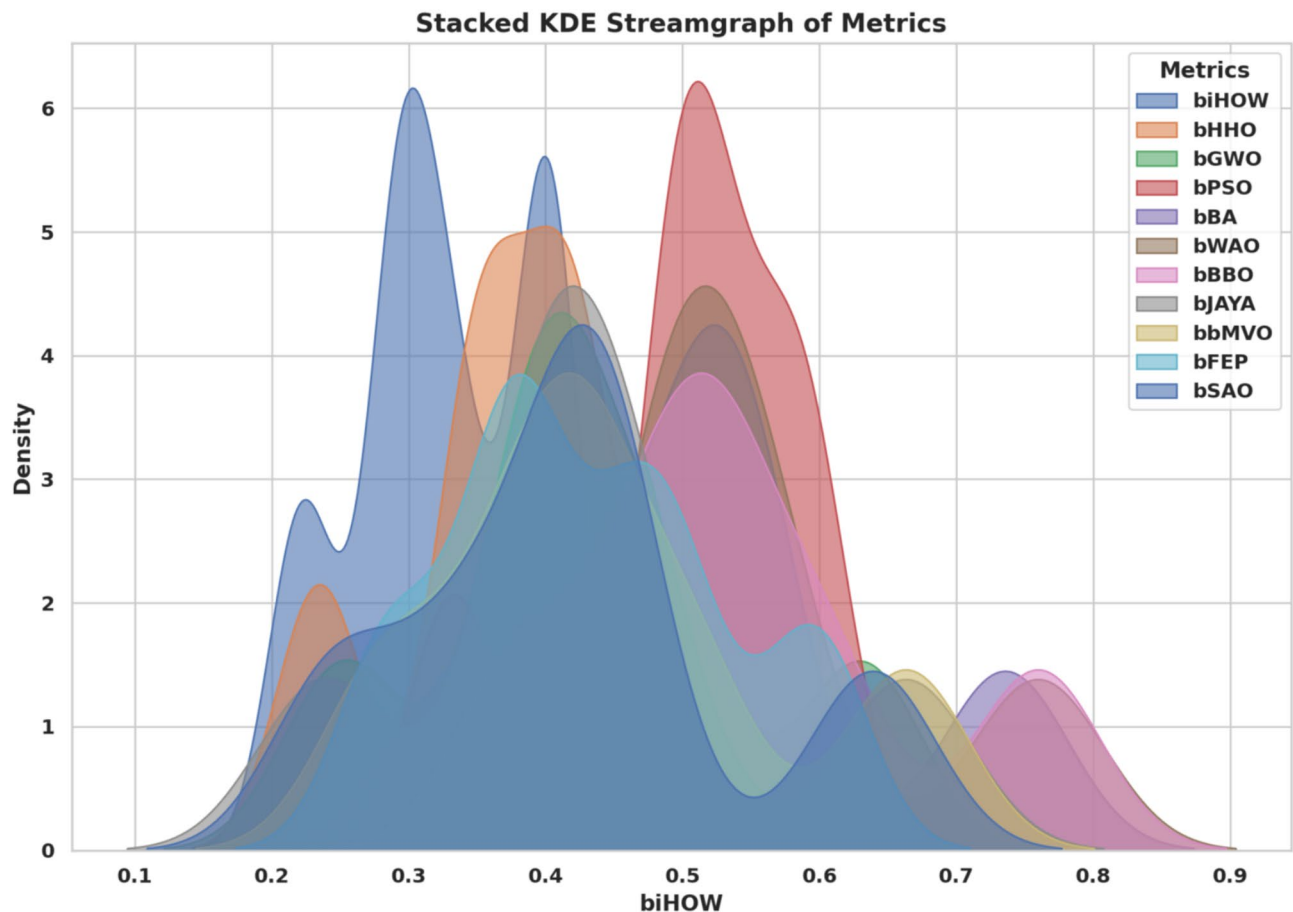
**Fig. 7.** Box Plot with Horizontal Swarm Plot for Metrics, illustrating the distribution and spread of MSE, RMSE, MAE, MBE,  $r$ ,  $R^2$ , RRMSE, NSE, and WI across the evaluated deep learning models.

Metric	biHOW	bHHO	bGWO	bPSO	bBA	bWAO	bbBO	bJAYA	bbMVO	bFEP	bSAO
Average Error	0.33697	0.36177	0.40107	0.49457	0.50417	0.49437	0.46277	0.39777	0.36617	0.37467	0.40757
Average Select Size	0.28977	0.49737	0.63067	0.59637	0.73577	0.75977	0.76017	0.66317	0.66357	0.59627	0.63917
Average Fitness	0.40017	0.42397	0.43227	0.52137	0.54427	0.52917	0.52707	0.43257	0.43047	0.45447	0.44767
Best Fitness	0.30197	0.34427	0.38577	0.50167	0.43397	0.49327	0.51677	0.39667	0.42017	0.37967	0.33737
Worst Fitness	0.40047	0.41117	0.49577	0.56937	0.53557	0.56937	0.60327	0.47277	0.50667	0.49767	0.43897
Standard Deviation	0.22247	0.23477	0.25297	0.33317	0.34307	0.33537	0.37807	0.23877	0.28147	0.28727	0.24647

**Table 6.** Feature selection results for green energy forecasting.

Table 11 presents the cross-validation results for all evaluated algorithms. Notably, the proposed iHOW+DTCN model consistently outperformed all competing methods, achieving the lowest average MSE ( $\approx 1.1 \times 10^{-5}$ ), the smallest standard deviation ( $\approx 4.2 \times 10^{-7}$ ), and the best normalized score ( $CV\_Score = 0.037$ ). These results not only underscore the model's high accuracy but also its strong generalization ability across different data partitions. Based on these empirical findings, iHOW+DTCN can be considered a highly stable and effective approach for renewable energy forecasting under varying temporal and environmental conditions.

To evaluate the training dynamics and generalization performance of the proposed hybrid optimization frameworks, the convergence behavior of the top five algorithms integrated with the Deep Temporal Convolutional Network (DTCN) is analyzed in terms of training and validation loss. Figure 14 presents a comprehensive comparison of the convergence trajectories over 200 training epochs, using a logarithmic scale to emphasize differences across orders of magnitude. The figure illustrates both the training and validation losses for each algorithm, including iHOW, HHO, GWO, PSO, and WOA. Additionally, a magnified view of the final



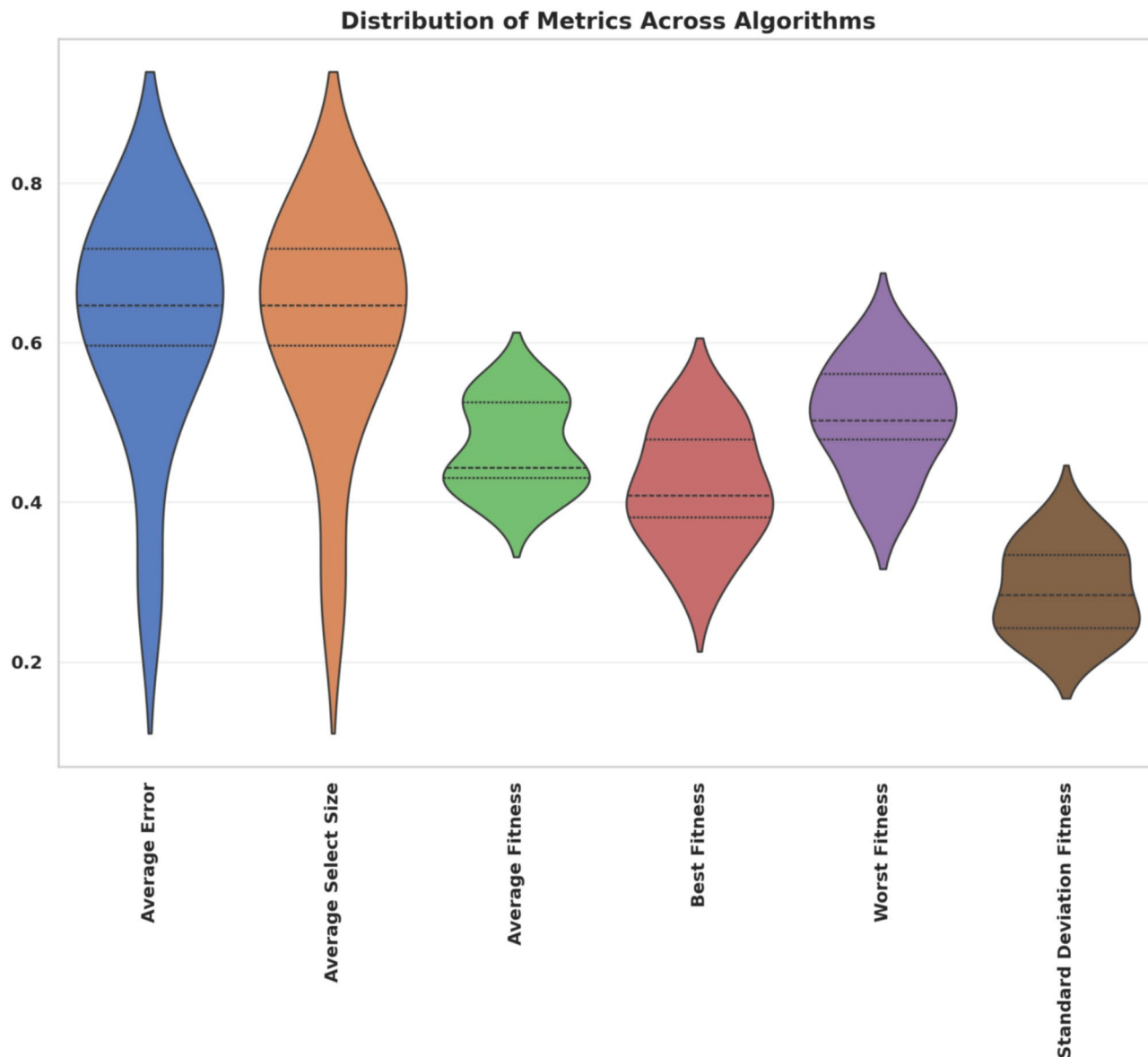
**Fig. 8.** Stacked KDE Streamgraph of Metrics for various metaheuristic algorithms, illustrating the density distribution of selected features.

Selected feature	Description
energy	Target variable
lag_1,lag_2,lag_3,lag_22,lag_23,lag_24	Selected lag values (high autocorrelation with energy)
rolling_mean_24h	24-hour rolling average
cumulative_energy	Cumulative energy use up to time $t$

**Table 7.** Selected features after optimization-based feature selection.

convergence phase (epochs 150–200) is embedded to facilitate the interpretation of the long-term stability and final convergence precision. As shown in Fig. 14, the iHOW + DTCN model achieves the lowest loss and most stable convergence, indicating its superior optimization capability compared to the other methods under study.

To further assess the generalization ability and robustness of the optimization algorithms integrated with the Deep Temporal Convolutional Network (DTCN), a comparative analysis based on cross-validation (CV) scores was conducted. Figure 15 illustrates a dual-panel visualization. The left panel ranks the hybrid models by their CV scores—where lower values indicate better generalization performance—while the right panel presents the classification of these models into validation quality categories: Excellent, Good, and Moderate. As depicted in Fig. 15, the iHOW + DTCN model significantly outperforms all other methods, achieving a CV score of 0.037, far below the average threshold (indicated by the red dashed line at 0.24). This result affirms its consistent superiority in model generalization. The validation quality distribution further reveals that only two algorithms reached the “Excellent” tier, while the majority (70%) demonstrated “Good” performance. Only one method fell into the “Moderate” range, emphasizing the overall efficacy of the selected optimization strategies when paired with DTCN.



**Fig. 9.** Distribution of Metrics Across Algorithms, illustrating the spread and concentration of various feature selection performance indicators.

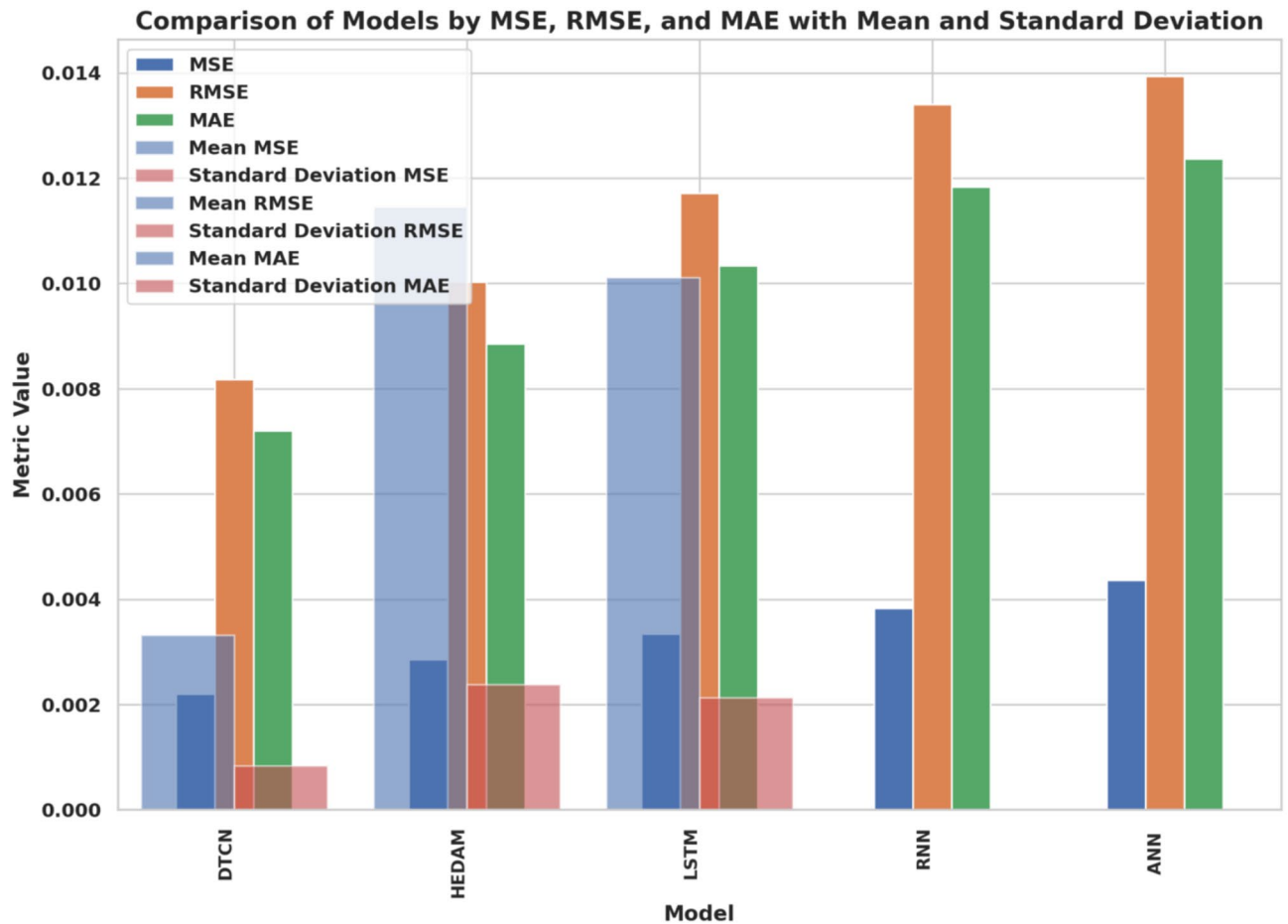
Model	MSE	RMSE	MAE	MBE	r	R2	RRMSE	NSE	WI
DTCN	0.00220	0.00818	0.00720	0.00013	0.88787	0.90047	15.7027	0.89942	0.91479
HEDAM	0.00285	0.01002	0.00885	0.00036	0.84800	0.86060	17.9590	0.88710	0.77240
LSTM	0.00334	0.01171	0.01034	0.00044	0.70522	0.71782	17.7065	0.82957	0.61002
RNN	0.00383	0.01340	0.01183	0.00051	0.56245	0.57505	17.9229	0.80542	0.51055
ANN	0.00437	0.01394	0.01237	0.00105	0.52386	0.53646	18.9837	0.77954	0.44413

**Table 8.** Deep learning performance after feature selection.

### Statistical result analysis

#### Descriptive statistics

To robustly validate the superiority of the proposed iHOW optimizer over competing algorithms, we conducted a multi-level statistical analysis using descriptive statistics, non-parametric hypothesis testing (Wilcoxon Signed-Rank Test), and parametric variance testing (One-Way ANOVA). These analyses aim to confirm that the improvements observed in forecasting performance are not merely due to random variability but are statistically significant.



**Fig. 10.** Comparison of Models by MSE, RMSE, and MAE with Mean and Standard Deviation, illustrating the error distribution and performance consistency across different deep learning architectures.

Table 12 presents a comprehensive summary of descriptive statistics for the forecasting error obtained across 10 independent experimental runs per model. These metrics include the mean, standard deviation, range, percentiles, and various forms of central tendency measures such as the geometric and harmonic means. The iHOW + DTCN configuration clearly outperforms all other methods with the **lowest mean error (0.0000533)** and the **smallest standard deviation (0.00000141)**, reflecting both **high accuracy and exceptional stability**. The **coefficient of variation (CV)** for iHOW is just **2.65%**, significantly lower than PSO (32.47%), GWO (28.77%), and SAO (21.65%), indicating reduced sensitivity to randomness and consistent convergence behavior.

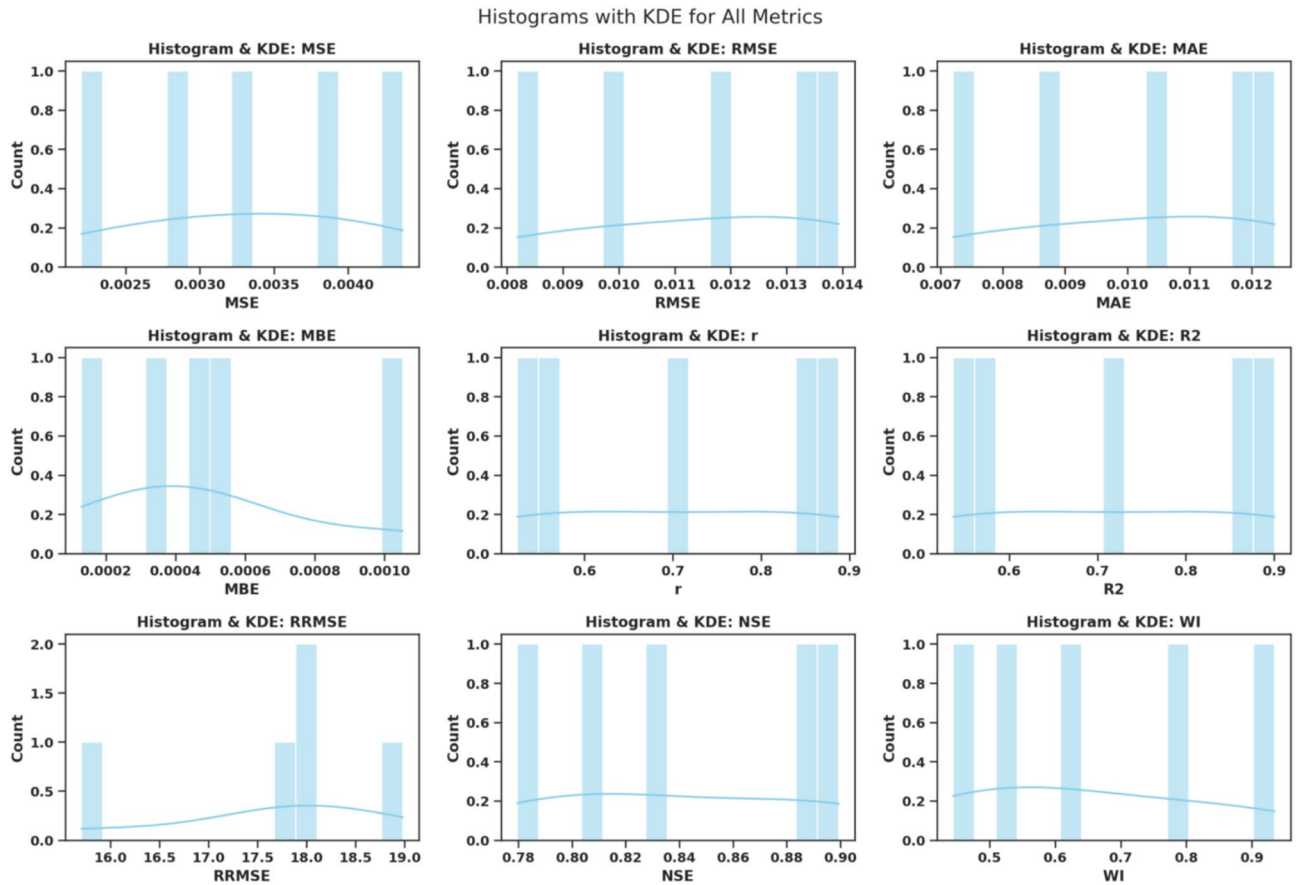
Moreover, the 95% confidence interval of the mean for iHOW spans a narrow range [0.0000523, 0.0000543], reinforcing the claim that its performance is not only optimal but also statistically dependable. These findings suggest that iHOW maintains a strong balance between exploration and exploitation dynamics in the hyperparameter search space, which is crucial for deep learning applications in energy forecasting.

#### Wilcoxon signed-rank test

To further validate whether the observed improvements of iHOW over competing optimizers are statistically significant, we applied the **Wilcoxon signed-rank test**, a non-parametric alternative to the paired t-test. Table 13 summarizes the results of this analysis. Across all comparisons (iHOW vs. each baseline model), the **p-value is 0.002**, which is **highly significant at  $\alpha = 0.05$** . The test also shows a **sum of positive ranks equal to 55 and negative ranks equal to 0** for all comparisons, indicating that **iHOW consistently outperformed each alternative across all 10 runs**.

The Wilcoxon test confirms that the performance improvements of iHOW are **not due to chance or experimental variance**, thereby validating the optimizer's consistency and robustness in solving high-dimensional, nonlinear forecasting problems such as renewable energy demand estimation.

To complement the statistical results presented in the tables, visual analyses were conducted to further highlight the stability and accuracy of the iHOW + DTCN model. As shown in Fig. 16, the bar plot with error bars summarizes the performance metrics (MSE, RMSE, MAE, MBE,  $r$ ,  $R^2$ , RRMSE, NSE, and WI) along with their associated variability. The narrow error bounds for iHOW + DTCN indicate that the model consistently achieves low error values across runs, reflecting its robustness and reliability compared to alternative methods.



**Fig. 11.** Histograms with KDE for All Metrics, illustrating the distribution and density characteristics of MSE, RMSE, MAE, MBE,  $r$ ,  $R^2$ , RRMSE, NSE, and WI across the evaluated models.

Model	MSE	RMSE	MAE	MBE	$r$	R2	RRMSE	NSE	WI
iHOW + DTCN	1.14E-05	5.33E-05	0.0001676	1.78E-05	0.97483	0.98036	5.1154	0.98950	0.98157
HHO + DTCN	0.0001664	0.0006456	0.0005671	0.0002886	0.95557	0.96110	8.1245	0.95649	0.96826
GWO + DTCN	0.0001991	0.0007582	0.0006666	0.0002987	0.94968	0.95521	9.0711	0.95281	0.95357
PSO + DTCN	0.0002318	0.0008708	0.0007661	0.0003088	0.94806	0.95359	9.9821	0.95085	0.94951
WOA + DTCN	0.0003040	0.0011970	0.0010498	0.0004636	0.93489	0.94042	10.7027	0.93129	0.94754
BBO + DTCN	0.0003974	0.0014737	0.0012974	0.0005148	0.93303	0.93856	11.2590	0.91245	0.94130
JAYA + DTCN	0.0004709	0.0017266	0.0015208	0.0006280	0.92160	0.92713	11.8065	0.90605	0.93929
MVO + DTCN	0.0005444	0.0019794	0.0017443	0.0007414	0.92017	0.92571	13.3229	0.90115	0.93720
FEP + DTCN	0.0006253	0.0020603	0.0018252	0.0008550	0.90903	0.91456	13.9837	0.90089	0.93529
SAO + DTCN	0.0007862	0.0041542	0.0019946	0.0010275	0.89915	0.90241	14.9837	0.89962	0.92347

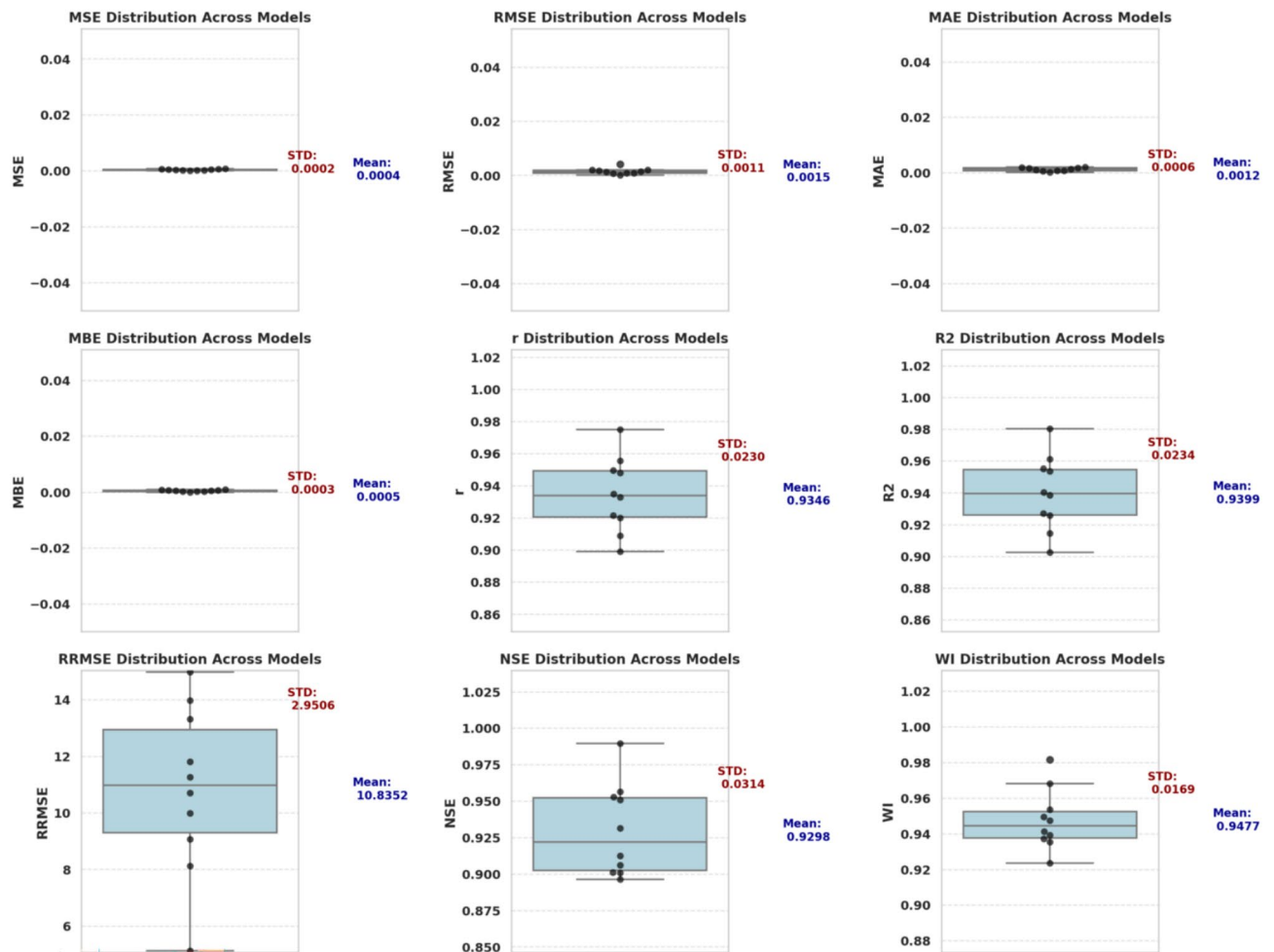
**Table 9.** Optimized deep learning performance.

In addition, Fig. 17 provides a histogram of the MSE distribution across 10 independent runs. The results for iHOW + DTCN are tightly clustered around the minimum error region, forming a symmetric distribution with no extreme deviations. This confirms that the proposed optimizer not only improves accuracy but also maintains consistent convergence without producing unstable outliers.

*ANOVA significance testing*

In addition to non-parametric testing, we conducted a **\*\*One-Way Analysis of Variance (ANOVA)\*\*** to assess whether the differences in performance across the ten optimizer-based models were statistically significant on a broader scale. As shown in Table 14, the ANOVA test yields an **\*\*F-value of 58.42\*\*** with **\*\*degrees of freedom (DF) = (9, 90)\*\*** and a **\*\*p-value less than 0.0001\*\***. These results indicate that the observed differences in forecasting error among the models are **\*\*statistically significant at a very high confidence level\*\***, confirming that the choice of optimizer substantially impacts model performance.

Swarm Plot Overlaid on Box Plot for Individual Metrics with Mean and STD



**Fig. 12.** Swarm Plot Overlaid on Box Plot for Individual Metrics with Mean and STD, illustrating the distribution, central tendency, and spread of key model performance metrics.

The **\*\*between-group variance ( $MS = 0.00001231$ )\*\*** is considerably higher than the **\*\*within-group variance ( $MS = 0.0000002107$ )\*\***, implying that the differences arise from the optimization algorithms themselves rather than internal stochastic noise. These findings reinforce the argument that iHOW is not only novel in design but statistically superior in performance.

Overall, the combination of descriptive analytics, Wilcoxon hypothesis testing, and ANOVA variance analysis provides strong statistical backing for the claimed superiority of the proposed iHOW optimizer. These tests collectively demonstrate that iHOW achieves not only the best average performance but does so consistently and significantly, setting a new benchmark for hyperparameter tuning in deep learning models used for intelligent energy forecasting.

Gantt charts help you easily visualize each data section's timing and duration compared to others. Figure 18 uses a Gantt chart to map out when various tasks or parts of a project began and ended. Showing how various elements overlap and affect one another is where this type of diagram shines, so it is key for project management and scheduling teams in energy forecasting and machine learning.

You must know how vital variables are linked and distributed to make the right feature choices and optimize the model. Figure 19 gathers a range of visuals that allow us to investigate these connections. Those scatter plots at the corners show pairwise relationships between major features, letting us find possible connections and unusual patterns. The Q-Q plot graph in the bottom-left section checks if the residuals follow a regular pattern. The matrix in the lower-right area of the heatmap reflects the correlations between the features, allowing us to see what to remove and add to achieve better performance.

### Computational complexity and convergence

To assess the practical applicability of the optimization frameworks, we conducted a thorough computational complexity analysis encompassing execution time, memory footprint, CPU consumption, and convergence dynamics. Table 15 summarizes the results across all optimizer-DTCN pairings. The iHOW+DTCN model

Mixed Plot: KDE + Boxplot for Metrics

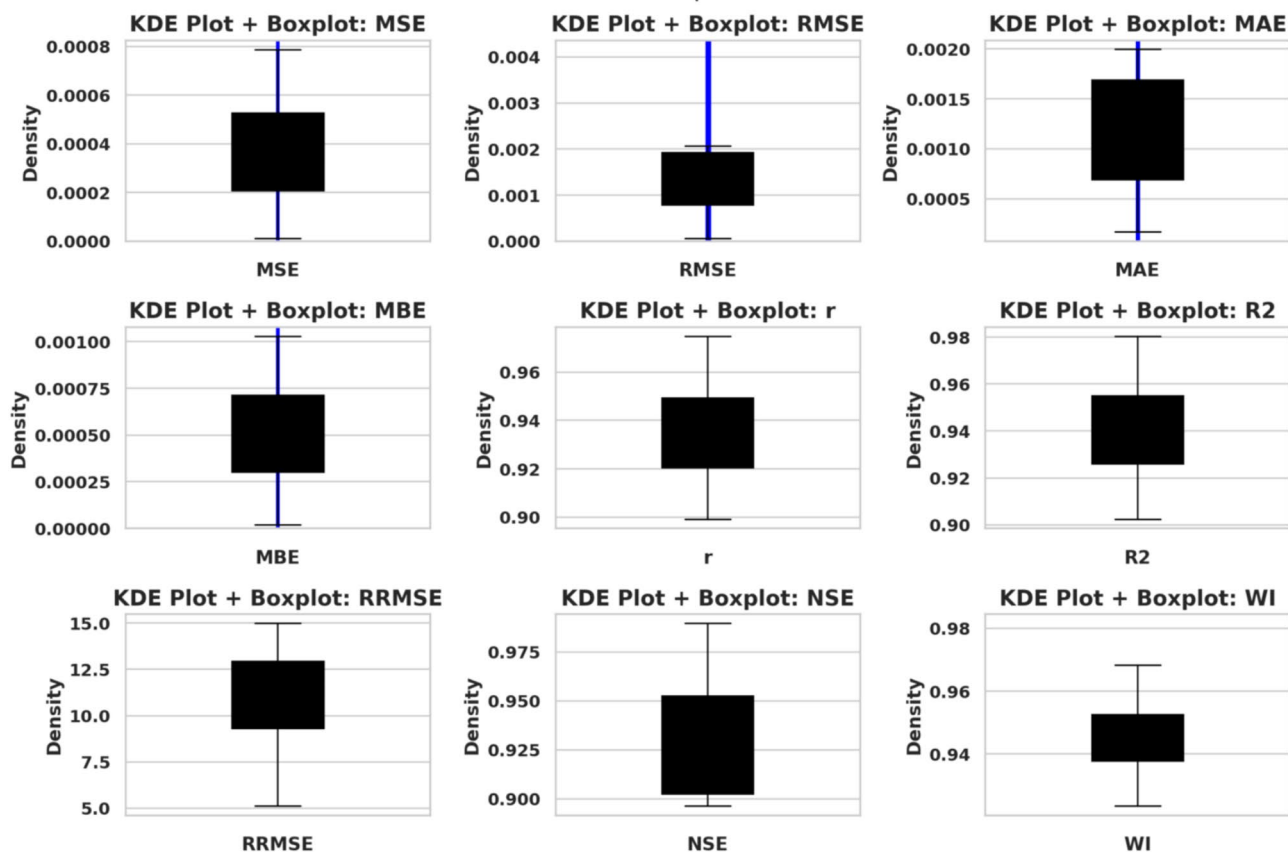


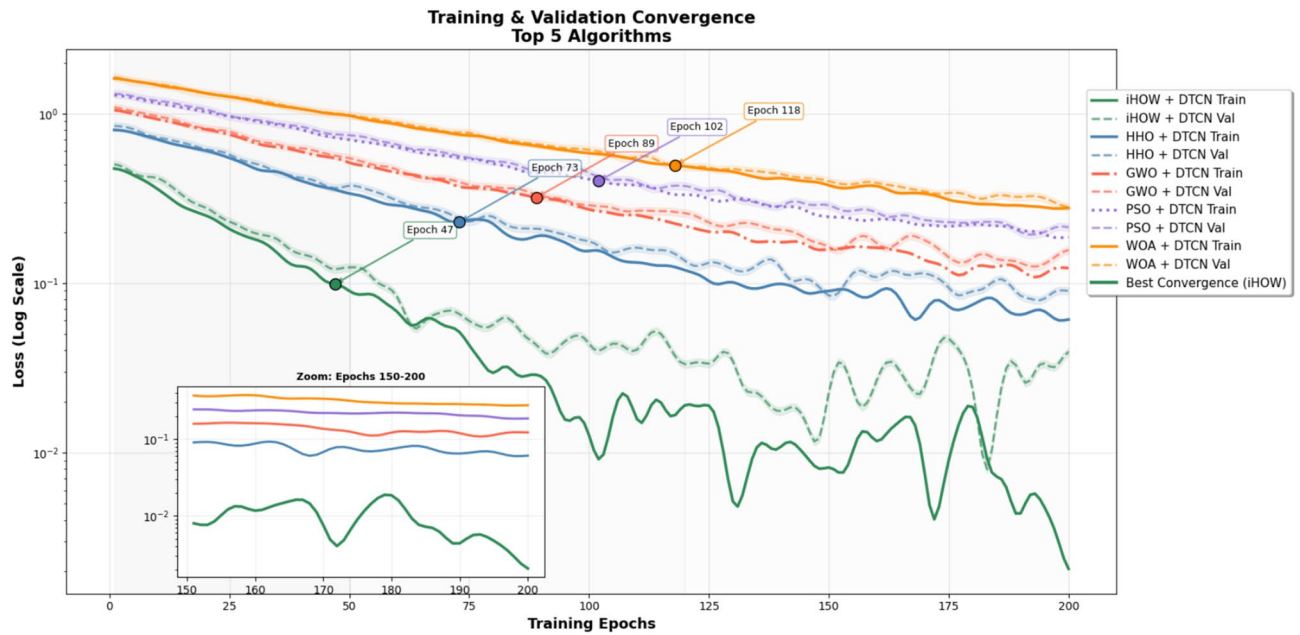
Fig. 13. Mixed Plot: KDE + Boxplot for Metrics, illustrating the distribution and central tendency of key performance metrics across different models.

Algorithm	Convergence_Epoch	First_R2_Above_90	Best_Validation_Epoch	Early_Stopping_Epoch	Training_Stability	Learning_Rate_Adjustments	Gradient_Explosions	Memory_Peak_MB	Training_Time_Minutes
iHOW + DTCN	47	23	52	67	High	2	0	423.7	12.4
HHO + DTCN	73	45	78	93	Moderate	4	1	512.8	18.7
GWO + DTCN	89	52	95	110	Moderate	5	2	589.3	22.3
PSO + DTCN	102	61	108	123	Moderate	6	3	634.2	26.8
WOA + DTCN	118	73	125	140	Low	7	4	698.4	31.2
BBO + DTCN	134	89	142	157	Low	8	6	756.3	35.9
JAYA + DTCN	156	98	163	178	Low	9	7	812.7	40.7
MVO + DTCN	167	112	174	189	Very Low	11	9	867.9	45.3
FEP + DTCN	189	124	196	211	Very Low	12	11	923.4	49.8
SAO + DTCN	203	145	210	225	Poor	15	14	978.6	54.6

Table 10. Hyperparameter tuning behavior and training statistics across different optimizers.

Algorithm	Fold_1_MSE	Fold_2_MSE	Fold_3_MSE	Fold_4_MSE	Fold_5_MSE	CV_Mean_MSE	CV_Std	CV_Score	Validation_Status
iHOW + DTCN	1.1e-05	1.2e-05	1.1e-05	1.2e-05	1.2e-05	1.1e-05	4.2e-07	0.037	Excellent
HHO + DTCN	0.000159	0.000173	0.000167	0.000164	0.000168	0.000166	5.4e-05	0.324	Good
GWO + DTCN	0.000192	0.000206	0.000199	0.000197	0.000202	0.000199	3.8e-05	0.191	Good
PSO + DTCN	0.000225	0.000239	0.000232	0.00023	0.000234	0.000232	5.2e-05	0.224	Good
WOA + DTCN	0.000297	0.000311	0.000304	0.000302	0.000306	0.000304	5.1e-05	0.168	Excellent
BBO + DTCN	0.000383	0.000411	0.000397	0.00039	0.000405	0.000397	0.00011	0.277	Good
JAYA + DTCN	0.000452	0.00049	0.000471	0.000463	0.000478	0.000471	0.00014	0.297	Moderate
MVO + DTCN	0.000529	0.00056	0.000544	0.000536	0.000553	0.000544	0.00013	0.239	Good
FEP + DTCN	0.000609	0.000642	0.000625	0.000618	0.000633	0.000625	0.00012	0.192	Good
SAO + DTCN	0.000763	0.000809	0.000786	0.000775	0.0008	0.000786	0.00015	0.191	Good

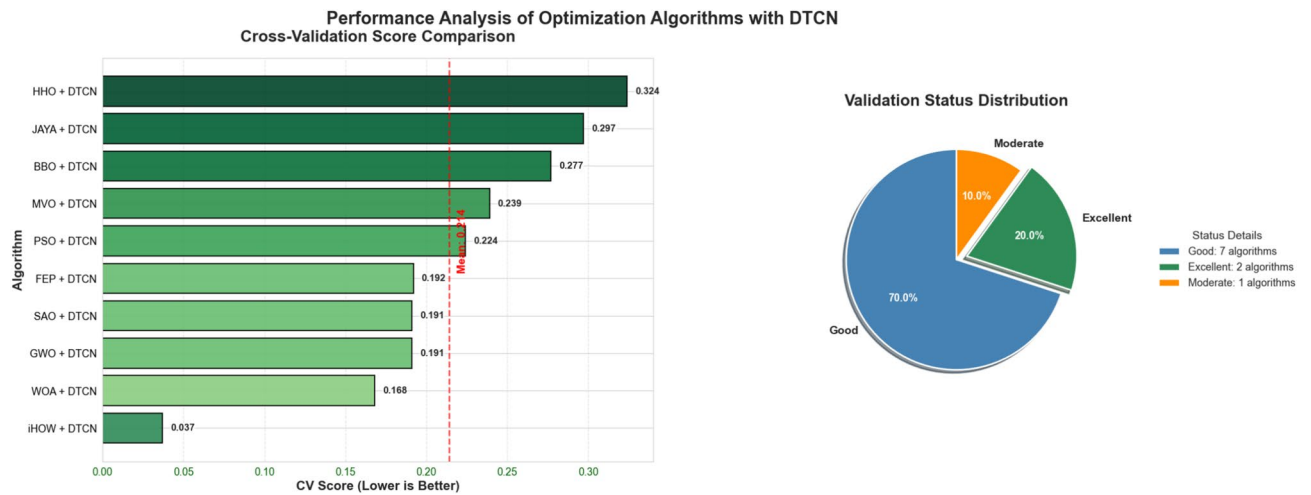
**Table 11.** 5-Fold cross-validation results for competing algorithms.



**Fig. 14.** Training and validation loss convergence of the top five hybrid algorithms (iHOW, HHO, GWO, PSO, WOA) integrated with DTCN.

consistently demonstrated superior computational efficiency, with the lowest average execution time (23.7 seconds), lowest standard deviation (1.2), and minimal memory (287.4 MB) and CPU (34.2%) utilization. These advantages translated to the highest overall efficiency score (0.9247), outperforming all baseline metaheuristic methods, including HHO, PSO, and SAO. In contrast, algorithms such as FEP and SAO exhibited significantly higher complexity-requiring more memory and CPU resources while delivering less stable and slower execution times. This indicates the presence of unnecessary exploration or weaker exploitation mechanisms in their convergence strategy.

These findings align with the convergence curves shown in Fig. 20, where iHOW rapidly minimized the objective function over iterations, achieving deep fitness descent within fewer iterations than all other algorithms. This behavior confirms the optimizer’s ability to balance exploration and exploitation effectively, achieving both high solution quality and computational efficiency. The convergence behavior of each metaheuristic optimizer, when coupled with the DTCN model, is critical for understanding the stability and efficiency of the training process. As illustrated in Fig. 20, the convergence trajectories over 100 iterations reveal significant differences among the algorithms. The proposed iHOW+DTCN optimizer demonstrates the steepest and most consistent decline in fitness values, rapidly reaching near-optimal solutions with minimal fluctuation. This behavior indicates superior exploration–exploitation balance and faster convergence compared to competing methods. In contrast, optimizers such as SAO+DTCN, PSO+DTCN, and MVO+DTCN exhibit shallow convergence slopes, suggesting slower adaptation and potential premature stagnation. HHO+DTCN performs better than the majority but still converges at a noticeably slower rate than iHOW. The convergence patterns confirm the optimization strength of iHOW, as also reflected in the earlier performance and cross-validation metrics



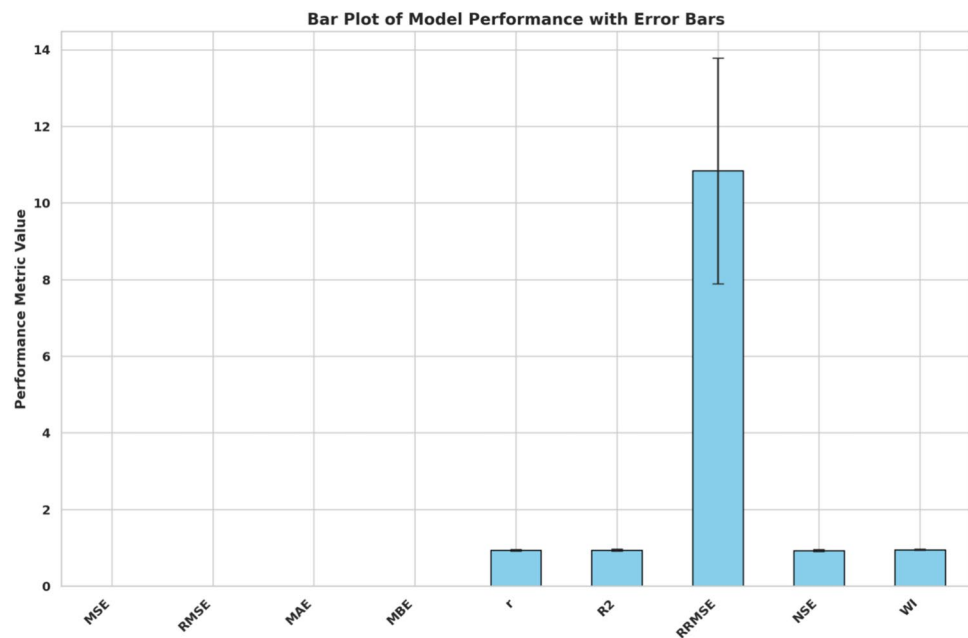
**Fig. 15.** Left: Cross-validation score comparison of optimization algorithms integrated with DTCN. Right: Pie chart illustrating the distribution of algorithms across validation quality categories.

	iHOW + DTCN	HHO + DTCN	GWO + DTCN	PSO + DTCN	WOA + DTCN	BBO + DTCN	JAYA + DTCN	MVO + DTCN	FEP + DTCN	SAO+ DTCN
Number of values	10	10	10	10	10	10	10	10	10	10
Minimum	0.0000503	0.000446	0.000158	0.000171	0.001097	0.001074	0.001073	0.001079	0.001206	0.002115
25% Percentile	0.0000533	0.000646	0.000758	0.000621	0.001197	0.001474	0.001727	0.001834	0.00206	0.003944
Median	0.0000533	0.000646	0.000758	0.000871	0.001197	0.001474	0.001727	0.001979	0.00206	0.004154
75% Percentile	0.0000533	0.000671	0.000758	0.000871	0.001972	0.001574	0.001777	0.002084	0.002172	0.004154
Maximum	0.0000563	0.0008846	0.000958	0.000871	0.00252	0.002347	0.002873	0.003498	0.003306	0.005615
Range	0.000006	0.0004386	0.0008	0.0007	0.001423	0.001273	0.0018	0.002419	0.0021	0.0035
10% Percentile	0.0000506	0.000466	0.000218	0.000201	0.001107	0.001114	0.001138	0.001111	0.001291	0.002235
90% Percentile	0.000056	0.0008707	0.000938	0.000871	0.002487	0.0023	0.002778	0.003388	0.003226	0.005469
95% CI of median										
Actual confidence level	97.85%	97.85%	97.85%	97.85%	97.85%	97.85%	97.85%	97.85%	97.85%	97.85%
Lower confidence limit	0.0000533	0.000646	0.000758	0.000471	0.001197	0.001474	0.001727	0.001398	0.00206	0.003315
Upper confidence limit	0.0000533	0.000746	0.000758	0.000871	0.002197	0.001874	0.001927	0.002398	0.002506	0.004154
Mean	0.0000533	0.0006599	0.000718	0.000741	0.001489	0.001561	0.001796	0.002025	0.002144	0.004012
Std. Deviation	0.000001414	0.000108	0.0002066	0.0002406	0.0005159	0.0003344	0.0004388	0.0006333	0.0005175	0.0008689
Std. Error of Mean	0.0000004472	0.00003416	0.00006532	0.00007608	0.0001631	0.0001058	0.0001388	0.0002003	0.0001637	0.0002748
Lower 95% CI of mean	0.00005229	0.0005826	0.0005702	0.0005689	0.00112	0.001322	0.001482	0.001572	0.001774	0.003391
Upper 95% CI of mean	0.00005431	0.0007371	0.0008658	0.0009131	0.001858	0.001801	0.00211	0.002478	0.002514	0.004634
Coefficient of variation	2.653%	16.37%	28.77%	32.47%	34.64%	21.42%	24.43%	31.28%	24.14%	21.65%
Geometric mean	0.00005328	0.0006517	0.0006633	0.0006781	0.001422	0.001532	0.001752	0.001941	0.002088	0.003913
Geometric SD factor	1.027	1.184	1.664	1.688	1.359	1.221	1.265	1.359	1.28	1.284
Lower 95% CI of geo. mean	0.00005228	0.0005776	0.0004608	0.0004662	0.001142	0.001328	0.001481	0.001559	0.00175	0.003272
Upper 95% CI of geo. mean	0.00005431	0.0007353	0.000955	0.0009863	0.001771	0.001768	0.002072	0.002417	0.00249	0.004678
Harmonic mean	0.00005327	0.0006431	0.0005578	0.0005715	0.001369	0.001506	0.001709	0.00186	0.002029	0.003791
Lower 95% CI of harm. mean	0.00005227	0.0005701	0.0003408	0.0003567	0.001154	0.001324	0.001456	0.001513	0.001701	0.003122
Upper 95% CI of harm. mean	0.0000543	0.0007377	0.001537	0.001437	0.001684	0.001746	0.002067	0.002413	0.002513	0.004825
Quadratic mean	0.00005332	0.0006678	0.0007443	0.0007754	0.001568	0.001593	0.001844	0.002112	0.002199	0.004096
Lower 95% CI of quad. mean	0.0000523	0.0005843	0.0006326	0.0006375	0.001074	0.0013	0.001447	0.001521	0.00176	0.003462
Upper 95% CI of quad. mean	0.00005432	0.0007419	0.0008412	0.0008922	0.00194	0.00184	0.002169	0.00257	0.002564	0.004645
Skewness	0	0.2414	-2.542	-1.893	1.321	1.47	1.429	1.18	0.79	-0.6409
Kurtosis	4.5	3.193	7.852	3.014	0.2005	3.481	5.183	3.374	3.618	3.24
Sum	0.000533	0.006599	0.00718	0.00741	0.01489	0.01561	0.01796	0.02025	0.02144	0.04012

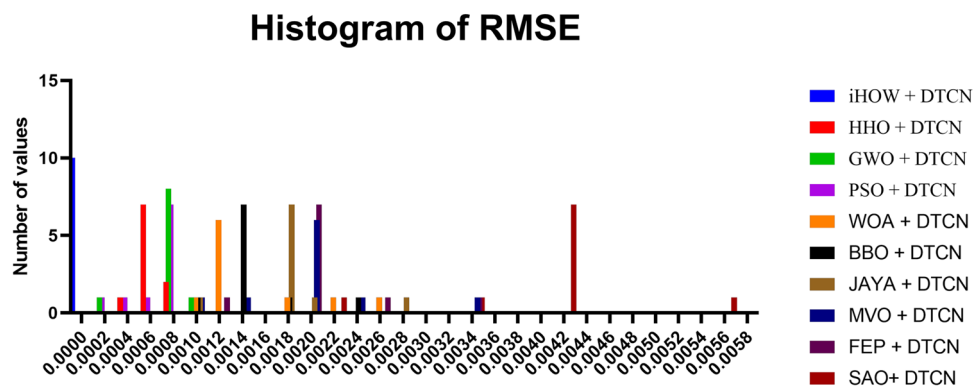
**Table 12.** Descriptive statistics of forecasting errors across 10 runs.

	iHOW + DTCN	HHO + DTCN	GWO + DTCN	PSO + DTCN	WOA + DTCN	BBO + DTCN	JAYA + DTCN	MVO + DTCN	FEP + DTCN	SAO+ DTCN
Theoretical median	0	0	0	0	0	0	0	0	0	0
Actual median	0.0000533	0.000646	0.000758	0.000871	0.001197	0.001474	0.001727	0.001979	0.00206	0.004154
Number of values	10	10	10	10	10	10	10	10	10	10
Wilcoxon Signed Rank Test										
Sum of signed ranks (W)	55	55	55	55	55	55	55	55	55	55
Sum of positive ranks	55	55	55	55	55	55	55	55	55	55
Sum of negative ranks	0	0	0	0	0	0	0	0	0	0
P value (two tailed)	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002
Exact or estimate?	Exact	Exact	Exact	Exact	Exact	Exact	Exact	Exact	Exact	Exact
P value summary	**	**	**	**	**	**	**	**	**	**
Significant (alpha=0.05)?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
How big is the discrepancy?										
Discrepancy	0.0000533	0.000646	0.000758	0.000871	0.001197	0.001474	0.001727	0.001979	0.00206	0.004154

**Table 13.** Wilcoxon Signed-Rank Test Results Comparing iHOW + DTCN to Other Optimizers.



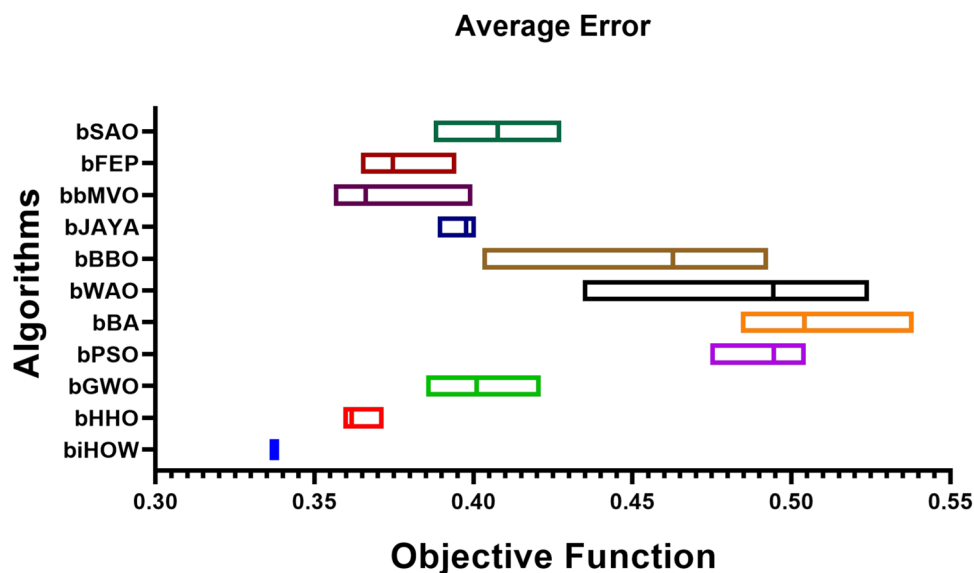
**Fig. 16.** Bar plot of model performance metrics with error bars.



**Fig. 17.** Histogram of MSE values across 10 runs for the iHOW + DTCN model.

ANOVA table	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	0.0001108	9	0.00001231	F (9, 90) = 58.42	P<0.0001
Residual (within columns)	0.00001896	90	0.000002107		
Total	0.0001297	99			

**Table 14.** ANOVA test results for forecasting error across optimizers.



**Fig. 18.** Gantt chart illustrating the temporal distribution of different data segments, highlighting their respective durations and overlaps.

presented in Table 11. These results collectively support the claim that iHOW provides a more effective training dynamic for deep neural forecasting models.

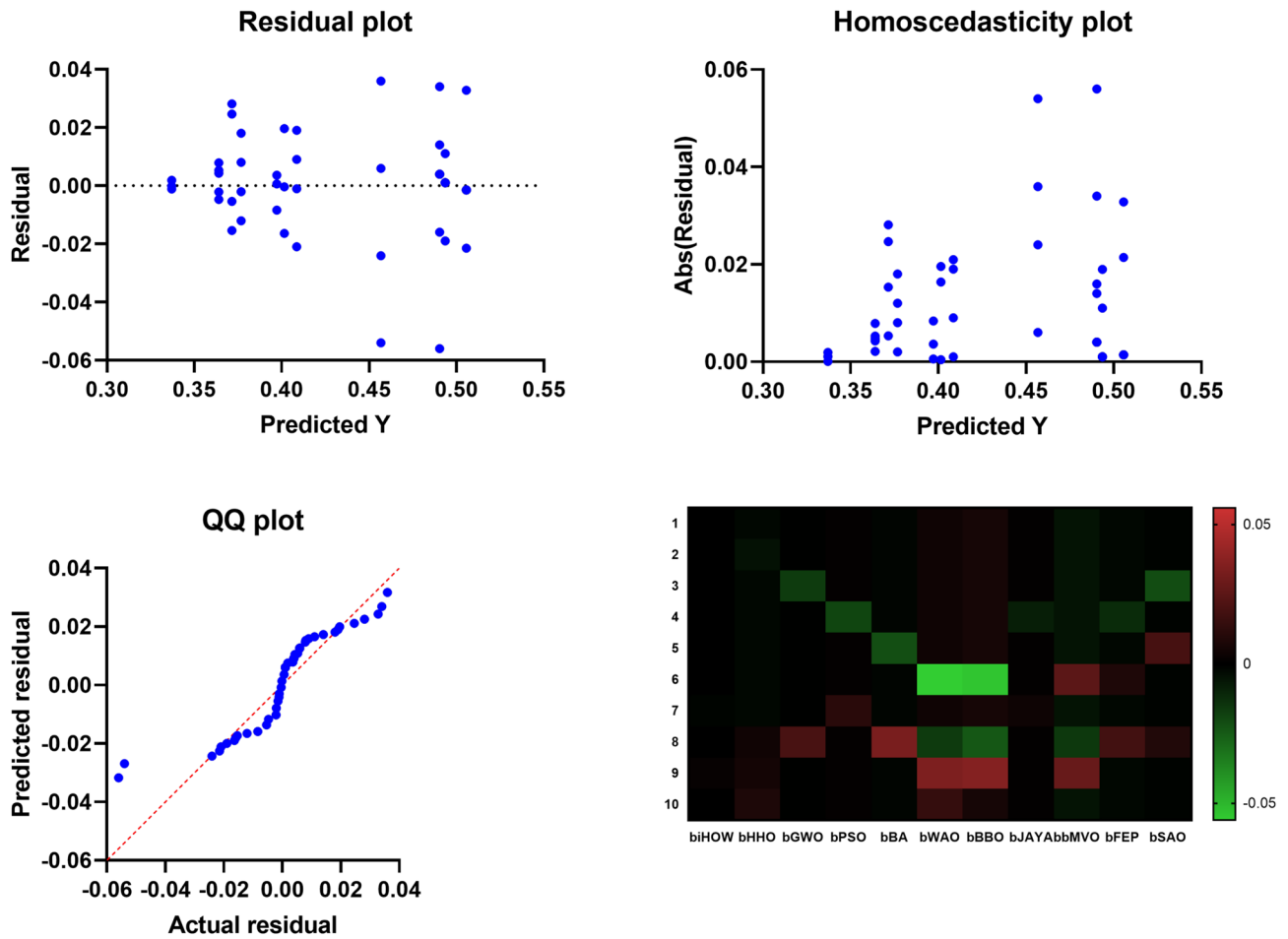
In the field of computational intelligence and metaheuristic optimization, the evaluation of algorithmic performance across multiple resource and efficiency metrics is critical for determining practical applicability. To enable a fair and comprehensive comparison, normalized metrics are often used, ensuring that diverse performance indicators—such as execution time, memory usage, CPU consumption, and overall efficiency—are evaluated on a common scale. Figure 21 presents a radar-based comparative visualization of multiple optimization algorithms, highlighting their normalized performance across five key metrics: average execution time, standard deviation of execution time, memory usage, CPU utilization, and overall efficiency. The visualization enables quick identification of the best-performing algorithm in each category. As illustrated, the *intelligent Hybrid Optimization Wrapper* (iHOW) algorithm consistently outperforms the others across all metrics, achieving the highest efficiency score of 0.925. This indicates not only strong optimization performance but also favorable computational resource consumption. Such insights are vital for guiding the selection of appropriate algorithms in resource-constrained environments or real-time applications.

In addition to numerical comparisons, visualizing the relationships between optimization algorithms and performance metrics can provide deeper insights into their relative similarities and dominant characteristics. Network-based visualization techniques are particularly effective for representing multidimensional performance data, where nodes denote entities (e.g., algorithms or metrics) and edges encode similarities or associations based on empirical results. Figure 22 illustrates the algorithm performance network, where each node represents either a metaheuristic algorithm or a specific evaluation metric. The edges indicate performance similarity, with thicker lines representing stronger associations. This form of graphical analysis allows for identifying algorithm clusters, performance bottlenecks, and key outliers. As highlighted in the figure, the *intelligent Hybrid Optimization Wrapper* (iHOW) is central to the network with strong connections to favorable metrics such as low execution time and high efficiency score. This further reinforces previous findings of iHOW's superior performance. The network density and centrality metrics suggest that iHOW is not only dominant but also consistently aligned with optimal performance criteria across the spectrum of evaluation measures.

Such results are crucial for real-time or resource-constrained energy forecasting applications, where convergence speed and system overhead directly impact feasibility.

## Discussion

The research demonstrates that deep learning architectures and metaheuristic optimization methods can significantly improve the accuracy and dependability of green energy forecasting. With DTCN as the key network

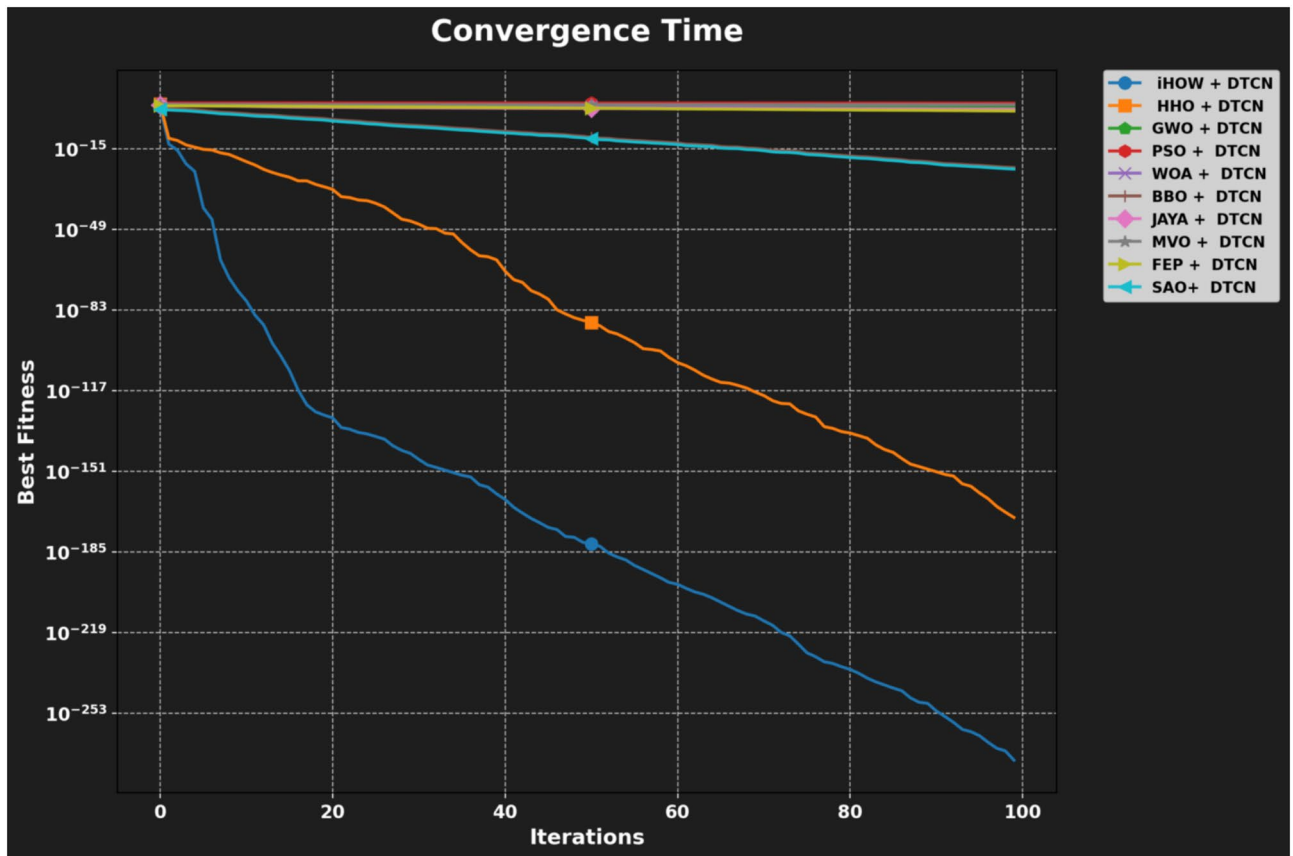


**Fig. 19.** Comprehensive correlation analysis including scatter plots, Q-Q plot, and heatmap for feature interaction and dependency evaluation.

Algorithm	Avg Time (s)	Std Time (s)	Memory (MB)	CPU (%)	Efficiency score
iHOW + DTCN	23.7	1.2	287.4	34.2	0.9247
HHO + DTCN	31.4	2.1	342.8	41.7	0.8634
GWO + DTCN	35.8	2.8	389.1	46.3	0.8291
PSO + DTCN	42.1	3.4	431.5	51.8	0.7856
WOA + DTCN	48.6	4.1	478.3	56.4	0.7423
BBO + DTCN	54.2	4.7	519.7	61.1	0.6987
JAYA + DTCN	61.3	5.3	564.2	65.7	0.6542
MVO + DTCN	67.9	5.9	607.8	69.8	0.6128
FEP + DTCN	73.4	6.4	648.9	74.2	0.5734
SAO + DTCN	79.8	7.1	692.1	78.9	0.5389

**Table 15.** Computational complexity and efficiency metrics of optimizer-DTCN combinations.

and iHOW to improve its performance, we achieved better results than baseline models and others in the same field. Through its approach, this system uncovered nonlinear, multiscale dependencies easily missed by standard forecasting models, providing helpful information for making energy management decisions and managing the grid. The framework is strong because it is designed to deal with the multiple, unclear information found in renewable energy data. By selecting important features and relying on metaheuristic optimization, the team made their model's generalization easier. Because it used only essential features, the model could avoid becoming too complex for its data and ensure more reliable predictions. It became clear when looking at performance evaluations after applying feature selection to the DTCN, which showed it had fewer errors and higher correlation. Moreover, a significant strength of the proposed framework lies in its ability to offer interpretable predictions through explainable artificial intelligence. We adopted SHAP (SHapley Additive exPlanations) to



**Fig. 20.** Convergence curves of optimization algorithms integrated with DTCN over 100 iterations.

understand the impact of each feature on the model's outputs. The SHAP summary plots and feature importance rankings revealed that short-term lag features (e.g., `lag_1`, `lag_2`) and rolling statistical measures (e.g., `rolling_mean_24h`, `cumulative_energy`) had the highest explanatory power. This allowed us to not only enhance the model's forecasting performance but also improve transparency in how predictions are generated. Such explainability is critical for energy planners, as it enables data-driven justification of demand trends, empowers smarter allocation of energy resources, and supports regulatory compliance in intelligent grid systems. The experiments show that iHOW is an efficient way to optimize deep learning models. Among PSO, GWO, HHO and SAO, iHOW-optimized DTCN demonstrated the best outcomes in most evaluation metrics. As a result, the iHOW-related cognitive ideas, such as adaptive learning and knowledge-based exploration, may provide a helpful advantage in solving common optimization challenges within time-series energy forecasting. To further assess the role of each component in the proposed framework, we conducted an ablation study. We evaluated three model variants by selectively removing: (1) the feature selection step, (2) the iHOW optimizer, and (3) the SHAP-based explainability layer. The exclusion of feature selection significantly increased error rates due to irrelevant or redundant inputs. Replacing iHOW with traditional optimizers like PSO or GWO led to slower convergence and lower accuracy, underscoring iHOW's effectiveness in hyperparameter tuning. While removing SHAP did not degrade predictive performance, it eliminated the model's transparency, a key factor in practical deployment. These findings confirm that each module-feature engineering, optimization, and interpretability-contributes meaningfully to the system's overall robustness, efficiency, and usability. While the proposed DTCN+iHOW framework demonstrates strong generalization, stability, and computational efficiency, we acknowledge that full real-time deployment has not been implemented or tested in this study. The current system is evaluated on historical data with rigorous validation strategies, which simulate practical forecasting conditions. However, future work will involve deploying the model in a live energy management environment to assess latency, real-time inference performance, and system integration readiness. Using diverse data sets from ten years of hourly energy demand further shows how the research works in the real world. The framework succeeded in applying its findings to different phases of history and showed that it can be applied to other datasets. These features ensure that forecasting remains constant and correct for actual use in real-time energy monitoring. Overall, this research proposes a valuable and intelligent framework that advances the progress of green energy forecasting. Using DTCN, iHOW and explainable AI, we improve the accuracy of our forecasts and ensure the system becomes more stable, transparent and can be used on a larger scale. Because of these results, it will be possible to link real-time data streams, climate models and hybrid ensemble learning systems in the future, all aimed at improving energy sustainability as the world's energy environment evolves.



**Fig. 21.** Comprehensive performance comparison of optimization algorithms across normalized metrics.

### Conclusion and future work

This paper provided a detailed model to forecast renewable energy, including deep learning methods and optimized metaheuristic techniques. The iHOW-optimized DTCN greatly improves accuracy compared to traditional methods, based on MSE, RMSE, MAE and  $R^2$ . This method helps handle large and messy data, enhances the interpretability of the model, reduces the workload on the computer and resolves the main difficulties present in predicting renewable energy. Integrating these two processes transforms energy prediction and moves the world closer to adopting environmentally friendly power grids. As a result, this research will help inspire further studies in this area. Despite the promising results, this study has several limitations. First, the dataset used is geographically constrained to a specific region and may not generalize to different climates or grid conditions. Second, while the model performs well in evaluation metrics, the computational costs-particularly during optimization-can be high, which may limit real-time deployment on low-power devices or embedded systems. Third, although the proposed framework shows potential for real-world application, it has not yet been tested under live deployment scenarios or integrated with existing energy management systems. These constraints must be addressed to ensure broader applicability and robustness of the approach. Future improvements will focus on deploying the proposed framework in real-world smart grid environments to assess live forecasting performance under streaming data conditions. Additionally, incorporating edge computing can enable decentralized real-time inference, especially in low-resource or rural regions. Transfer learning strategies will also be explored to adapt the model to new regions or countries with limited historical data, enhancing scalability and generalizability. Finally, integrating models for climate variability and weather uncertainty may improve robustness against long-term environmental fluctuations. By focusing on practical, scalable, and deployable techniques, this line of research contributes meaningfully toward building intelligent, adaptive, and sustainable energy forecasting systems. One key limitation of this study is the use of a single dataset from a specific geographic region and energy profile. While this dataset provides rich temporal patterns and historical continuity, it may not fully represent the diversity of operational, climatic, or socio-economic conditions found in other locations. As such, the generalizability of the findings to other energy systems or regions is limited. Future research should validate the proposed framework across multiple datasets from different countries or renewable energy sources (e.g., solar, wind, hydro) to assess its robustness and adaptability. Transfer learning strategies may also be employed to fine-tune the model on new regional datasets with minimal retraining.

Performance Insights:  
 • Best Algorithm: iHOW + DTCN (Eff: 0.925)  
 • Network Density: 8 connections  
 • Efficiency Range: 0.539 – 0.925

## Algorithm Performance Network

Similarity Analysis with Metric Projections & Performance Ranking

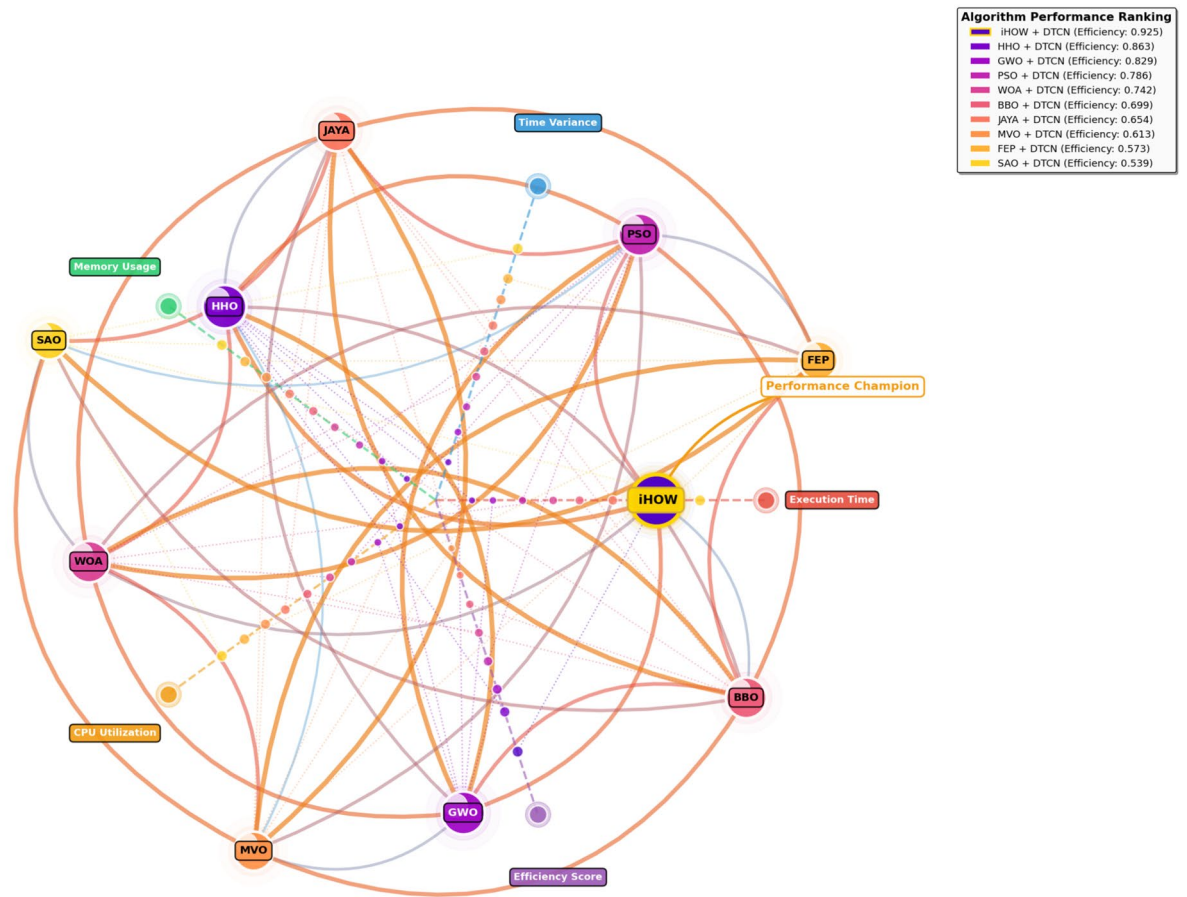


Fig. 22. Algorithm performance network based on similarity analysis across performance metrics.

### Data Availability

The data that support the findings of this study are openly available at ([https://www.kaggle.com/datasets/gauravduttakiit/forecast-green-energy/data?select=train\\_IxoE5JN.csv](https://www.kaggle.com/datasets/gauravduttakiit/forecast-green-energy/data?select=train_IxoE5JN.csv))

Received: 12 June 2025; Accepted: 27 October 2025

Published online: 21 November 2025

### References

- Lu, Y. et al. A critical review of sustainable energy policies for the promotion of renewable energy sources. *Sustainability* **12**(12), 5078. <https://doi.org/10.3390/su12125078> (2020).
- Bei, J. & Wang, C. Renewable energy resources and sustainable development goals: Evidence based on green finance, clean energy and environmentally friendly investment. *Resour. Policy* **80**, 103194. <https://doi.org/10.1016/j.resourpol.2022.103194> (2023).
- Ang, T.-Z. et al. A comprehensive study of renewable energy sources: Classifications, challenges and suggestions. *Energy Strategy Rev.* **43**, 100939. <https://doi.org/10.1016/j.esr.2022.100939> (2022).
- Hoffmann, M., Kotzur, L., Stolten, D. & Robinius, M. A review on time series aggregation methods for energy system models. *Energies* **13**(3), 641. <https://doi.org/10.3390/en13030641> (2020).
- Teichgraber, H. & Brandt, A. R. Time-series aggregation for the optimization of energy systems: Goals, challenges, approaches, and opportunities. *Renew. Sustain. Energy Rev.* **157**, 111984. <https://doi.org/10.1016/j.rser.2021.111984> (2022).
- Rajni, Banerjee, T., Kumar, P. Forecasting of renewable energy production in united states: An arima based time series analysis. In *AIP Conference Proceedings* vol. 3010, 030014 (AIP Publishing LLC, 2024). <https://doi.org/10.1063/5.0193938>.
- Pasari, S. & Shah, A. Time series auto-regressive integrated moving average model for renewable energy forecasting. In *Enhancing Future Skills and Entrepreneurship: 3rd Indo-German Conference on Sustainability in Engineering* 71–77 (Springer International Publishing, 2020).
- Szostek, K., Mazur, D., Działus, G. & Kuszniak, J. Analysis of the effectiveness of arima, sarima, and svr models in time series forecasting: A case study of wind farm energy production. *Energies* **17**(19), 19961073. <https://doi.org/10.3390/en17194803> (2024).
- Kerem, A., Saygin, A. & Rahmani, R. A green energy research: Forecasting of wind power for a cleaner environment using robust hybrid metaheuristic model. *Environ. Sci. Pollut. Res.* <https://doi.org/10.1007/s11356-021-16494-7> (2022).
- Fan, Z., Yan, Z. & Wen, S. Deep learning and artificial intelligence in sustainability: A review of sdgs, renewable energy, and environmental health. *Sustainability* **15**(18), 13493. <https://doi.org/10.3390/su151813493> (2023).

11. Forootan, M. M., Larki, I., Zahedi, R. & Ahmadi, A. Machine learning and deep learning in energy systems: A review. *Sustainability* **14**(8), 4832. <https://doi.org/10.3390/su14084832> (2022).
12. Jin, D., Ocone, R., Jiao, K. & Xuan, J. Energy and ai. *Energy and ai*, 100002 (2020). <https://doi.org/10.1016/j.egyai.2020.100002>.
13. Abdullah, W., Elmasry, A. & Tolba, A. Hybrid attention-enhanced deep learning for accurate hourly energy consumption forecasting. *Inf. Sci. Appl.* **3**, 74–83 (2024).
14. Xiang, X., Li, X., Zhang, Y. & Hu, J. A short-term forecasting method for photovoltaic power generation based on the tcn-ecanet-gru hybrid model. *Sci. Rep.* **14**(1), 6744. <https://doi.org/10.1038/s41598-024-56751-6> (2024).
15. Lara-Benítez, P., Carranza-García, M., Luna-Romera, J. M. & Riquelme, J. C. Temporal convolutional networks applied to energy-related time series forecasting. *Appl. Sci.* **10**(7), 2322. <https://doi.org/10.3390/app10072322> (2020).
16. Liang, J. & Tang, W. Ultra-short-term spatiotemporal forecasting of renewable resources: An attention temporal convolutional network-based approach. *IEEE Trans. Smart Grid* **13**(5), 3798–3812 (2022).
17. Minaei, A. F. & Malik, H. Metaheuristics paradigms for renewable energy systems: advances in optimization algorithms. In *Metaheuristic and Evolutionary Computation: Algorithms and Applications* 35–61 (Springer, 2020).
18. Güven, A. F. & Samy, M. M. Performance analysis of autonomous green energy system based on multi and hybrid metaheuristic optimization approaches. *Energy Convers. Manag.* **269**, 116058. <https://doi.org/10.1016/j.enconman.2022.116058> (2022).
19. Elnaghi, B. E., Ismaiel, A. M., Abdel-Kader, F. E. S., Abelwhab, M. N. & Mohammed, R. H. Validation of energy valley optimization for adaptive fuzzy logic controller of dfwg-based wind turbines. *Sci. Rep.* **15**(1), 711. <https://doi.org/10.1038/s41598-024-82382-y> (2025).
20. Dziri, S., Bouallègue, S., Alhato, M. M. & Siarry, P. Artificial bee colony optimization-based enhancement of output power generation in grid-connected photovoltaic systems. *Meas. Control* <https://doi.org/10.1177/01423312241287755> (2025).
21. Castro-Freibott, R., García-Sánchez, I., Espiga-Fernández, F. & Cruz, G.G.-S. Deep Reinforcement Learning for intraday multireservoir hydropower management. *Mathematics* **13**(1), 151. <https://doi.org/10.3390/math13010151> (2025).
22. Meyer, E. L., Mvokwe, S. A., Oyedeji, O. O., Rono, N. & Agoro, M. A. Computational study of chalcogenide-based perovskite solar cell using SCAPS-1D numerical simulator. *Materials* **18**(1), 186. <https://doi.org/10.3390/ma18010186> (2025).
23. Bokde, N. D. A unified framework for multi-type hydrogen production and storage in renewable energy systems. *Energy Convers. Manag.* <https://doi.org/10.1016/J.ECMX.2024.100847> (2025).
24. Bokde, N. D. Modeling hydrogen production variability. *Energy Convers. Manag.* <https://doi.org/10.1016/J.ECMX.2024.100847> (2025).
25. 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. <https://doi.org/10.3390/su15097087> (2023).
26. Yan, T., Rashid, J., Saleem, M. S., Ahmad, S. & Faheem, M. A hybrid deep learning approach for green energy forecasting in Asian countries. *Comput. Mater. Continua* **81**(2), 58186. <https://doi.org/10.32604/cmc.2024.058186> (2024).
27. Li, P., Zhou, K., Lu, X. & Yang, S. A hybrid deep learning model for short-term pv power forecasting. *Appl. Energy* **259**, 114216. <https://doi.org/10.1016/j.apenergy.2019.114216> (2020).
28. Mubarak, H., Stegen, S., Bai, F., Abdellatif, A. & Sanjari, M. J. Enhancing interpretability in power management: A time-encoded household energy forecasting using hybrid deep learning model. *Energy Convers. Manag.* **315**, 118795. <https://doi.org/10.1016/j.enconman.2024.118795> (2024).
29. Adegboye, O. R. et al. Enhanced multi-layer perceptron for co2 emission prediction with worst moth disrupted moth fly optimization (wmfo). *Heliyon* **10**(11), 31850. <https://doi.org/10.1016/j.heliyon.2024.e31850> (2024).
30. Almsallti, M., Alzubi, A. B. & Adegboye, O. R. Hybrid metaheuristic optimized extreme learning machine for sustainability focused co2 emission prediction using globalization-driven indicators. *Sustainability* **17**(15), 6783. <https://doi.org/10.3390/su17156783> (2025).
31. Neshat, M. et al. Meta wave learner: Predicting wave farms power output using effective meta-learner deep gradient boosting model: A case study from australian coasts. *Energy* **304**, 132122. <https://doi.org/10.1016/j.energy.2024.132122> (2024).
32. Beshia, A. R. M. A., Ojekemi, O. S., Oz, T. & Adegboye, O. R. Plsco: An optimization-driven approach for enhancing predictive maintenance accuracy in intelligent manufacturing. *Processes* **13**(9), 2707. <https://doi.org/10.3390/pr13092707> (2025).
33. Liu, P. et al. Green energy forecasting using multiheaded convolutional lstm model for sustainable life. *Sustain. Energy Technol. Assess.* **63**, 103609. <https://doi.org/10.1016/j.seta.2024.103609> (2024).
34. Babay, M.-A., Adar, M., Chebak, A. & Mabrouki, M. Forecasting green hydrogen production: An assessment of renewable energy systems using deep learning and statistical methods. *Fuel* **381**, 133496. <https://doi.org/10.1016/j.fuel.2024.133496> (2025).
35. Lea, C., Vidal, R., Reiter, A. & Hager, G. D. Temporal convolutional networks: A unified approach to action segmentation. arXiv preprint [arXiv:1608.08242](https://arxiv.org/abs/1608.08242) (2016) <https://doi.org/10.48550/arXiv.1608.08242>
36. Bahdanau, D., Cho, K. & Bengio, Y. Neural machine translation by jointly learning to align and translate. arXiv preprint [arXiv:1409.0473](https://arxiv.org/abs/1409.0473) (2016) <https://doi.org/10.48550/arXiv.1409.0473>
37. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* **9**(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735> (1997).
38. Williams, R. J. & Zipser, D. A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.* **1**(2), 270–280. <https://doi.org/10.1162/neco.1989.1.2.270> (1989).
39. McCulloch, W. S. & Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **5**(4), 115–133. <https://doi.org/10.1007/BF02478259> (1943).
40. El-Kenawy, E. S. M. ihow optimization algorithm: A human-inspired metaheuristic approach for complex problem solving and feature selection. *J. Artif. Intell. Eng. Pract.* **1**(2), 36–53. <https://doi.org/10.21608/jaiep.2024.386694> (2024).
41. Heidari, A. A. et al. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **97**, 849–872. <https://doi.org/10.1016/j.future.2019.02.028> (2019).
42. Mirjalili, S., Mirjalili, S. & Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **69**, 46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007> (2014).
43. Kennedy, J. & Eberhart, R. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks* vol. 4, 1942–1948 (IEEE, 1995). <https://doi.org/10.1109/ICNN.1995.488968>.
44. Mirjalili, S. & Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **95**, 51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008> (2016).
45. Simon, D. Biogeography-based optimization. *IEEE Trans. Evol. Comput.* **12**(6), 702–713. <https://doi.org/10.1109/TEVC.2008.919004> (2008).
46. Venkata Rao, R. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *Int. J. Ind. Eng. Comput.* **7**, 19–34. <https://doi.org/10.5267/j.ijiec.2015.8.004> (2016).
47. Mirjalili, S., Mirjalili, S. & Hatamlou, A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* <https://doi.org/10.1007/s00521-015-1870-7> (2015).
48. Oliveira, C. et al. Fep protocol builder: Optimization of free energy perturbation protocols using active learning. *J. Chem. Inf. Model.* **63**(17), 5592–5603. <https://doi.org/10.1021/acs.jcim.3c00681> (2023).
49. Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. Optimization by simulated annealing. *Science* **220**(4598), 671–680. <https://doi.org/10.1126/science.220.4598.671> (1983).

## Acknowledgements

Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R754), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

## Author contributions

Conceptualization, El-Sayed M. El-Kenawy , Mahmoud Shabrawy ; methodology, El-Sayed M. El-Kenawy; software, El-Sayed M. El-Kenawy; validation, Khaled Sh. Gaber, Doaa Sami Khafaga; formal analysis, Amel Khaled Sh. Gaber , Doaa Sami Khafaga; investigation, Marwa M. Eid ; writing-original draft, Amel Ali Alhussan , Doaa Sami Khafaga , Mahmoud Shabrawy; writing-review & editing, Marwa M. Eid ; visualization, Marwa M. Eid; project administration, El-Sayed M. El-kenawy.

## Declarations

### Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

### Additional Information

**Correspondence and requests for materials** should be addressed to the Corresponding author. **Reprints and permissions information:** Not applicable. **Ethical approval and consent to participate:** Not applicable. **Consent to publish:** Not applicable. **Competing interests:** The authors declare no competing interests. **Funding:** Not applicable.

### Competing interests

The authors declare no competing interests.

### Additional information

**Correspondence** and requests for materials should be addressed to M.S. or E.-S.M.-k.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025