Enhance Solar Power Generation Using Advanced Computational Techniques for Improved Forecast Accuracy and Efficiency

Huiting Wei*‡, Yongguang Chen**, Qi Wang***

*Laboratory and Equipment Management Center, Xuchang University, Xuchang City, Henan Province, 461000, China

**School of Computer Science and Technology, Zhoukou Normal University, Zhoukou City, Henan Province, 466001, China

*** School of Artificial Intelligence, Zhoukou Normal University, Zhoukou City, Henan Province, 466001, China

(weihuiting@xcu.edu.cn, cyg216@163.com, 20141003@zknu.edu.cn)

‡Corresponding Author; Huiting WEI, weihuiting@xcu.edu.cn

Received: 31.05.2025 Accepted:26.06.2025

Abstract- Solar energy forecasting using machine learning (ML) involves creating computational models to predict photovoltaic (PV) system output based on real-time and historical environmental data. This study applies ML regression techniques—Random Forest Regression (RFR), Support Vector Regression (SVR), and Light Gradient Boosting Machine Regression (LGBR)—to model solar energy generation using key input features such as solar irradiance, temperature, humidity, and panel specifications. Feature engineering processes, including missing value imputation, random permutation, and normalization, are employed to enhance input quality. To improve model performance, these regression models are hybridized with Fire Hawk Optimization (FHO), which fine-tunes hyperparameters to boost predictive accuracy. Performance is assessed using metrics such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), R^2, Normalized Mean Square Error (NMSE), and Average Index of Optimization (AIO) across training, validation, and testing phases. In the training phase, the best results are achieved by LGFH (RMSE = 121), RFFH (RMSE = 140), and SVFH (RMSE = 152), showing that FHO integration significantly enhances prediction accuracy. The hybrid models display lower RMSE and improved robustness compared to standalone methods. Despite challenges posed by data variability and model complexity, the proposed methodology demonstrates reliable forecasting capabilities, contributing to improved energy management, grid stability, and better integration of solar power into renewable energy systems. Future work may involve extending the framework to broader datasets and refining optimization for enhanced generalization in diverse environments.

Keywords: Solar Generation, photovoltaic systems, machine learning, metaheuristic algorithm.

1. Introduction

Solar energy is generated by nuclear processes in the sun's core and body. Electromagnetic radiations carry this energy to the surface of the planet. Solar energy is a viable substitute for fossil fuels in energy production since it is a nearly permanent and renewable resource, which may greatly lower emissions [1], [2]. Solar energy may generate electricity in two different ways [3], [4]. The first method involves using the sun as a heat source to produce heat using solar energy. A heat engine, like a steam turbine, is powered by this focused and absorbed heat. The second method involves using photovoltaic cells to directly transform solar energy into electrical power [5], [6]. The cost of employing solar systems, such as photovoltaic cells, is still relatively high, even though solar energy has significantly increased the world's energy output, and its costs have decreased during the past 50 years [7], [8].

Using solar systems to produce freshwater, heat, and electricity all at once is a viable way to increase their economic performance. One important step toward meeting future energy demands is using a hybrid photovoltaic thermal (PV/T) collector, which uses a single device to convert solar energy into heat and electricity. Industries that use electrical and thermal energy, such as water desalination, solar

greenhouses, solar stationary devices, solar heat pumps, and air conditioning systems, benefit greatly from PV/T systems [9], [10]. In terms of economic performance, combining solar and combined cooling, heating, and power (CCHP) systems may not be the best option, but it may be highly beneficial for generating electricity, reducing emissions, and creating high-temperature and high-pressure fluid [11]. High-temperature heat and electricity may be produced efficiently by expanding and commercializing Solar Combined Heat and Power (SCHP) systems and integrating them with industrial facilities [12]. One of the practical and affordable methods of lowering greenhouse gas emissions is the integration of solar energy into cogeneration systems.

In buildings, manufacturing facilities, and farms, combining photovoltaic or thermal collectors with heat pumps to produce electricity, heat, and hot water simultaneously enhances the overall performance of renewable systems [13]. A PV/T system's ability to produce electricity and low-temperature heat simultaneously boosts the system's overall efficiency as well as the efficiency of the solar cells [14], [15], [16]. Another benefit of this clean energy is its use for desalination. Based on water evaporation created by absorbing solar energy, the solar still is the most widely used solar desalination method [17]. In regions with limited freshwater supplies, humidification-dehumidification (HDH) desalination systems provide a renewable method of generating fresh water thanks to their energy source from beaches with high levels of sun radiation [18]. In regions with a lot of available salt water, like the seashore, the use of concentrated solar power (CSP) plants and desalination systems for the simultaneous production of fresh water and electricity can be feasible; however, these systems require a more thorough economic analysis [19]. Compared to traditional fossil fuel-based systems, solar energy desalination is more expensive and produces less water. Therefore, there may be a stronger economic case for these systems if they are used in remote locations on a small scale or if manufacturing is done simultaneously [20], [21]. There are two ways to use solar energy for water desalination systems: either by directly absorbing solar radiation energy and using it to generate heat in devices like solar collectors or by using the electric power generated by devices like PV panels [22].

1.1. Related works

Peng et al. [23] developed differential equations (DE) and long short-term memories (LSTM), which are used to forecast power costs. Nevertheless, there have been problems in using LSTM to solve time series and non-linear regression problems. The goal of this novel evolutionary technique is to efficiently acquire optimum solutions by defining an appropriate hyperparameter for LSTM. The findings imply that the suggested DE-LSTM model outperforms existing models for accurate prediction. Almalaq et al. [24] proposed using temporal frames and hidden network neurons to maximize objective functions using the Novel Hybrid Prediction Approach (NHPA), which combines long-term memory and genetics. Using data sets on residential and commercial buildings from public buildings, the present optimization model has been tested for extremely short-term forecasts. The findings demonstrate that developmental DL models outperform both traditional and sophisticated prediction models.

Wang et al. [25] designed the LSTM-Convolutional Network hybrid deep-learning model implemented for photovoltaic power prediction. The data's time characteristics are extracted first in the proposed hybrid model by the long-term memory network and then by the coevolutionary neural model for spatial data. The results point to a more resilient combination than the Convolutional-LSTM system and a better hybrid prognosis model than a single forecast model. Harbola et al. [26] created convolutional neural networks (CNNs) in one dimension (1D) to forecast the dominating wind speed and wind direction over time. The suggested methods have employed 1D classification techniques that automatically learn their characteristics from their initial wind data collection, in contrast to other systems that utilize regression techniques using manually constructed features to identify speed and direction. Additionally, predicting the dominating trajectory and velocity in this study would be beneficial for wind turbine constructions whose energy output is dependent on the aforementioned characteristics. Aprillia et al. [27] suggested an innovative method for predicting PV power production that combines the slap cluster algorithm (SSA) and a convolutional neural network (CNN). The CNN rating then shows the weather for the next day. Five CNN regression models are developed for various weather conditions. The optimal parameter for each CNN regression may be optimized by developers using the Slap Swarm Algorithm (SSA). Methods for evaluating the effectiveness of the suggested procedures have been compared between the Long-term Neural Memory Network (LSTM-SSA) and the SSA vector support (SVM-SSA). Yu et al. [28] found that by collecting the local data context and producing keys and questions that fit locally, the Convolutional Self-Attention-based Long Short-Term Memory (CA-LSTM) framework greatly increased prediction accuracy. The investigators validate the suggested model's implementation in both PV electricity production and power use forecasts using real data. Designers test their designs extensively. The suggested model has an average Absolute Percentage Error (MAPE) of 10% lower than the canonical-based DNN, LSTM, and LSTM in terms of electrical consumption data. According to the aforementioned research, weather can make it challenging to deliver solar electricity accurately. In recent years, artificial intelligence has gained widespread acceptance and has been applied to several technical fields because deep learning provides a lot of promise for incorporating more in-depth knowledge of probabilistic methodologies.

1.2. Objectives

The objective of this study is to design a precise and effective prediction model for solar energy generation using ML regression techniques. Towards this, several regression models are trained from real-time and historical data, incorporating critical environmental and system-specific parameters such as solar irradiance, temperature, humidity, and panel characteristics. Predictive accuracy is enhanced by integrating these models with optimization techniques,

minimizing error rates, and maximizing model effectiveness. Data preprocessing and feature selection play a critical role in this study in order to enhance input quality and maintain model stability. Three preprocessing techniques, namely, Clean Miss Value, Rand Perm, and Normalization, are applied to make the dataset more credible. The Clean Miss Value method is applied to deal with missing values so that faulty and inconsistent records do not compromise model accuracy. RandPerm is applied to introduce controlled randomness in the data selection to make the model more generalizable and unbiased when assessing its performance. Normalization is applied in order to scale input features in a uniform range so that variables with larger scales do not overwhelm the model and the model converges faster during training. The systematic integration of these preprocessing methods with ML algorithms and optimization techniques increases reliability in solar generation forecasting. The proposed models are validated using precision, error rates, and computational complexity metrics. Ultimately, this method increases the management of renewable energy, supports grid stability, and overcomes challenges in terms of variability in the data and computational complexity in solar energy forecasting.

While FHO has previously been used in ML and energy applications, its integration with RF, SVR, and LightGBM models has not been systematically studied in the context of solar energy forecasting. This study uniquely applies FHO to fine-tune these models and compares their performance across training, validation, and testing phases using robust evaluation metrics and interpretability tools. Furthermore, we perform a detailed analysis of residual distributions and feature importance rankings, offering insights into model behavior and data influence—an aspect often overlooked in optimization-focused energy forecasting research.

2. Data collection

The dataset used in this study comprises 3,000 samples, which were randomly chosen from https://www.kaggle.com/datasets/arunkanagolkar/solargeneration after a rigorous preprocessing phase. The process includes treating missing values with the Clean-Miss Value method, dataset augmentation with Randperm, feature distribution normalization, and outlier removal with the Interquartile Range (IQR) method to enhance the data's reliability and model performance. The Clean-Miss Value method ensures that missing values in the data are handled systematically, reducing biases that could affect model precision. Randperm is a dataset augmentation technique that enhances dataset variety by permuting feature values without changing the underlying distribution. Normalization scales numerical features so that features become uniform across variables, which is crucial in optimizing ML model convergence and stability. The IQR method removes outliers more than 1.5 times the interquartile range, ensuring the dataset is free from skewed distributions that could affect model performance. With such preprocessing techniques, the dataset is cleansed, making it more representative and ensuring that the constructed models can learn from quality, well-balanced information. Preprocessing is critical in ensuring that ML applications deliver reliable, understandable results.

Statistical features, including range, standard deviation, skewness, and relative variability, were computed to determine which variable contributes most to model performance from Table 1. Range, defined as the difference between the maximum and minimum, indicates that AC Power in Watts has the highest range at 329,513, followed by IRR at 1,296.815, DC Current at 954.980, and WIND_Speed at 592.683, with Amb_Temp having the smallest range at 14.877.

Standard deviation, a measure of dispersion in values, ranks AC Power in Watts highest at 86,839.204, followed by IRR at 303.186, DC Current at 254.107, and WIND_Speed at 175.859, with Amb_Temp lowest at 3.776. Skewness, a measure of asymmetry in the distribution, ranks WIND_Speed with the highest absolute value at -1.236 (left-skewed), followed by IRR at 0.653, DC Current at 0.601, and AC Power at 0.547, with Amb_Temp being nearly symmetric at 0.141. Relative variability, the standard deviation divided by the mean, ranks IRR highest at 0.784, DC Current at 0.780, AC Power at 0.745, WIND_Speed at 0.381, and Amb_Temp at 0.160. AC Power in Watts is noteworthy for having a wide range and standard deviation, suggesting a strong impact on model performance, particularly if it is the target variable or a primary predictor. IRR, with the highest relative variability, would be a strong influence in proportion to its mean. DC Current has the same characteristics but less variability, and WIND_Speed's range and skewness suggest a possible influence. In the lack of more model context, AC Power in Watts is most likely the most influential since it has the most extreme absolute variability, which would be a problem for model stability unless it is normalized. In the event that relative variability is more critical, IRR would be more influential. In all situations, Amb_Temp is the least variable. More information about the model or target variable would refine this assessment, but AC Power's scale suggests it is the most influential factor.

Table 1. The Min-Max, Ave, St.Dev., and Skew values of the variables.

Variables	Unit	Value				
		Min.	Max.	Ave.	St. Dev.	Skew.
MODULE_TEMP	°C	18.158	67.415	37.936	12.572	0.284
Amb _Temp	°C	16.364	31.241	23.563	3.776	0.141
WIND _Speed	km/h	2.380	595.063	461.810	175.859	-1.236
IRR	W/m²	4.259	1301.074	386.622	303.186	0.653

DC Current	Amps	0.600	955.580	325.600	254.107	0.601
AC Ir	Amps	7.900	439	157.016	114.481	0.524
AC Iy	Amps	7.700	438.500	156.875	114.444	0.523
AC Ib	Amps	8	438.500	157.027	114.483	0.524
AC Power	Watts	2647	332160	116568.872	86839.204	0.547

2.1. Description of Datasets

The dataset consists of six unique variables, all of which are measurable parameters that would most likely be found in a mechanical or environmental system: Ambient Temperature (Amb Temp): A continuous variable, typically measured in degrees Celsius (°C) or degrees Fahrenheit (°F), representing the temperature in the environment. It can be measured by a thermometer or sensor and is subject to variation by location and time.

Infrared (IR): Likely a continuous measurement in the form of infrared radiation, such as in watts per square meter (W/m^2) or as a temperature from IR emissions (e.g., in °C). This could be heating signatures from objects or surfaces picked up by an IR sensor. AC Current In (AC IB in A): A continuous variable in amperes (A), measuring alternating current (AC), which is the motion of electric charge in an AC circuit, such as to a device that is consuming power or generating it.

DC Current In (DC Current in amperes): A continuous variable for DC in amperes (A), indicating steady electric charge, likely from a battery or DC power source. Model Temperature (Model Temp): A continuous variable, likely in °C or °F, for the temperature of a specific system, component, or model, distinguished from ambient temperature. Wind Speed: A continuous variable, likely in meters per second (m/s) or miles per hour (mph), for measuring movement in the atmosphere, measured employing an anemometer. These variables suggest a dataset associated with a thermal, electrical, and environmental monitoring system.

Fig. 1 shows the relation between the input variables and the output variable.

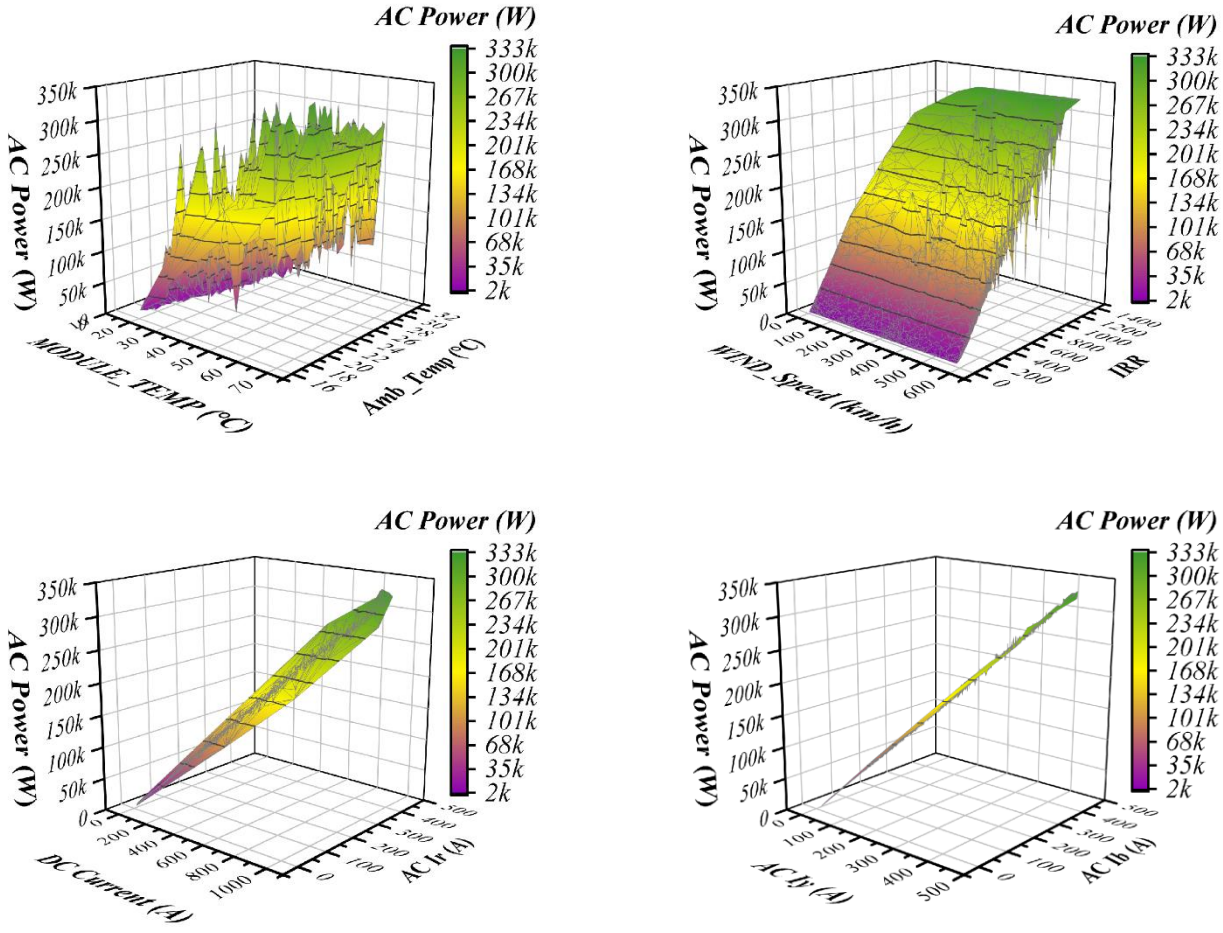


Fig. 1. 3D surface plot for the relation between the variables and the output.

2.2. Data Normalization

Data normalization is a crucial precondition for ANN modeling since it speeds up learning and guarantees that all input features are within a similar range. Eq. (1) has been used to normalize the data set within - 1 and 1:

$$X_{norm} = l + \left[\frac{x - x_{min}}{x_{max} - x_{min}}\right](u - l) \quad (1)$$

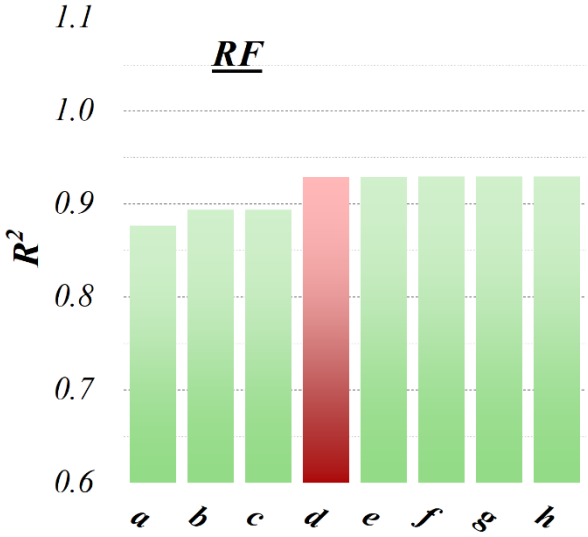
where X_{norm} is the normalized value and x is the observed value. The maximum and minimum values of the parameters are denoted by x_{max} and x_{min}, respectively. The data must be normalized between the lower and upper limit values, denoted by l and u. A lower bound of -1 and an upper bound of 1 were chosen for this work.

2.3. Mean Substitution of Missing Values

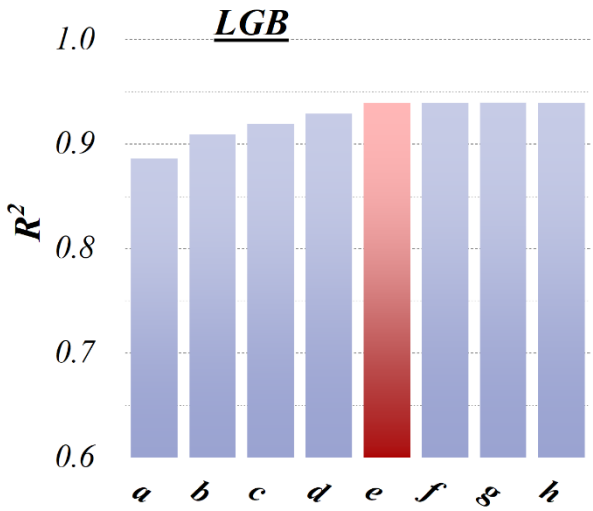
As the information used for model training was collected from multiple research studies, it is inescapable that there are discrepancies in the structure and types of data given, and some studies do not cite all parameters collected as part of the database. However, problems during model training may arise due to missing values. In one example, the authors noticed a large loss of target-prediction pairs when a significant number of missing values were apparent in the training dataset. Thus, if predictor data is missing, the presented MLs may not be able to produce a prediction. In turn, this resulted in fewer comparisons between the network's projected value and the target value of the experiment, which generated concerns about the repeatability of results. For this purpose, the mean replacement approach was applied for all predictors before model training, where parameter missing values are substituted with the mean value [29]. This approach only impacted model training because it was not used for target variables, which is the crucial stage of interest to assess the performance of the models.

2.4. Feature selection

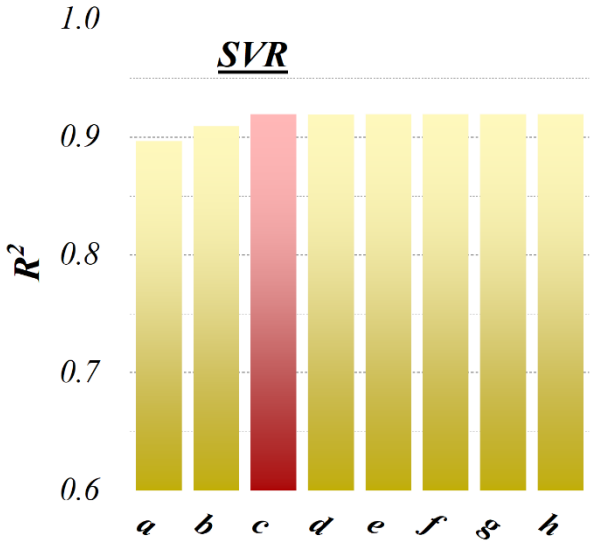
Fig. 2 indicates the study's feature selection process based on Recursive Feature Elimination (RFE). Across RF, LGB, and SVR models, the efficacy of feature selection datasets was quantified by R^2 values for datasets a through h corresponding to features such as MODULE_TEMP, Amb_Temp, WIND_Speed, IRR, DC Current in A, AC Ir in Amps, AC Iy in Amps, and AC Ib in Amps, with ranks and sets corresponding to their ranking and range. In RF, dataset d (IRR, rank 7, set 5,6,7,8) led with $R^2 \approx 0.95$ approx. $R^2 \approx 0.95$, followed by e (rank 3, set 5,6,7,8,4) and f (rank 0, set 5,6,7,8,4,1), both with R^2 less than but very close to 1.0, showing IRR's superiority in capturing variance, most likely due to salience in solar energy systems. LGB was the same, with d at $R^2 \approx 0.95$, e (DC Current in A) at 0.9, and f (AC Ir in Amps) at 0.9, confirming the equal importance of these features, but with lower R^2 values for LGB than for RF. SVR was different, with c (rank 6, set 5,6,7) at $R^2 \approx 0.95$, followed by d and e, both at 0.9, showing SVR's sensitivity to wind-related variance. In all the models, IRR and DC Current in A were highly ranked in all, ranking in the top three in all, while WIND_Speed was a standout for SVR, and AC Ir in Amps was a standout for RF and LGB due to its top rank (0). Altogether, the most effective datasets in all the models are d (IRR), e (DC Current in A), and a tie between c (WIND_Speed) and f (AC Ir in Amps), showing their excellent explanatory power in energy-related contexts.



		Rank	Set	
1	MODULE_TEMP (°C)	4	a	5
2	Amb_Temp (°C)	5	b	5,6
3	WIND_Speed (km/h)	6	c	5,6,7
4	IRR	7	d	5,6,7,8
5	DC Current (A)	3	e	5,6,7,8,4
6	AC Ir (A)	0	f	5,6,7,8,4,1
7	AC Iy (A)	1	g	5,6,7,8,4,1,2
8	AC Ib (A)	2	h	5,6,7,8,4,1,2,3



		Rank	Set	
1	MODULE_TEMP (°C)	4	a	5
2	Amb_Temp (°C)	5	b	5,6
3	WIND_Speed (km/h)	6	c	5,6,7
4	IRR	7	d	5,6,7,8
5	DC Current (A)	3	e	5,6,7,8,4
6	AC Ir (A)	0	f	5,6,7,8,4,1
7	AC Iy (A)	1	g	5,6,7,8,4,1,2
8	AC Ib (A)	2	h	5,6,7,8,4,1,2,3



		Rank	Set	
1	MODULE_TEMP (°C)	4	a	5
2	Amb_Temp (°C)	5	b	5,6
3	WIND_Speed (km/h)	6	c	5,6,7
4	IRR	7	d	5,6,7,8
5	DC Current (A)	3	e	5,6,7,8,4
6	AC Ir (A)	0	f	5,6,7,8,4,1
7	AC Iy (A)	1	g	5,6,7,8,4,1,2
8	AC Ib (A)	2	h	5,6,7,8,4,1,2,3

Fig. 2. The feature selection analyses based on RFE to select the optimal set for the prediction.

3. Methods

3.1. Random Forest Regression (RFR)

An ensemble of predictors such that each tree $h(x{:}\theta_k)$, $k = 1, \ldots, K$, where x is the demonstrated input (values are related) vector of length p and θ_k are independently and identically distributed, is called a random forest [30].

While Y is the main focus of RFR with a numerical result, classification problems are also addressed. The training data, which consists of n p 1-tuples (x_1), is seen as being taken separately from the combined distribution of X and Y. The unadjusted average throughout the acquisition is the random forest forecast for regression: $h(x) = (1/k)\sum_{k=1}^{K} h(x) = (x{:}\theta_k)$. As $k \to \infty$ the Law of Big Figures guarantees:

$$E_{X,Y}(Y - \bar{h}\,(X))^2 \to E_{X,Y}(Y - E_{\,\theta} h(X{:}\theta))^2 \qquad (2)$$

PE_f forests do not overfit, which is the RFR's forecasting (or generalization) mistake, is shown on the right. For each tree $h(X{:}\theta)$, define the mean forecast error as follows:

$$PE_t{=}E_\theta\ E_{\,X,Y}(Y - h(X{:}\theta))^2 \qquad (3)$$

Assuming that the tree is impartial for every θ, that is, $E_Y{=}E_X\ h(X{:}\theta)$, then:

$$PE_f \leq \bar{\rho}\ PE_t \qquad (4)$$

This outlines the requirements for precise random forest regression: (i) There is little association between the residuals of the forest's many trees, and (ii) each tree has a low prediction error. Additionally, it is anticipated that the random forest would result in fewer individual trees. It is anticipated that random forest would decrease individual tree errors PE_t the factor $\bar{\rho}$. As a result, the inserted randomization aims for minimal correlation. The following is the approach taken to achieve these goals:

1. To reduce personal error, cultivate trees as deeply as possible.

2. To lessen residual connection, develop each tree at random utilizing:

a) Example of bootstrapped data from training.

b) Find out how many variables there are ($m \ll p$). Choose the best split based on the m covariates chosen for every node in the tree.

Although technique 1 removes bias, this study contends it does not control variation. This is due to the possibility of unstable maximum trees, which might result in inflated prediction errors. The lack of this conclusion in empirical evaluations of random forests may be due to the characteristics of the UCI repository.

The size of each tree in the forest is determined by the random forest software using a tuning parameter called $nthsize$. The maximum tree size is determined by the number of cases in a node below which the tree will not split. Regression forests are often configured at $nthsize = 5$, which is thought to yield good results. It has been demonstrated that the default setting for classification forests, $nthsize = 1$, consistently yields good results. According to the user handbook, however, larger datasets can employ higher values for efficiency and memory concerns without sacrificing accuracy. The outcomes change when $nthsize$ and the crucial tuning parameter, m, are changed. A tuning parameter is designed to handle overfitting in deep trees and is tested. The nsplit parameter determines the number of splits permitted per tree. Similar control may be achieved by adaptively modifying nthsize; however, this is more challenging to execute. Different split variables and/or cut-points may be present in trees of the same size from the two approaches.

Another justification for limiting the permitted number of splits is boosting. Although some contend that boosting and random forests are comparable, Hastie et al. [31] contend that these parallels are only surface-level and that boosting is distinct. The combination of the latter's results and the former's viewpoint implies that limiting splits per tree could be something to consider. Limiting the allowable tree size resulted in notable gains, according to Hastie et al. [31].

Random forests have a number of characteristics and are quite good at making predictions. Getting precise internal estimates of test set prediction error, distinguishing between black-box predictors (such as neural nets), and measuring covariate relevance are all crucial. Out-of-bag (oob) estimations, not included in present evaluations (Section 4), are used because approximately 1/e ≈ 1/3 of the samples for each tree are bootstrap samples. The test set prediction error can be easily estimated:

1. Use the matching tree to apply each (oob) case, then wait for the response.

2. Calculate $\hat{y}_i$ by averaging the response estimates for each case, i, across trees that I was not familiar with.

3. For regression forests with squared error loss, the test set estimate of PE_f is $1/n\sum_{i=1}^{n}(y_i-(y_2))^2$.

4. The same technique is used with an appropriate loss function for classification problems.

3.2. Support Vector Regression (SVR)

A technique known as support vector machines (SVM) was employed to find patterns. Initial stages in the study of recognition of patterns. Vapnik [32] first suggested this method, which later promoted the use of the SVM algorithm to solve function approximation problems. As a result of this suggestion, the SVR technique, a novel and promising approach to data regression, was eventually developed. A dataset of $\bar{N}$ pieces $\{(X_i, y_i), i = 1,2, \dots, \bar{N}\}$, is used in the SVR approach, where N is the total number of training instances. The i-th component of an N-dimensional variable is characterized by the X_i, where $X_i = \{x_1, x_2, \dots, x_n\} \in R^n$ and $y_i \in R$ is the actual value corresponding to X_i. The fundamental idea behind the SVR is to precisely map training data points, represented by X_i, onto a feature space, which normally has l dimensions, using an ML approach.

In order to precisely determine the non-linear connection between the input independent variables and the output data, the feature space is designed to provide an optimal hyperplane. Eq. (5) formally represents the mathematical formula for SVR.

$$f(x) = V^T \emptyset(x) + a \tag{5}$$

In this case, V is the l-dimensional weight factor, $f(x)$ stands for the estimated values, and a is the adjustable factor. The mapping of each element (X_i) into a subspace with large dimensions is represented by the function $\emptyset(x)$. The ε-insensitive loss function is expressed mathematically in Eq. (6).

$$|y - f(x)|_\varepsilon = \max(0, |y - f(x)| - \varepsilon) \tag{6}$$

Eq. (7) uses the error between the actual value, y, and the anticipated parameter, f(x), to represent the leftover.

$$R(x, y) = y - f(x) \tag{7}$$

As seen below, the optimal regression model is found by including the complete remaining element within a specified range of ε.

$$-\varepsilon \le R(x, y) \le \varepsilon \tag{8}$$

The supposition for the entire training dataset is given by Eq. (8). Consequently, when the residual meets the requirement $R(x, y) = \pm\varepsilon$, the dataset reflects the largest departure from the plane. The next mathematical expression is used to determine the physical distance between a given data point (x, y) and the factor $|R(x, y)|/\|V^*\|$.

$$V^* = (1, -V^T)^T \tag{9}$$

According to the study's premise, the variable δ may describe the largest displacement between the dataset (x, y) and the hyperplane $R(x, y) = 0$. Therefore, it may be concluded that the entire training dataset satisfies the requirements listed in Eq. (10). Reaching the highest value of δ suggests that the SVR model may perform the best in generalization.

$$|R(x, y)| \le \delta\|V^*\| \tag{10}$$

When the function $R(x,y)$'s absolute value matches a predetermined ε value, the greatest distance is attained. It is thus possible to rewrite and express Eq. (11) as Eq. (12). Eliminating. $\|V\|$ is now optimizing issue, meanwhile $\|V^*\|^2 = \|V\|^2 + 1$, and minimizing $\|V^*\|$ is essential to reaching the greatest value of δ.

$$\varepsilon = \delta\|V^*\| \tag{11}$$

There is a chance that certain mistakes may surpass the range of $(-\varepsilon, \varepsilon)$, even if efforts were made during the training phase to decrease errors within this range. Training errors smaller than $-\varepsilon$ are represented by ζ_i, whereas training errors larger than ε are represented by V^*. Eq. (11) and (12), respectively, serve as the foundation for the definitions of the transcriptions ζ_i and $\zeta^*{}_i$.

$$\zeta_i = \begin{cases} 0 & R(x_i, y_i) - \varepsilon \le 0 \\ R(x_i, y_i) - \varepsilon & others \end{cases} \tag{12}$$

$$\zeta^*{}_i = \begin{cases} 0 & \varepsilon - R(x_i, y_i) \le 0 \\ \varepsilon - R(x_i, y_i) & others \end{cases} \tag{13}$$

Finding the hyper-plane that produces the best outcome while minimizing the difference between the training error and the training error is the main goal of SVR. The ε unresponsive loss function is used to achieve this. Eq. (14) is the equation that represents the goal function for optimizing SVR:

$$minF(W, b, \zeta_i, \zeta^*{}_i) = \frac{1}{2}\|W\|^2 + c\sum_{i=1}^{N}(\zeta_i + \zeta^*{}_i) \tag{14}$$

With the limitations:

$$y_i - W^T\varphi(x_i) - b \le \varepsilon + \zeta_i \quad i = 1, 2, \dots, \bar{N}$$

$$W^T\varphi(x_i) + b - y_i \le \varepsilon + \zeta^*{}_i \quad i = 1, 2, \dots, \bar{N}$$

$$\zeta_i \ge 0, \zeta^*{}_i \ge 0 \quad i = 1, 2, \dots, \bar{N}$$

In order to strike a compromise between reducing training mistakes and preserving an ideal margin between the hyperplane space and the learning data points in SVR, parameter C is essential.

The first component of Eq. (10), which maintains a flat regression function, penalizes high-weight values. The ε-insensitive loss function is used in the second section to balance error margins and experience risk.

The factor W is obtained by using the concepts described in Eqs. (15) and (16) once the quadratic optimization problem, which incorporates inequality constraints, has been successfully solved.

$$W = \sum_{i=1}^{N}\left(\beta^*{}_i - \beta_i\right)\varphi(x_i) \tag{15}$$

The Lagrangian boosts must be found by solving a dynamic programming issue in order to compute $\beta^*{}_i$ and β_i.

Eq. (12) provides a mathematical representation of the SVR function.

$$f(x) = \sum_{i=1}^{N}\left(\beta^*{}_i - \beta_i\right)K(x_i - x) + b \tag{16}$$

The kernel function may be used to non-linearly transfer the method of training onto a l-dimensional high-dimensional continuum $K(x_i - x)$.

Since the Kernel function K(x_i-x) has the ability to non-linearly project the training data onto an exhaust system with several measurements (l-dimensions), it is perfectly suited to handle issues involving non-linear interactions. This is particularly useful for electrical prediction [33].

3.3. Light GBM Regression (LGBR)

Light GBM is a new model that employs decision trees and boosting as a gradient learning framework. Light GBM can save memory by discretizing continuous floating-point eigenvalues into k bins and producing a k-width histogram. Light GBM uses a depth-limited leafwise growth technique to optimize the model while maintaining accuracy. Additionally, the approach does not require additional storage for previously sorted results. The precision of a decision tree's segmentation point is meaningless since it is a poor learner. Even higher division points have a normalizing impact that helps prevent overfitting. The level-wise development approach of the decision tree methodology uses a lot of memory that may not be needed to handle all of the leaves in the same layer, making it uneconomical. The leaf-wise approach, which selects the foliage that provides the greatest branching added benefit among all the leaves, shows more success as it moves through the branching process in comparison to the level-wise technique. This method improves accuracy and decreases mistakes in comparable segmentation cycles. The disadvantage of the leaf-wise approach is that it might lead to overfitting by creating deeper decision trees. To avoid overfitting and preserve high efficiency, Light GBM has a maximum depth limit at the top of the leaf [34].

Light GBM aims to estimate $\hat{f}(x)$ of the variable $f^*(x)$ by lowering predicted values of given anything less $L(y, f(x))$using the training dataset $X = \{(x_i, y_i)\}, i = 1, .., m$.

$$\hat{f}(x) \arg min_f E_{y,x}L(y, f(x)) \tag{17}$$

The final model is described in the following steps, and LightGBM employ multiple regression trees $\sum_{t=1}^{T} f_t(X)$ to represent it.

$$f_T(x) = \sum_{t=1}^{T} f_t(X) \tag{18}$$

Regression trees are described using the notation $w_q(x)$, where q can be any value between 0 and 1 and N [35]. The vector w represents the values obtained of the leaf nodes, while q represents the tree's selection method. Eq. (19) is used to train the model modifiers at step t.

$$\eta_t \cong \sum_{j=1}^{N} L(y_i, f_{t-1}(x_i) + f_t(x_i) \tag{19}$$

The Newton approach may be used to estimate the goal function quickly. When the continuous sentence in Eq. (20) is eliminated, it becomes:

$$\eta_t \cong \sum_{j=1}^{N}(u_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i) \quad (20)$$

The different factors u_i and h_i reflect the mathematical outcomes of algorithms at the gradients of order one and two. If the specimen, a group of leaves j, is symbolized by I_j, then Eq. (21) becomes Eq. (22), is represented by:

$$\eta_t = \sum_{j=1}^{J}((\sum_{i\epsilon I_j} u_i)w_j + \frac{1}{2}(\sum_{i\epsilon I_j} h_i + \lambda)w_j^2) \quad (21)$$

Eqs. (22) and (23) provide the largest value of η_t and the optimal leaf weights w_j for the tree design $q(x)$.

$$w_j^* = -\frac{\sum_{i\epsilon I_j} u_i}{\sum_{i\epsilon I_j} h_i + \lambda} \quad (22)$$

$$\eta_t^* = -\frac{1}{2}\sum_{j=1}^{J}\frac{(\sum_{i\epsilon I_j} u_i)^2}{\sum_{i\epsilon I_j} h_i + \lambda} \quad (23)$$

The weight function is defined as the use of weight to assess the structural quality of the tree $q(x)$. To get the goal function, incorporate the separation in the following sequence:

$$O = \frac{1}{2}\left[\frac{\left(\sum_{i\epsilon I_l} u_i\right)^2}{\sum_{i\epsilon I_l} h_i + \lambda} + \frac{\left(\sum_{i\epsilon I_r} u_i\right)^2}{\sum_{i\epsilon I_r} h_i + \lambda} + \frac{(\sum_{i\epsilon I} u_i)^2}{\sum_{i\epsilon I} h_i + \lambda}\right] \quad (24)$$

Therefore, the samples from the left and right branches are denoted by I_r and I_l stand, including both.

3.4. Fire Hawk Optimization (FHO)

This section presents the basic stages of the FHO. Similar to other MH techniques, the FHO usually uses the following formula to determine a beginning value for a group of N agents [36].

$$X_{ij} = rand \times \left(u_j - l_j\right) + l_j\,,\ \ j = 1,2,\ldots.,D \quad (25)$$

Where the ith agent at dimension j is denoted by X_{ij}. The parameters limits at the jth dimension are denoted by U_j and L_j. D is each X_i's dimension, and $rand \in [0,1]$ is a random value.

Next, each X_i performance is assessed using the optimal solution. After that, the best answers are called fire hawks $(FH_l, l = 1,2,\ldots,n)$; the other responses are called prey $(PR_k, k = 1,2,\ldots,m)$. Separating FH from PR is determined using the following equation.

$$D_{lk} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad l = 1,2,\ldots,n \quad k = 1,2,\ldots,m \quad (26)$$

The FH and PR numbers are denoted by m and n, including both. The next step is distributing PR to determine FH's territory. After that, each FH is updated using the Eq. (27).

$$FH_l(t+1) = FH_l(t) + \left(r_1 \times X_b - r_2 \times FH_n(t)\right),\quad l = 1,2,\ldots,n \quad (27)$$

where X_b is the ideal solution and $FH_n(t)$ is one fire hawk. $[0,1]$ is where the random numbers r_1 and r_2 are obtained. The next step is to determine where the prey gathers to stay safe during a hazard. The following expression may be obtained by computing SP_l and SP:

$$SP_l = \frac{\sum_{q=1}^{r} PR_q}{r}\ \ ,q = 1,2,\ldots,r,\quad l = 1,2,\ldots,n \quad (28)$$

The motions of PR_k inside the FH zone are then used to simulate the behavior of animals. Because of this habit, the prey can update its position using the following computation.

$$PR_q(t+1) = PR_q(t) + \left(r_3 \times FH_l - r_4 \times SP_l(t)\right),\quad l = 1,2,\ldots,n\,, q = 1,2,\ldots,r \quad (29)$$

SP_l indicates the safe spot beneath the lth fire hawk's zone. It is necessary to update the safe location outside the lth FH, which is accomplished as follows:

$$SP = \frac{\sum_{k=1}^{m} PR_k}{r}\,, k = 1,2,\ldots,m \quad (30)$$

After that, keep updating the prey's location using the formula below.

$$PR_q(t+1) = PR_q(t) + \left(r_5 \times FH_a - r_6 \times SP(t)\right), l = 1,2,..,n\,,\quad q = 1,2,\ldots,n \quad (31)$$

The procedure of updating the solutions is continued until the stop requirements are true, at which point the best solution, X_b, is returned.

3.5. Evaluation of Metrics

In the current section, the metrics utilized in the assessment of the built models, e.g., RMSE, R^2, MAE, IOA, and NMSE, are explained. These metrics all offer specific advantages in model assessment: RMSE is useful in defining the size of the model error, R^2 is a proportionate measure of the variance explained by the model, MAE is a straightforward indicator of mean model error, IOA is a metric that expresses agreement between forecasted and observed values, and NMSE is a normalization of the error that produces a scale-invariant metric for model comparison. These metrics, in combination, allow for a full assessment of model performance.

$$R^2 = \left(\frac{\sum_{i=1}^{n}(b_i - \bar{b})(m_i - \bar{m})}{\sqrt{\left[\sum_{i=1}^{n}(b_i - \bar{b})^2\right]\left[\sum_{i=1}^{n}(m_i - \bar{m})^2\right]}} \right)^2 \quad \text{Coefficient Correlation } (R^2) \quad (32)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(m_i - b_i)^2} \quad \text{Root Mean Square Error (RMSE)} \quad (33)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|b_i - m_i| \quad \text{Mean Absolute Error (MAE)} \quad (34)$$

$$NMSE = \frac{1}{n}\sum_{i=1}^{n}\frac{(m_i - b_i)^2}{b_i . m_i} \quad \text{Normalized Mean Square Error (NMSE)} \quad (35)$$

$$IOA = 1 - \frac{\sum_{i=1}^{n}(x_{i,a} - x_{i,p})^2}{\sum_{i=1}^{n}(|x_{i,p} - \overline{x_a}| + |x_{i,a} - \overline{x_a}|)^2} \quad \text{Index of Agreement (IOA)} \quad (36)$$

- The predicted value is represented by b_i.
- $\bar{m}$ and $\bar{b}$, respectively, stand for the measured and mean predicted values.
- The measured value is denoted by m_i.

4. Results

In this section, the outcomes from the built models were systematically compared and analyzed in a stringent examination by plots and tables. These were used to visualize differences in performance, highlight trends, and enable critical assessment of the forecasting capability of the models. The trained model hyperparameter settings in Table 2 illustrate the most critical settings that influence their performance. The Random Forest-based model (RFFH) was optimized with a maximum-depth parameter set at 81, which is a deep tree architecture that can accommodate enhanced learning capability but, conversely, could result in overfitting. The model utilized 60 concurrent jobs (n-jobs) and computational efficiency and 69 estimators (n-estimators), which stabilize ensemble learning.

For the Support Vector-based model (SVFH), the hyperparameters consist of a regularization parameter (C) equaling 0.71 and a gamma equaling 0.27. A moderate C value indicates a good balance between margin maximization and classification precision, and the gamma parameter is responsible for regulating the impact of individual training samples, which affects model complexity and generalization.

The Light Gradient Boosting-based model (LGFH) was configured with a maximum-depth parameter at 100, enabling a highly deep decision tree architecture. The learning rate was configured at 0.1, such that the weight update is regulated to prevent overfitting but with successful learning progression. The n-estimators value is configured at -1, which might be either an unspecified parameter or auto-tuned, with the option for tuning according to dataset characteristics for improved model performance.

The FHO algorithm is a population-based metaheuristic inspired by the hunting behavior of fire hawks. It iteratively searches the hyperparameter space to minimize prediction error by balancing exploration and exploitation. The key steps are summarized in Algorithm 1.

Table 2. The hyperparameter results of the developed models

Models	**Hyperparameter**		
	max-depth	n-jobs	n-estimators
RFFH	81	60	69
	C	gamma	
SVFH	0.71	0.27	
	max-depth	n-estimators	learning-rate
LGFH	100	-1	0.1

Algorithm 1. Key step of the hyperparameter tunning.

Initialize the population of candidate hyperparameter sets

While stopping criteria are not met:

Evaluate the fitness (validation error) of each candidate

Update candidate positions using FHO rules (chasing, attacking, evading)

Retain best candidates

Return hyperparameter set with best fitness

Table 3 presents the results of 5-fold cross-validation for three machine learning models—SVR, RF, and LGB—evaluated using R^2 and RMSE metrics. Among these, LGB consistently achieves the highest R^2 values (ranging from 0.909 to 0.936) and the lowest RMSE values (22,660.52 to 25,745.43), indicating strong predictive performance and generalization ability across all folds. RF shows moderate stability with R^2 between 0.893 and 0.919 and RMSE ranging from 26,137.87 to 27,538.88, suggesting reliable but slightly less accurate performance than LGB. In contrast, SVR exhibits the lowest R^2 values (0.881 to 0.908) and the highest RMSE (28,077.13 to 28,941.19), implying greater sensitivity to data partitioning and weaker generalization. In summary, LGB outperforms RF and SVR in both accuracy and stability, making it the most suitable model among the three for reliable prediction.

Table 3. Result of the K-fold cross-validation.

Metric	Model	K1	K2	K3	K4	K5
R^2	SVR	0.908	0.907	0.881	0.895	0.892
	RF	0.917	0.919	0.893	0.91	0.905
	LGB	0.933	0.936	0.909	0.922	0.927
RMSE	SVR	28077.13	28679.75	28941.19	28500	28300
	RF	26137.87	26138.16	27538.88	26838.02	26738.5
	LGB	23335.25	22660.52	25745.43	24500	23800

The performance of the proposed models in Table 3 was measured in various phases with RMSE, R^2, MAE, IOA, and NMSE metrics. LGFH was optimal in the training phase with the lowest RMSE (12148.537) and MAE (7375.441) and the highest R^2 (0.981), which indicates negligible error and excellent prediction capability. RFFH was second, with RMSE being equal to 14068.291 and R^2 equal to 0.974, indicating competitive performance. SVR was the worst, with the highest RMSE (28077.134) and the lowest R^2 (0.908), which indicates larger prediction errors and lower reliability. RF and LGB were moderate in their precision, with RF having a larger RMSE (26137.873) than their optimized forms.

During the validation, LGFH continued performing best, with the highest IOA (0.995) and lowest NMSE (586771.636), showing strong consistency in prediction. Similarly, in testing, LGFH continued with superior performance, with IOA reaching 0.994 and the lowest NMSE (596471.409), affirming its strength. RFFH continued with strong performance, with IOA reaching 0.993 and NMSE reaching 744633.149 in testing. SVR, in contrast, continued with the lowest results, with the lowest IOA (0.968) and highest NMSE (3310641.841), with huge errors. RF and LGB continued with moderate performance, with RF worsening in testing with lower IOA (0.971) and higher NMSE (2997588.324).

In all phases, LGFH was the most performing model, with the highest accuracy and stability. LGFH achieved the lowest RMSE and MAE in training and the highest IOA and lowest NMSE in validation and testing. LGFH was, therefore, the most stable model, with the highest predictive accuracy and generalization in all datasets.

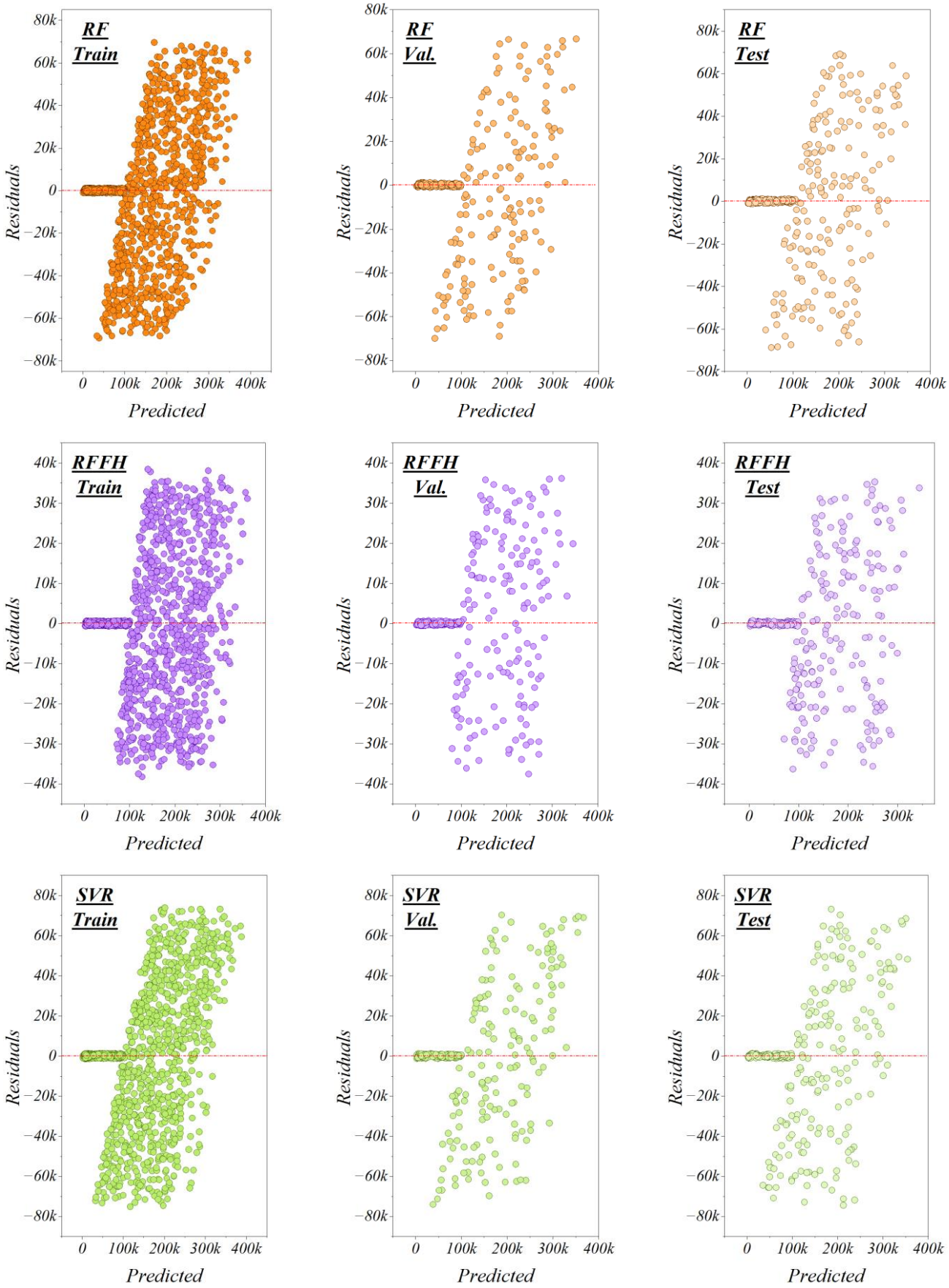
Although the RMSE differences among the hybrid models are modest, the superior performance of LGFH can be attributed to the inherent strengths of Light Gradient Boosting. LGBM efficiently constructs additive models in a forward stage-wise manner, which enables it to capture complex, non-linear feature relationships. When combined with the Fire Hawk Optimization algorithm, which effectively searches for optimal hyperparameters by balancing exploration and exploitation, LGFH gains enhanced predictive power and reduced overfitting. This explains its slight performance edge over RFFH and SVFH models.

Table 4. The performance of the developed models based on 5 evaluation metrics

Model	**RMSE**	**R^2**	**MAE**	**IOA**	**NMSE**
	Training Metrics				

RF	26137.87	0.917	15870.401	0.978	337710.53
RFFH	14068.29	0.974	8562.637	0.994	97833.32
SVR	28077.13	0.908	17128.052	0.975	389681.39
SVFH	15279.92	0.971	9295.295	0.992	115410.71
LGB	23335.25	0.933	14239.044	0.982	269171.51
LGFH	12148.54	0.981	7375.441	0.995	72954.50
	Validation Metrics				
RF	26138.16	0.919	15406.696	0.978	2700407.976
RFFH	14421.32	0.974	8771.138	0.993	822033.584
SVR	28679.75	0.907	17581.040	0.974	3251098.573
SVFH	14554.65	0.973	8904.916	0.993	837303.392
LGB	22660.52	0.936	13707.295	0.983	2029641.644
LGFH	12184.14	0.981	7437.806	0.995	586771.636
	Testing Metrics				
RF	27538.88	0.893	17224.229	0.971	2997588.324
RFFH	13725.60	0.971	8782.053	0.993	744633.149
SVR	28941.19	0.881	18014.348	0.968	3310641.841
SVFH	15630.85	0.963	9771.551	0.991	965705.815
LGB	25745.43	0.909	16540.114	0.975	2619870.033
LGFH	12284.43	0.978	7499.004	0.994	596471.409

The residual plots' examination, in Fig. 3, for the six "RF Train," "RFFH Train," "SVR Train," "SVFH Train," "LGB Train," and "LGFH Test" reveals differing degrees of performance. "LGFH Test" is the best, with a tight residual spread from -40k to 40k, most points between -10k and 10k, minimal heteroscedasticity, and minimal outliers at ±30k. Its randomness and testing in a test set show better generalization, making it the most reliable for real prediction, most likely through successful tuning or feature management in the LightGBM variant. "RFFH Train" and "SVFH Train" come in as strong runners-up, with both having tight spreads (-40k to 40k), minimal heteroscedasticity, and outliers at ±30k, with optimal randomness scatter. However, as training set metrics, they could overfit, and their testing is untested, barely dipping below the "LGFH Test." The "FH" modification—highly likely feature management- enhances all these models' performance. "RF Train" is in the middle range, with residuals from -80k to 60k, a definite funnel shape indicating heteroscedasticity, and outliers at -80k and 60k. While it is free from systematic bias, with a wider spread and larger errors, it is less accurate than the top performers but better than the bottom performers. The base Random Forest model is good but could be optimized further. "SVR Train" and "LGB Train" come in as the bottom performers, with wide residual spreads (-80k to 80k), much heteroscedasticity, and far-reaching outliers at ±80k. Their poor predictive precision and resilience likely stem from inadequate tuning or feature management, with "SVR Train" worse by a little through the maximum outlier range. The superiority of the "FH" versions emphasizes the merit of such enhancements, with the non-"FH" versions failing to capture the complexity of the dataset accurately.


RF Train
RF Val.
RF Test
RFFH Train
RFFH Val.
RFFH Test
SVR Train
SVR Val.
SVR Test
Residuals
Predicted

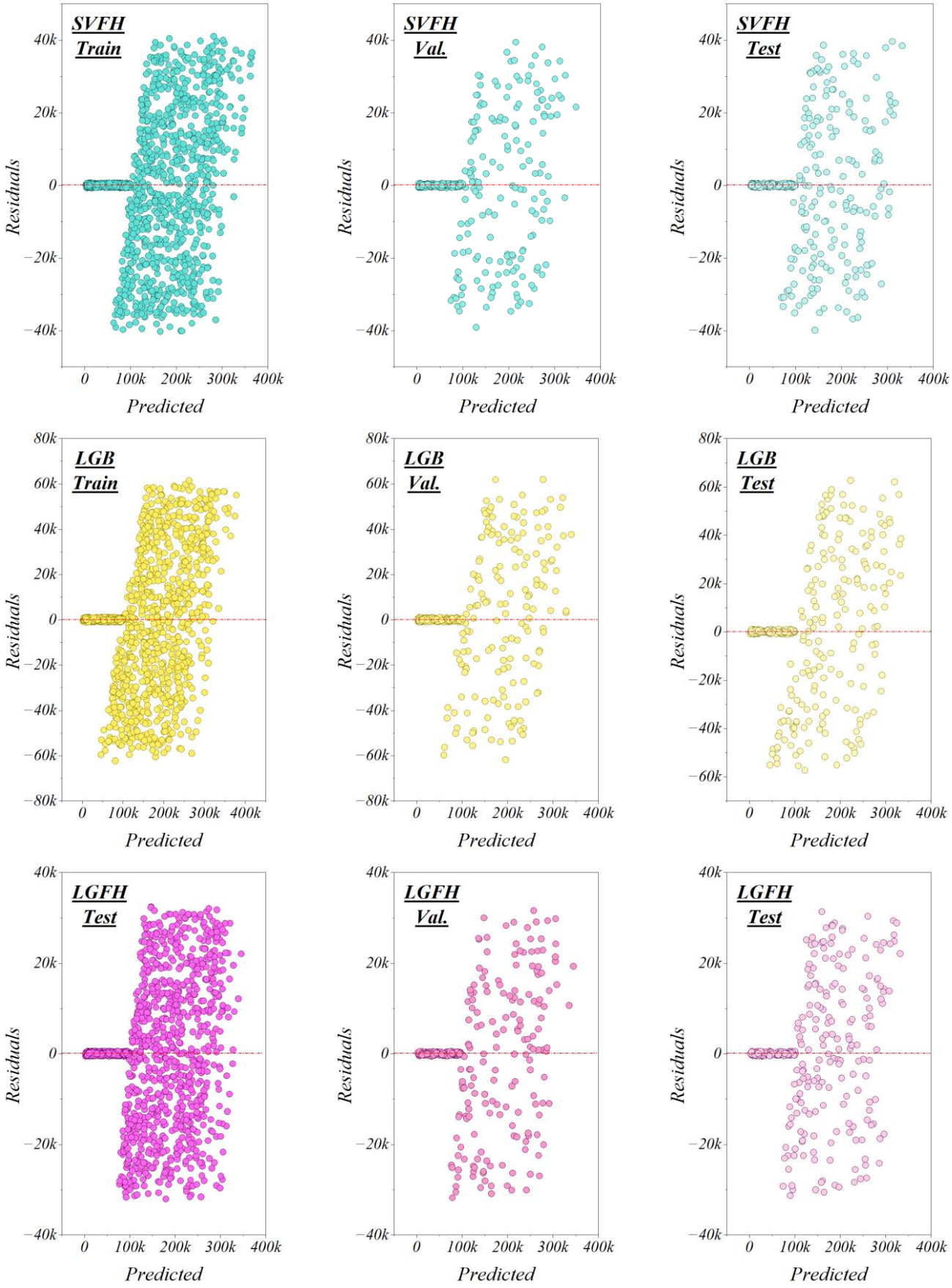


Fig. 3. Residual plot for the accuracy of the developed models in training, validation, and testing phases.

The three rug plots compare "Train," "Validation," and a third dataset (RFH, SVFH, LGFH) error percentages through kernel density estimation in Fig. 4. In the first plot (RFH), "Train" and "Validation" overlap very tightly around 0% error with narrow spread within ±20%, indicating excellent generalization. Nevertheless, "RFH" is the widest, reaching ±40%, indicating very high variability and occasional larger errors. The second plot (SVFH) also overlaps "Train" and "Validation" very tightly, with around 0% error, with most density within ±20%. "SVFH" is wider, reaching ±30%, but less so than "RFH," indicating moderate variability. The third plot (LGFH) follows the same pattern, with "Train" and "Validation" overlapping very tightly around 0% error within ±15%, while "LGFH" reaches ±30%, with variability between "RFH" and "SVFH." In all plots, "Train" and "Validation" show strong model performance with little bias since their symmetric distributions peak at 0%. The third datasets differ: "RFH" is the most variable, "SVFH" is the least variable, and "LGFH" is in between. The x-axis ranges (RFH and SVFH to ±40%, LGFH to ±30%) and peak heights (LGFH at 0.08, RFH and SVFH at 0.07) reveal variations in sample sizes or bandwidths. The narrowing spread from "RFH" through "SVFH" to "LGFH" indicates possible improvements in feature management or model tuning. "SVFH" is best, with the most tightly distributed error among third datasets, indicating stable performance. "LGFH" is middling, with a balanced spread, but "RFH" is the worst, with the most variability and sensitivity to features or noise. These observations lead towards the "SVFH" setup being the most reliable in minimizing error variability, while "RFH" is in need of further investigation to correct for inconsistency.

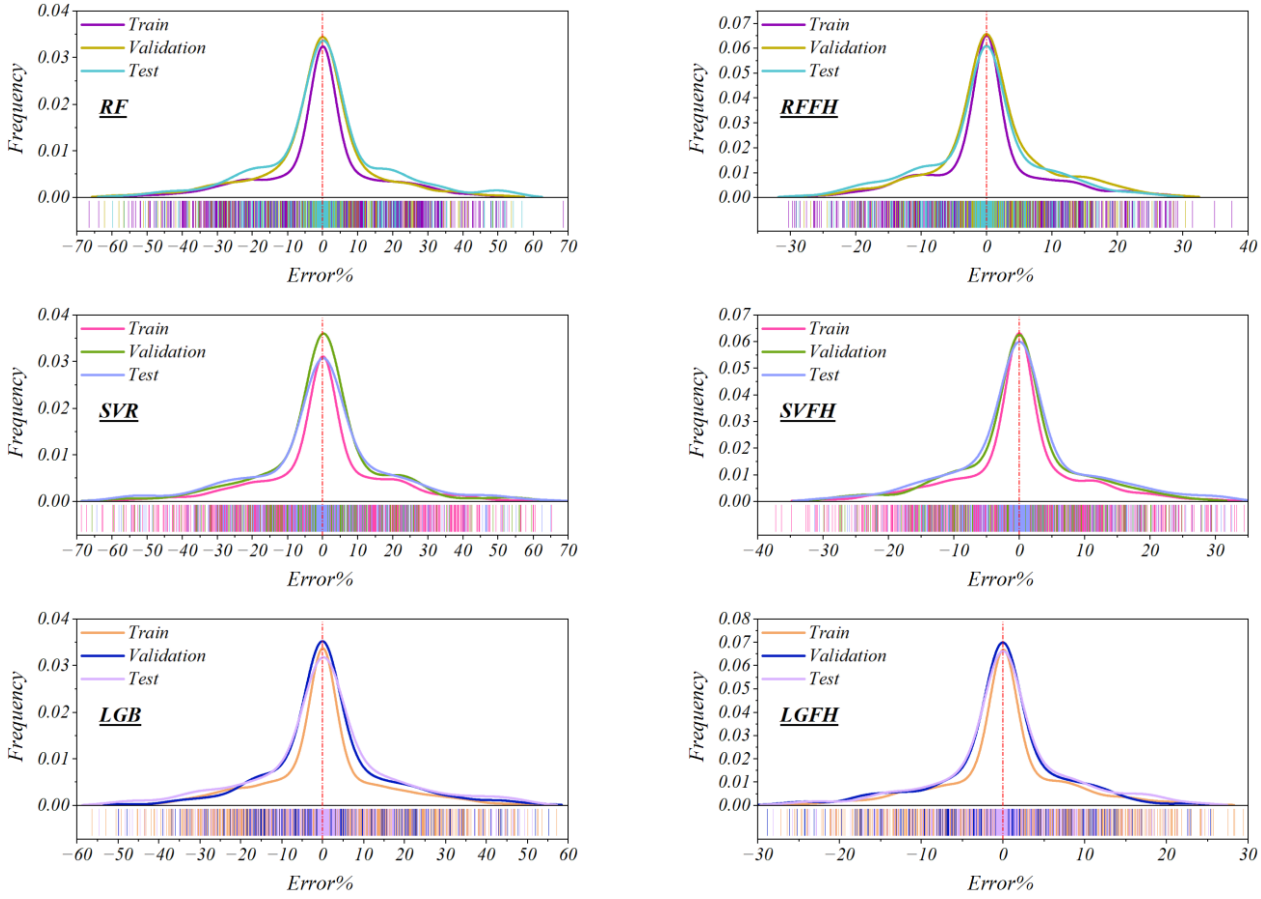


Fig. 4. Kernel density estimation (rug plots) of error percentages for the developed models across train, validation, and test datasets.

Table 5 presents the results of ANOVA-based statistical analysis for six predictive models. All models show statistically significant performance differences, as indicated by P-values below the 0.05 threshold. Among them, the SVFH model achieved the highest F-value (41942.99) with a P-value of 2.14E-07, demonstrating the most substantial variance and strongest statistical significance. LGFH (F = 158.93, P = 4.04E-07) and RFFH (F = 98.0, P = 6.18E-05) also exhibited highly significant differences, suggesting that the hybrid models substantially outperform their base versions. While SVR, LGB, and RF models also showed significant results, their lower F-values indicate less pronounced but still meaningful differences. Overall, the ANOVA results confirm that all model variations differ significantly in performance, with hybrid models offering superior statistical strength.

Table 5. Statistical analyses based on ANOVA.

Models	F-value	P-value
RFFH	98.0	6.18E-05
RF	17.40281	0.001635
SVFH	41942.99	2.14E-07
SVR	84.85646	0.002289
LGFH	158.9277	4.04E-07

LGB	60.4546	0.001021

5. Discussion

Solar forecasting with ML algorithms has significant practical implications. With accurate forecasting of solar generation, utility companies can optimize grid management, reduce the use of fossil fuels, and maximize the efficiency of energy transmission. The predictive function facilitates the integration of renewable energy with existing grids, ensuring the balance between demand and supply with minimal wastage. Policymakers obtain evidence-based inputs from such models in favor of incentives for solar adoption and investment in infrastructure. In real-life applications, such algorithms can be applied by solar farm operators and energy suppliers to forecast generation capacity in response to varying weather conditions. They can, for instance, optimize activity in terms of forecasted cloud cover or seasonal variations, ensuring a steady power supply. Domestic solar consumers could also obtain benefits through home automation systems that optimize energy usage in a forecasted generation. The reliability of such models is a function of the quality of input information, such as historical solar irradiance, temperature, and panel efficacy, and the generalizability of the selected algorithm across a variety of conditions. More advanced approaches, such as neural networks or ensemble methods, tend to be superior in their predictive function, but their reliability is subject to robust training sets and regular validation.

There are, nonetheless, restrictions. ML algorithms can be baffled by rare, unusual, and extreme weather events, as well as by unseasonal patterns that were inadequately captured in training sets. Computational complexity and the need for real-time processing can also be a limitation, particularly for smaller operators with limited resources. Additional studies could be focused on model flexibility through real-time satellite inputs, as well as scalability for mass application. Hybrid models that combine physical simulations with ML can further improve precision. The relevance of the study is in the contribution made towards sustainable energy transitions. With solar power expanding globally, such optimization tools improve energy security, economic viability, and climate targets, making them a cornerstone for modern energy systems.

6. Conclusion

This study demonstrates the potential of machine learning (ML) models, particularly when optimized using metaheuristic techniques, for accurately forecasting solar energy generation. By leveraging a comprehensive dataset of environmental and operational variables—including irradiance, temperature, and electrical currents—three regression models (Random Forest Regression, Support Vector Regression, and Light Gradient Boosting Machine Regression) were developed and enhanced using the Fire Hawk Optimization (FHO) algorithm. This hybridization significantly improved predictive accuracy across all models, especially in testing scenarios. Among the developed models, the LGFH (LightGBM + FHO) model consistently outperformed its counterparts, achieving the highest R^2 value of 0.981 during the validation phase. This result underscores the strength of gradient boosting frameworks in capturing complex non-linear patterns, especially when optimized with intelligent search strategies such as FHO. The comparative analysis revealed that irradiance (IRR) was the most influential feature across all models, followed closely by datasets ‘e’ and ‘c’, each demonstrating strong predictive relevance depending on the regression method applied. These findings provide valuable insight into which input features drive the highest forecasting accuracy, aligning with domain knowledge of solar power systems. The implications of this research are both practical and academic. On a practical level, improved solar forecasting contributes to more efficient grid management, reduced dependence on fossil fuels, and better alignment between energy production and demand. On an academic level, this study bridges data science and renewable energy forecasting, offering a reproducible hybrid approach that balances model complexity, interpretability, and computational efficiency. Furthermore, it highlights the significance of hyperparameter tuning in small- to medium-sized datasets—a frequent constraint in real-world solar applications. While the models demonstrated robust performance, the relatively small dataset remains a limitation for broader generalization. Future work should explore the inclusion of larger, more diverse datasets and assess the comparative performance of this hybrid approach against deep learning architectures, such as Transformer-based or physics-informed models. These directions could further validate the generalization capabilities and improve real-time forecasting reliability. In conclusion, the integration of ML models with metaheuristic optimization presents a promising strategy for enhancing solar power forecasting, offering both technical accuracy and operational value in advancing clean energy adoption.

References

[1] M. D. A. Munnaf and T. Islam, “Enhancing Solar Energy Prospects: Predicting Direct Normal Irradiance in Qinghai Province Using ALO-RF Modeling,” Advances in Engineering and Intelligence Systems, vol. 003, no. 01, pp. 85–98, 2024, doi: 10.22034/aeis.2024.444686.1172.

[2] A. Al Kassem, M. Al Ahmadi, and A. Draou, “Modeling and Simulation Analysis of a Hybrid PV-Wind Renewable Energy Sources for a Micro-Grid Application,” in 2021 9th International Conference on Smart Grid (icSmartGrid), 2021, pp. 103–106. doi: 10.1109/icSmartGrid52357.2021.9551215.

[3] N. H. Alombah, J. N. Mungwe, A. Harrison, W. F. Mbasso, and H. B. Fotsin, “Advanced IoT-Based Monitoring System for Real-Time Photovoltaic Performance Evaluation: Conception, Development and Experimental Validation,” Sci Afr, p. e02763, 2025.

[4] D. Y. Liu, J. Wang, X. Q. Feng, S. Guo, and C. Xu, “Investigation and analysis on the combined operation of solar thermal power and conventional thermal power,” in 2009 International Conference on Sustainable Power

Generation and Supply, 2009, pp. 1–6. doi: 10.1109/SUPERGEN.2009.5348131.

[5] N. H. Alombah, A. Harrison, W. F. Mbasso, H. Belghiti, H. B. Fotsin, P. Jangir, S. F. Al-Gahtani, and Z. M. S. Elbarbary., "Multiple-to-single maximum power point tracking for empowering conventional MPPT algorithms under partial shading conditions," Sci Rep, vol. 15, no. 1, p. 14540, 2025.

[6] Y. Iwasaki, K. Yukita, T. Hosoe, and K. Ikeda, "Basic study for the construction of a microgrid using small wind turbines as the main power source [PDF Not Yet Available In IEEE Xplore]," in 2021 9th International Conference on Smart Grid (icSmartGrid), 2021, p. 1. doi: 10.1109/icSmartGrid52357.2021.9551236.

[7] H. A. Kazem, A. H. A. Al-Waeli, M. T. Chaichan, K. Sopian, A. S. Al Busaidi, and A. Gholami, "Photovoltaic-thermal systems applications as dryer for agriculture sector: A review," Case Studies in Thermal Engineering, vol. 47, p. 103047, 2023.

[8] V. Kamble, A. Bansode, T. Waghmare, and K. Bamane, "Development of Hybrid Compensators for Enhancing Power Quality in Electrical Distribution Systems Using Innovative Optimization Techniques," 2025.

[9] X. Wei, A. Harrison, A. T. Naser, W. F. Mbasso, I. Dagal, N. H. Alombah, P. Jangir, M. Sharaf, and M. El-Meligy, "A new intelligent control and advanced global optimization methodology for peak solar energy system performance under challenging shading conditions," Appl Energy, vol. 390, p. 125808, 2025.

[10] L. Navinkumar Rao, R. Somalwar, K. Joshi, and P. Sarathi Subudhi, "An Improved MPPT algorithm for Performance Improvement of 50 kW Grid Connected PV System," 2025.

[11] J. Wang, Z. Han, and Z. Guan, "Hybrid solar-assisted combined cooling, heating, and power systems: A review," Renewable and Sustainable Energy Reviews, vol. 133, p. 110256, 2020.

[12] A. Kasaeian, G. Nouri, P. Ranjbaran, and D. Wen, "Solar collectors and photovoltaics as combined heat and power systems: A critical review," Energy Convers Manag, vol. 156, pp. 688–705, 2018.

[13] A. Badiei, Y. G. Akhlaghi, X. Zhao, S. Shittu, X. Xiao, J. Li, Y. Fan, and G. Li, "A chronological review of advances in solar assisted heat pump technology in 21st century," Renewable and Sustainable Energy Reviews, vol. 132, p. 110132, 2020.

[14] M. M. Hasan, S. Hossain, M. Mofijur, Z. Kabir, I. A. Badruddin, T. M. Yunus Khan, and E. Jassim, "Harnessing solar power: a review of photovoltaic innovations, solar thermal systems, and the dawn of energy storage solutions," Energies (Basel), vol. 16, no. 18, p. 6456, 2023.

[15] B. Anand, R. Shankar, S. Murugavelh, W. Rivera, K. M. Prasad, and R. Nagarajan, "A review on solar photovoltaic thermal integrated desalination technologies," Renewable and Sustainable Energy Reviews, vol. 141, p. 110787, 2021.

[16] J. Chen, A. Harrison, N. H. Alombah, W. F. Mbasso, R. J. J. Molu, A. M. Alharbi, and P. Jangir, "Irradiance sensorless PSO-based Integral Backstepping and Immersion & invariance algorithm for robust MPPT control with real-climatic microcontroller-in-the-loop experimental validation," Computers and Electrical Engineering, vol. 123, p. 110279, 2025.

[17] Z. Rahimi-Ahar, M. S. Hatamipour, and L. R. Ahar, "Air humidification-dehumidification process for desalination: A review," Prog Energy Combust Sci, vol. 80, p. 100850, 2020.

[18] A. Omar, A. Nashed, Q. Li, G. Leslie, and R. A. Taylor, "Pathways for integrated concentrated solar power-Desalination: A critical review," Renewable and Sustainable Energy Reviews, vol. 119, p. 109609, 2020.

[19] A. Bamasag, E. Almatrafi, T. Alqahtani, P. Phelan, M. Ullah, M. Mustakeem, M. Obaid, and N. Ghaffour, "Recent advances and future prospects in direct solar desalination systems using membrane distillation technology," J Clean Prod, vol. 385, p. 135737, 2023.

[20] H. S. Son, M. W. Shahzad, N. Ghaffour, and K. C. Ng, "Pilot studies on synergetic impacts of energy utilization in hybrid desalination system: multi-effect distillation and adsorption cycle (MED-AD)," Desalination, vol. 477, p. 114266, 2020.

[21] S. H. Son HyukSoo, M. W. Shahzad, N. G. Noreddine Ghaffour, and N. K. Ng KimChoon, "Pilot studies on synergetic impacts of energy utilization in hybrid desalination system: multi-effect distillation and adsorption cycle (MED-AD).," 2020.

[22] A. Modi, F. Bühler, J. G. Andreasen, and F. Haglind, "A review of solar energy based heat and power generation systems," Renewable and Sustainable Energy Reviews, vol. 67, pp. 1047–1064, 2017.

[23] L. Peng, S. Liu, R. Liu, and L. Wang, "Effective long short-term memory with differential evolution algorithm for electricity price prediction," Energy, vol. 162, pp. 1301–1314, 2018.

[24] A. Almalaq and J. J. Zhang, "Evolutionary deep learning-based energy consumption prediction for buildings," ieee access, vol. 7, pp. 1520–1531, 2018.

[25] K. Wang, X. Qi, and H. Liu, "Photovoltaic power forecasting based LSTM-Convolutional Network," Energy, vol. 189, p. 116225, 2019.

[26] S. Harbola and V. Coors, "One dimensional convolutional neural network architectures for wind prediction," Energy Convers Manag, vol. 195, pp. 70–75, 2019.

[27] H. Aprillia, H.-T. Yang, and C.-M. Huang, "Short-term photovoltaic power forecasting using a convolutional neural network–salp swarm algorithm," Energies (Basel), vol. 13, no. 8, p. 1879, 2020.

[28] D. Yu, W. Choi, M. Kim, and L. Liu, "Forecasting day-ahead hourly photovoltaic power generation using convolutional self-attention based long short-term memory," Energies (Basel), vol. 13, no. 15, p. 4017, 2020.

[29] S. Van Buuren and S. Van Buuren, Flexible imputation of missing data, vol. 10. CRC press Boca Raton, FL, 2012.

[30] P. F. Smith, S. Ganesh, and P. Liu, "A comparison of random forest regression and multiple linear regression for prediction in neuroscience," J Neurosci Methods, vol. 220, no. 1, pp. 85–91, 2013.

[31] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," J Comput Syst Sci, vol. 55, no. 1, pp. 119–139, 1997.

[32] V. N. Vapnik, "The nature of statistical learning," Theory, 1995.

[33] V. Vapnik, "Statistical Learning Theory. New York: John Willey & Sons," Inc, 1998.

[34] J. Fan, X. Ma, L. Wu, F. Zhang, X. Yu, and W. Zeng, "Light Gradient Boosting Machine: An efficient soft computing model for estimating daily reference evapotranspiration with local and external meteorological data," Agric Water Manag, vol. 225, p. 105758, 2019.

[35] X. Sun, M. Liu, and Z. Sima, "A novel cryptocurrency price trend forecasting model based on LightGBM," Financ Res Lett, vol. 32, p. 101084, 2020.

[36] M. A. Elaziz, A. Dahou, D. A. Orabi, S. Alshathri, E. M. Soliman, and A. A. Ewees, "A Hybrid Multitask Learning Framework with a Fire Hawk Optimizer for Arabic Fake News Detection," Mathematics, vol. 11, no. 2, pp. 1–15, 2023.