Journal of Engineering and Applied Science

RESEARCH

Open Access

Innovative compressive strength prediction for recycled aggregate/concrete using K-nearest neighbors and meta-heuristic optimization approaches

Min Duan[1*]

*Correspondence:
duanmin1981223@126.com

[1] College of Design, Chongqing College of Finance and Economics, Yongchuan 402160, Chongqing, China

Abstract

This paper presents a groundbreaking method for predicting the compressive strength (F_c) of recycled aggregate concrete (RAC) through the application of K-nearest neighbors (KNN) analysis. The task of designing mixture proportions to achieve the desired F_c can be remarkably intricate, owing to the intricate interplay among the components involved. Machine learning (ML) algorithms have exhibited considerable promise in tackling this complexity effectively. In pursuit of enhanced prediction accuracy, this research introduces a semi-empirical approach that seamlessly integrates strategies, including optimization techniques. This study incorporates two meta-heuristic methods, the Fire Hawk optimizer (FHO) and Runge–Kutta optimization (RUK) to enhance model accuracy. The research results reveal three separate models: KNFH, KNRK, and a single KNN model, each providing valuable insights for precise F_c prediction. Remarkably, the KNFH model stands out as a top performer, boasting an impressive R^2 value of 0.994 and a meager RMSE value of 1.122. These findings not only validate the accuracy and reliability of the KNFH model but also highlight its effectiveness in predicting F_c outcomes. This approach holds great promise for precise F_c forecasting in the construction industry. Integrating meta-heuristic algorithms significantly improves model accuracy, leading to more reliable forecasts with profound implications for construction projects and their outcomes. This research marks a significant advancement in predicting F_c using ML, offering valuable tools for engineers and builders.

Keywords: Compressive strength, Recycled aggregate concrete, K-nearest neighbor, Fire Hawk optimizer, Runge–Kutta optimization

Introduction

Compressive strength (F_c) stands as a pivotal parameter within structural engineering and construction materials. It functions as a fundamental gauge of a material's ability to endure axial loads, those forces that compress or shorten it. More precisely, F_c quantifies the maximum axial stress a material, typically concrete, can withstand without succumbing to failure or collapse [1–4]. This characteristic carries immense significance in the

planning and construction of vital structures like buildings, bridges, dams, and various infrastructure projects. A comprehensive grasp of F_c proves indispensable to engineers and architects, as it holds direct sway over structural integrity and safety. Factors such as the composition of concrete mixes, curing conditions, and environmental factors wield substantial influence over F_c. Consequently, researchers and professionals are pursuing refining their comprehension and predictive methodologies in this realm. Recent years have borne witness to the deployment of advanced techniques such as machine learning, finite element analysis, and non-destructive testing, all aimed at augmenting the precision of F_c predictions. Moreover, the evolution of concrete technology, encompassing the incorporation of supplementary cementitious materials and alternative aggregates, has ushered in an era of more sustainable construction practices. Notably, these innovations have not come at the expense of compressive strength; in many instances, they have improved it [5–7]. In essence, the study of F_c serves as the linchpin in guaranteeing the durability and reliability of civil engineering structures. The ceaseless march of research and innovation in this field redefines the future of construction materials and practices, ensuring that they align with the demands of an ever-expanding construction industry while considering environmental sustainability [8–10].

The relentless expansion of the construction industry necessitates vast quantities of aggregates, primarily employed as one of the primary constituents in concrete production. In stark contrast, the demolition of aging structures begets an abundance of discarded concrete, often occupying precious landfill space, engendering severe environmental concerns such as land depletion. This predicament has spurred the exploration of recycling and repurposing demolished concrete as an eco-friendly alternative to non-renewable virgin aggregates [11–13]. The utilization of recycled concrete aggregate (RCA), derived from the crushing of demolished concrete, has emerged as a promising solution, capable of ameliorating the sustainability of natural resources while mitigating the adverse environmental repercussions associated with the mere disposal of demolished concrete. Nevertheless, it is essential to acknowledge that RCA differs in properties from natural aggregate (NA).

Consequently, the physical and mechanical attributes of RAC crafted from RCA exhibit disparities compared to their natural aggregate concrete (NAC) counterparts. These distinctions chiefly arise due to the higher porosity and water absorption characteristics exhibited by RCA in contrast to NA [14–16]. One pivotal mechanical property in the concrete industry, the elastic modulus, gauges a material's deformation is particularly noteworthy. RAC generally demonstrates a lower elastic modulus value when compared to NAC formed with an equivalent water-to-cement ratio (w/c). Various researchers have proposed equations aimed at correlating the elastic modulus of concrete with other properties, such as F_c. However, it is essential to acknowledge that these equations are primarily rooted in experimental data gathered from NAC, casting doubt upon their applicability to RAC.

A more nuanced approach is indispensable in light of the complex and multifaceted nature of experimental trials, particularly those teeming with a myriad of parameters, some of which exert only marginal influence on outcomes [17]. Computer scientists have responded to this challenge by crafting selection algorithms founded on data-driven models [18]. These algorithms exhibit a remarkable capacity to discern

the most pivotal independent variables, promptly trimming the dimensionality of the input matrix and, in turn, enhancing efficiency. The domain of engineering components, systems, and materials is experiencing an escalating demand for soft computing tools in predictive modeling. This upward trajectory underscores the continued prominence of machine learning (ML) models, particularly artificial neural networks (ANNs), lauded for their adeptness in generating precise outcome predictions closely mirroring empirical observations [19–21]. In an era marked by the relentless march of technology, these data-driven tools are revolutionizing our capacity to predict the F_c of RAC, providing invaluable insights into the behavior of this environmentally conscious construction material.

This study is dedicated to refining the accuracy of predictions concerning the F_c development in RAC by improving the K-nearest neighbors (KNN) model. However, realizing the full predictive potential of KNN necessitates the meticulous optimization of its parameters. To tackle this challenge, the study integrates two competent optimization algorithms: the Fire Hawk optimizer (FHO) and Runge–Kutta optimization (RUK). This amalgamation aims to amplify the efficiency of processes associated with both the design and construction of F_c in RAC, ultimately conferring benefits upon the infrastructure sector and the constructed environment. To validate the robustness of the proposed framework, an extensive dataset about F_c is employed. A comprehensive comparative analysis is meticulously conducted to establish its superiority over conventional optimization methods. Esteemed statistical metrics, including R^2, RMSE, and MSE, are harnessed with precision to assess the performance of the ML models incorporated in this research.

Methods

Data gathering

The study thoroughly investigated the compressive strength (F_c) of recycled aggregate concrete (RAC) while considering multiple variables. To enhance the efficiency of analysis, the dataset was meticulously partitioned into three distinct subsets: a training set (70%), a validation set (15%), and a testing set (15%). The study made use of Table 1's thorough analysis of input variables crucial to concrete production to predict F_c behavior using a KNN model. Understanding and controlling the final concrete product's quality relies heavily on these factors. Four hundred forty-one

Table 1 The statistic properties of the input variable of F_C

Variables		Indicators			
	Category	Min	Max	Avg	St. Dev
w/c	Input	0.30	1.03	0.55	0.15
CA/C	Input	1.00	7.40	3.32	1.21
r	Input	0.00	1.00	0.52	0.39
F_A/T_A	Input	0.00	0.58	0.40	0.07
SG(SSD)	Input	0.00	6.23	2.28	0.76
Wa	Input	0.00	28.00	3.71	3.06
F_c(Mpa)	Output	8.00	82.57	40.72	15.34

observations make up the dataset used in this study, ensuring robust statistical properties. The study provides a thorough explanation of each variable below:

1. Water-to-cement ratio (w/c)

This variable represents the proportion of water to cement in the concrete mix, ranging from 0.30 to 1.03. It has an average of 0.55 and a standard deviation of 0.15. A lower value indicates reduced water content, typically leading to stronger concrete.

2. Coarse aggregate to cement ratio (CA/C)

Denoting the ratio of coarse aggregates to cement spans from 1.00 to 7.40, with an average of 3.32 and a standard deviation of 1.21. CA/C significantly influences the structural properties of concrete.

3. Cement fineness (r)

This variable measures the fineness of cement particles, with values ranging from 0.00 to 1.00. The average is 0.52, with a standard deviation of 0.39. Finer cement particles can enhance both workability and strength.

4. Fine aggregate to total aggregate ratio (FA/TA)

The ratio of fine aggregates to total aggregates varies from 0.00 to 0.58, with an average of 0.40 and a standard deviation of 0.07. This ratio significantly impacts concrete workability and long-term durability.

5. Specific gravity of saturated surface-dry aggregates (SG)

The specific gravity of saturated surface-dry aggregates ranges from 0.00 to 6.23. The average is 2.28, with a standard deviation of 0.76, reflecting aggregate density.

6. Water absorption of aggregates (Wa)

This variable represents the water absorption capacity of aggregates, with values spanning from 0.00 to 28.00. The average is 3.71, with a standard deviation of 3.06. Lower water absorption is desirable for concrete quality.

Based on a dataset of 441 observations, this in-depth analysis of these variables offers vital insights for optimizing concrete mix designs to achieve desired strength and performance characteristics. The statistical properties provide invaluable information for quality assurance and determining variability in concrete production [22]. Marginal histograms, which are visual representations of the distributions of specific variables along the edges of a scatter plot or two-dimensional graph, are shown in Fig. 1. They give a quick overview of the distribution of the data, making it easier to spot trends, outliers, and patterns within each variable while also visualizing how those variables relate to one another.

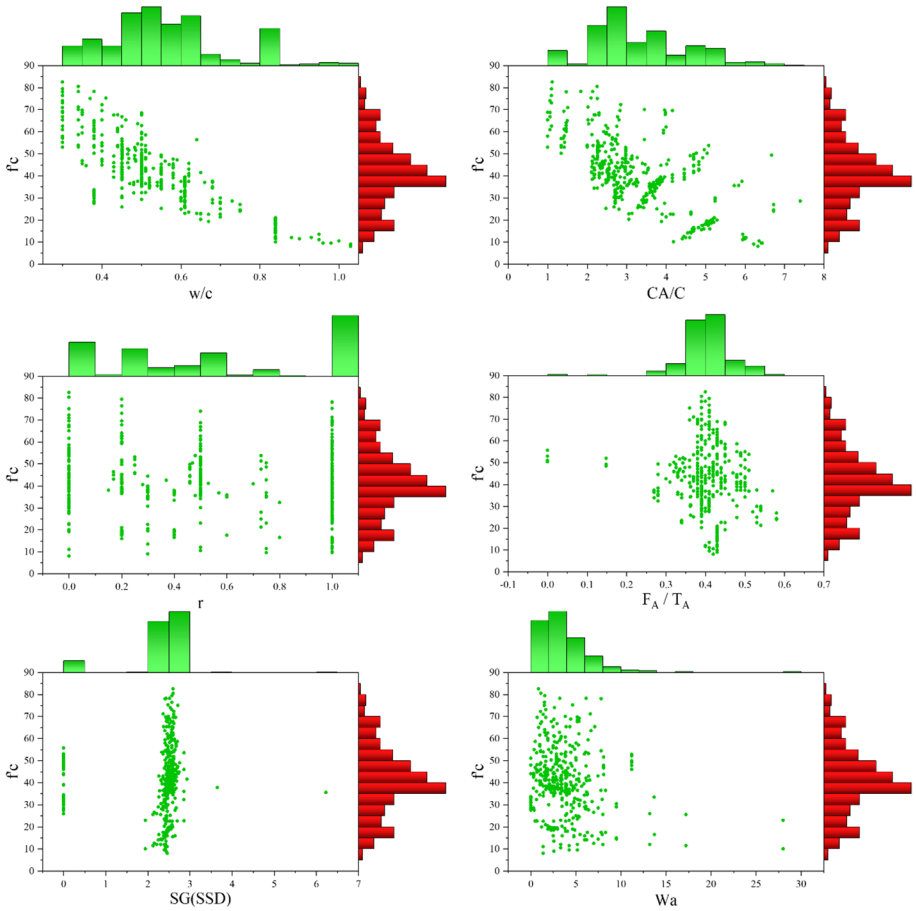

Fig. 1 The marginal histograms plot for input and output

K-nearest neighbor's (KNN)-based

The *KNN* algorithm makes predictions based on the most frequently occurring feedback from K data points nearest the test point. Before applying the algorithm, it is essential to address the normalization of these parameters using Eq. (1).

$$x_{normalization} = \frac{x - Min}{Max - Min} \tag{1}$$

Afterward, utilize Eq. (2) to compute the Euclidean distance between the test and data points.

$$H\left(x_i, x_j\right) = \left(\sum_{h=1}^{m} \left|x_i^{(h)} - x_j^{(h)}\right|^2\right)^{\frac{1}{2}} \tag{2}$$

Equation (2) calculates the distance H between the original data points (x_i) and the test point (x_j) using Euclidean distance, where m is the number of argument points [23]. However, since different parameters have varying impacts on thermal comfort even when the exact value is changed, such as a $1°C$ change in air temperature has a more significant impact than a 1% change in air humidity to remove the inconsistent

effects of indoor thermal parameters on thermal comfort, it is necessary to modify the Euclidean distance for all parameters using Eq. (3).

$$H(x_i, x_j) = \left(\sum_{h=1}^{m} \left(w_h * \left| x_i^{(h)} - x_j^{(h)} \right|^2 \right) \right)^{\frac{1}{2}} \tag{3}$$

The weight (w_h) assigned to each indoor thermal parameter that impacts thermal comfort [24]. Distances are calculated to determine the K data points closest to the test point [25]. The feedback from the subjects at the current test point is then taken to be the feedback that occurs the most frequently among these K data points. Cross-validation can be used to determine the value of K, which establishes the quantity of necessary data points. It is crucial to pick a K value that is in the middle between the two extremes. The model may be overly sensitive to sample points close to the test point if K is too small, leading to an excessive amount of interference from noise points. On the other hand, if K is too high, the model's accuracy might suffer. The flowchart of *KNN* has been shown in Fig. 2.

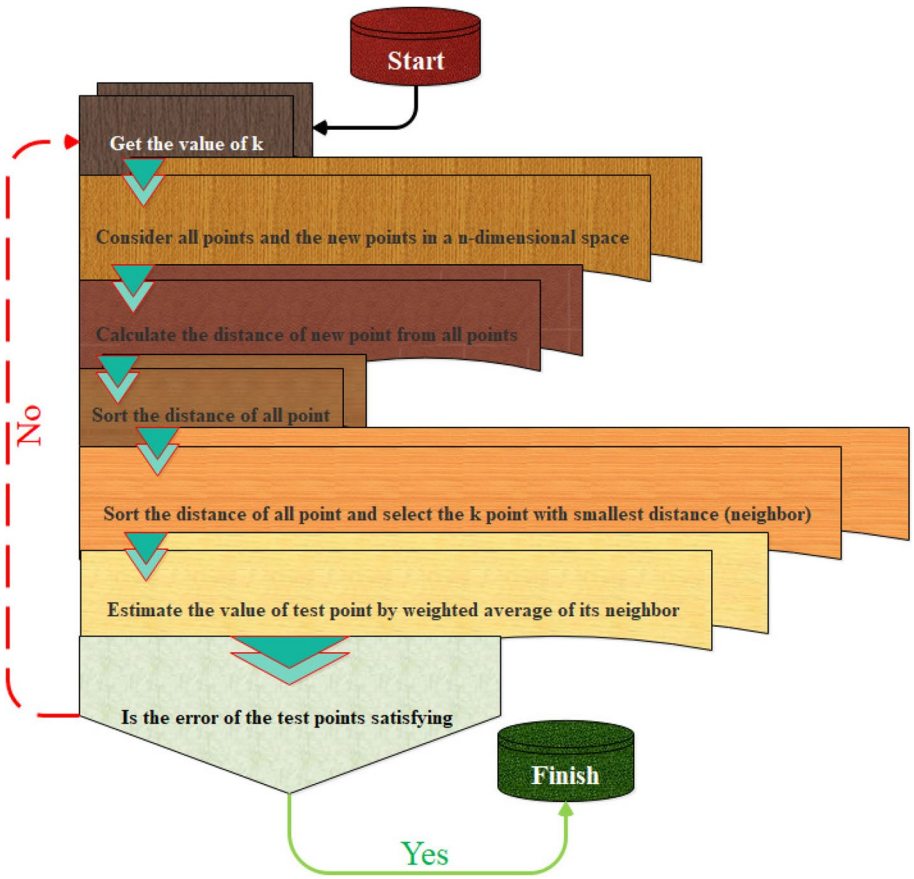

Fig. 2 The flowchart of the KNN mode

Fire Hawk optimizer (FHO)

The *FHO* steps are introduced in this section. The starting population X of *FHO* is given a value, and it has N solutions with D values [26]. This procedure is shown as in Eq. (4).

$$X_{ij} = rand \times \left(U_j - L_j\right) + L_j, j = 1, 2, \ldots, D \tag{4}$$

U_j and L_j are utilized in Eq. (4) to represent the search domain's boundaries at dimension j. A random value is indicated by rand $<$ *spanclass* $='$ *reftype*$'$ $>$ $[0, 1]$ $<$ /*span* $>$. Each solution X_i then calculates its fitness value and the best one (X_b) as having the highest fitness value. The best n solutions are then used to construct the fire Hawks ($FH_{l,l} = 1, 2, ..., n$), while the rest refer to the prey ($PR_{k,k} = 1, 2, ..., m$). The distance between *FH* and *PR* is then calculated as follows:

$$D_{lk} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, l = 1, 2, \ldots, n, k = 1, 2, \ldots, m \tag{5}$$

The following equation will then be used to modify the value of *FK*.

$$FH_l(t + 1) = FH_l(t) + (r_1 \times X_b - r_2 \times FH_n(t)), l = 1, 2, \ldots, n \tag{6}$$

where there is one Fire Hawk, $FH_n(t)$. r_1 and r_2 are random values found in the range $<$ *spanclass* $='$ *reftype*$'$ $>$ $[0, 1]$ $<$ /*span* $>$. The safe prey area is then allocated, and this is shown using the formula below to find the safe position (SP_l) inside the Fire Hawk region [27].

$$SP_l = \frac{\sum_{q=1}^{r} PR_q}{r}, q = 1, 2, \ldots, r, l = 1, 2, \ldots, n \tag{7}$$

The next step involves simulating animal behavior via *PK* movement within the *FH* zone. This simulation updates the prey's position as follows:

$$PR_q(t + 1) = PR_q(t) + (r_3 \times FH_l - r_4 \times SP_l(t)), l = 1, 2, \ldots, n, q = 1, 2, \ldots, r \tag{8}$$

After that, the following formula updates the safe location outside the l th *FH*.

$$SP = \frac{\sum_{k=1}^{m} PR_k}{r}, k = 1, 2, \ldots, m \tag{9}$$

The prey then changes its location based on the calculation below.

$$PR_q(t + 1) = PR_q(t) + (r_5 \times FH_a - r_6 \times SP(t)), l = 1, 2, \ldots, n, q = 1, 2, \ldots, r \tag{10}$$

The stop criteria are then checked to see if they have been satisfied. If they have, the best solution is the output of *FHO*; otherwise, the updating process is repeated [28].

Runge–Kutta optimization (RUN)

The *RUN* optimization algorithm is based on the Runge–Kutta method (*RKM*), which was employed to compute solutions associated with differential equations of the first order. The *RUN* algorithm's mathematical formulation comprises a series of stages, which are elaborated upon below:

- The initialization stage involves creating the initial solutions for N agents based on the search space's boundaries $[LB, UB]$. This is accomplished by employing the subsequent Eq. (11):

$$\begin{aligned} Z_{ij} &= LB_j + r_1 \times \left(UB_j - LB_j\right) \\ i &= 1, 2 \ldots, N, j = 1, 2, \ldots, P \end{aligned} \tag{11}$$

The formula takes into account the dimension of the problem, denoted by P, LB_j, and UB_j signify the lower and upper limits of the jth variable in the solution set Z_{ij}, where i ranges from $1 to N$, representing the overall quantity of search agents [29].

- During the solution refinement stage, the *RUN* algorithm employs a search mechanism (*SM*) that utilizes the *RKM* to modify the current solution's position at every iteration [30, 31]. This mechanism is expressed as follows:

$$Z_i = \begin{cases} Z_{CF} + S_{FM} + \mu \times randn \times Z_{mc}, if rand \leq 0.5 \\ Z_{mF} + S_{FM} + \mu \times randn \times Z_{ra}, otherwise \end{cases} \tag{12}$$

In Eq. (11), $Z_{CF} = (Z_c + r \times SF \times g \times Z_c)$ and $S_{FM} = SF \times SM$. $Z_{ra} = (Z_{r1} - Z_{r2})$, $Z_{MF} = \left(Z_M + r \times SF \times g \times Z_m\right)$, and $Z_{MC} = (Z_m - Z_c)$. The integer value r, which lies between -1 and 1, is utilized to alter the direction of the search process. On the other hand, the symbols g and μ are random numbers ranging from $0 to 2$ and $0 to 1$, respectively. The adaptive factor SF is specified as follows:

$$\begin{aligned} SF &= 2 \times (0.5 - rand) \times f \\ f &= a \times \exp\left(-b \times rand \times \left(\tfrac{t}{tmax}\right)\right) \end{aligned} \tag{13}$$

The total number of iterations is represented by $tmax$. The values of Z_c and Z_m used in Eq. (14) are defined as follows:

$$Z_c = \varphi \times Z_i + (1 - \varphi) \times Z_{r1} \tag{14}$$

$$Z_m = \varphi \times Z_b + (1 - \varphi) \times Z_{pb} \tag{15}$$

Equation (15) includes a randomly generated number represented by the φ, which lies between 0 and 1. Here, Z_b and Z_{pb} denote the best agent at each iteration and the $best - so - far$ agent, respectively. The SM parameter mentioned in Eq. (11) is updated using the following formula:

$$\begin{gathered}
\mathrm{SM} = \frac{1}{6}(Z_{RK})\Delta Z; \\
Z_{RK} = k_1 + 2 \times k_2 + 2 \times k_3 + k_4 \\
\mathrm{k_1} = \frac{1}{2\Delta \mathrm{Z}}(\mathrm{rand} \times \mathrm{Z_w} - \mathrm{u} \times \mathrm{Z_b}) \\
\mathrm{k_2} = \frac{1}{2\Delta \mathrm{Z}}(\mathrm{rand} \times (\mathrm{Z_w} + rand_1 \times \mathrm{k_1} \times \Delta \mathrm{Z}) - \mathrm{UZ} \\
\mathrm{k_3} = \frac{1}{2\Delta \mathrm{Z}}\left(\mathrm{rand} \times (\mathrm{Z_w} + rand_1 \times (\frac{1}{2}\mathrm{k_2}) \times \Delta \mathrm{Z}\right) - \mathrm{UZ}_b \\
\mathrm{k_4} = \frac{1}{2\Delta \mathrm{Z}}(\mathrm{rand} \times (\mathrm{Z_w} + rand_1 \times \mathrm{k_3} \times \Delta \mathrm{Z}) - \mathrm{UZ}_{b2} \\
u = round(1 + rand) \times (1 - rand) \\
UZ = (U \times Z_b + rand_2 \times k_1 \times \Delta Z) \\
UZ_b = \left(U \times Z_b + rand_2 \times (\frac{1}{2}k_2) \times \Delta Z\right) \\
UZ_{b2} = (U \times Z_b + rand_2 \times k_3 \times \Delta Z)
\end{gathered} \tag{16}$$

The symbols $rand_1$ and $rand_2$ represent random numbers. The ΔZ value is calculated as follows:

$$\begin{gathered}
\Delta Z = 2 \times rand \times |Stp|; \\
Stp = rand \times ((Z_b - rand \times Z_{avg}) + y) \\
y = rand(Z_n - rand \times (u - l)) \times \exp(-4 \times \frac{t}{tmax})
\end{gathered} \tag{17}$$

The values of Z_w and Z_b are updated according to the following equations:

if

$$f\ (Zi) < f\ (Zpb)$$

$$Zb = Zi$$

$$Zw = Zpb$$

Else

$$Zb = Zpb$$

$$Zw = Zi$$

• During the enhanced solution quality stage, various operators are employed to improve the convergence rate and avoid local optima. The objective is to enhance the quality of solutions, which is achieved through the following process:

$$\begin{gathered}
Z_{new2} = \begin{cases} Z_{new1} + r \times \omega \times \left|(Z_{new1} - Z_{avg}) + randn\right|, if\ \omega < 1 \\ (Z_{new1} \times Z_{avg}) + r \times \omega \times Z_{na} otherwise \end{cases} \\
Z_{na} = \left|(u \times Z_{new1} - Z_{avg}) + randn\right|, c = 5 \times rand \\
\omega = rand(0, 2).\exp\left(-c\left(\frac{t}{tmax}\right)\right), Z_{avg} = \frac{Z_{r1} + Z_{r2} + Z_{r3}}{3}
\end{gathered} \tag{18}$$

$$Z_{new1} = \delta \times Z_{avg} + (1 - \delta) \times Z_b \tag{19}$$

The formula in Eq. (19) involves a random number, which lies between 0 and 1, and an integer number r that can take on the values of $1, 0$, or -1. According to [30], if the fitness value of Z_{new2} is not superior to the fitness value of Z_i, then there is another

opportunity to update the value of Z_i. This can be achieved by utilizing the subsequent Eq. (20):

$$\begin{aligned} Z_{new3} &= (Z_{new2} - r_1 \times Z_{new2}) + SF \times D_Z \\ D_Z &= (r_2 \times Z_{RK} + (v \times Z_b - Z_{new2})) \end{aligned} \tag{20}$$

This equation involves a random value r_1, r_2, and r_3. The value of v is computed as twice the difference of r_3 and 0.5, where r_3 is a random number in the range $< spanclass =' reftype' > [0, 1] < /span >$.

Performance evaluation methods

This study introduces several criteria for evaluating hybrid models according to their correlations and error rates. The evaluation metrics looked at include root mean square error (RMSE), mean absolute relative error (MARE), coefficient correlation (R^2), mean square error (MSE), and U95. The relevant formulas for each of these metrics are given below. An algorithm that achieves a high R^2 value near 1 performs excellently in the three training, validation, and testing phases. In contrast, metrics with lower values, like RMSE and MSE, are preferred because they show that the model has less error.

$$R^2 = \left(\frac{\sum_{i=1}^{M} (p_i - \overline{p}) \left(l_i - \overline{l} \right)}{\sqrt{\left[\sum_{i=1}^{M} (p_i - p)^2 \right] \left[\sum_{i=1}^{M} \left(l_i - \overline{l} \right)^2 \right]}} \right)^2 \tag{21}$$

$$RMSE = \sqrt{\frac{1}{M} \sum_{i=1}^{M} (l_i - p_i)^2} \tag{22}$$

$$MSE = \frac{1}{M} \sum_{i=1}^{M} l_i^2 \tag{23}$$

$$MARE = \frac{1}{M} \sum_{i}^{M} \frac{|l_i - p_i|}{\left| \overline{l} - \overline{p} \right|} \tag{24}$$

$$U_{95} = \frac{1.96}{M} \sqrt{\sum_{i=1}^{M} (l_i - p_i)^2 + \sum_{j=1}^{M} \left(l_i - p_j \right)^2} \tag{25}$$

Equations (21–25) use the variables M to indicate the number of samples, p_i to represent the predicted value, $\overline{p}$ and $\overline{l}$ to denote the mean predicted and measured values, respectively, and l_i to indicate the measured value alternatively.

Results and discussion

Findings and detailed explanation for Table 2

The study employed three distinct models, namely KNN, KNFH, and KNRK, to forecast compressive strength (F_c) of recycled aggregate concrete. These models

Table 2 The result of developed models for KNN

Model	Phase	Index values				
		RMSE	R^2	MSE	U95	MARE
KNN	Train	2.529	0.977	6.394	7.008	0.052
	Validation	3.536	0.959	12.506	9.802	0.075
	Test	3.123	0.970	9.755	8.657	0.080
	All	2.795	0.973	7.812	7.747	0.060
KNFH	Train	1.122	0.994	1.259	3.110	0.028
	Validation	2.123	0.985	4.505	5.843	0.035
	Test	2.254	0.980	5.081	6.247	0.051
	All	1.522	0.990	2.317	4.217	0.032
KNRK	Train	1.780	0.986	3.166	4.930	0.044
	Validation	2.572	0.975	6.617	7.118	0.048
	Test	2.776	0.970	7.706	7.694	0.055
	All	2.089	0.982	4.362	5.787	0.046

underwent comprehensive evaluation across three phases: training, validation, and testing, with careful data partitioning to ensure fairness. The evaluation process incorporated five vital statistical metrics, including R^2, RMSE, MARE, U95, and MSE, to facilitate a detailed comparison of model performance. Table 2 shows the results of the developed models, and the comparison between the models is as follows:

- The primary focus of the evaluation centered on R^2 values, which indicate the extent to which the independent variable explains variations in the dependent variable. Notably, the KNFH model demonstrated exceptional predictive accuracy, achieving a superior R^2 value of 0.994 during training and consistently outperforming the alternative models. In contrast, the KNN model yielded slightly lower R^2 values of 0.977 during training.
- Furthermore, an in-depth analysis of other error indicators, particularly RMSE, revealed a range spanning from 1.122 to 2.529. Impressively, the KNFH model exhibited the lowest error, while the KNN model exhibited relatively higher errors.
- During the training phase, the KNFH model displayed the lowest MARE value of 0.028, suggesting its superiority. In contrast, the KNN and KNRK models exhibited higher MARE values of 0.052 and 0.044, respectively.
- In terms of MSE and U95 during training, the KNFH model also produced the lowest values, with an MSE of 1.259 and a U95 of 3.110. Interestingly, in the training phase, the MSE and U95 values for the KNN model were the highest.

The study's findings undeniably demonstrated that the KNFH model outperformed the KNN and KNRK models in specific phases. However, when selecting a model for real-world applications, it is vital to consider additional factors such as model complexity, computational efficiency, and ease of implementation. In conclusion, the results provide compelling evidence that FHO optimization successfully enhanced the KNN model's predictive capabilities in predicting F_c.

Enhanced presentation of figures in the results section

Figure 3 displays a scatter plot that evaluates the performance of hybrid models during three stages: training, validation, and testing. The evaluation is based on two crucial criteria, R^2 and RMSE. R^2 measures the similarity between predicted and observed values, while RMSE quantifies the prediction error dispersion. The KNFH model's data points were closely grouped around the central line, indicating its outstanding accuracy across all three phases. The tight clustering between predicted and actual values suggests minimal dispersion and a high level of agreement. On the other hand, the KNRK and KNN models had data points that were more evenly spread around the central line, indicating similar performance levels. However, compared to the KNFH model, this broader dispersion suggests a higher error and somewhat lower accuracy in the KNRK and KNN models.

In Fig. 4, there is a line plot that compares projected and observed values of F_c of RAC. This visual representation is divided into three main sections: training, validation, and testing. The accuracy of this representation depends on how closely the projected behavior matches the observed behavior. The KNFH model predicts values slightly higher than actual measurements, causing slight differences in performance between the three phases. The KNN and KNRK models show minimal deviation between projected

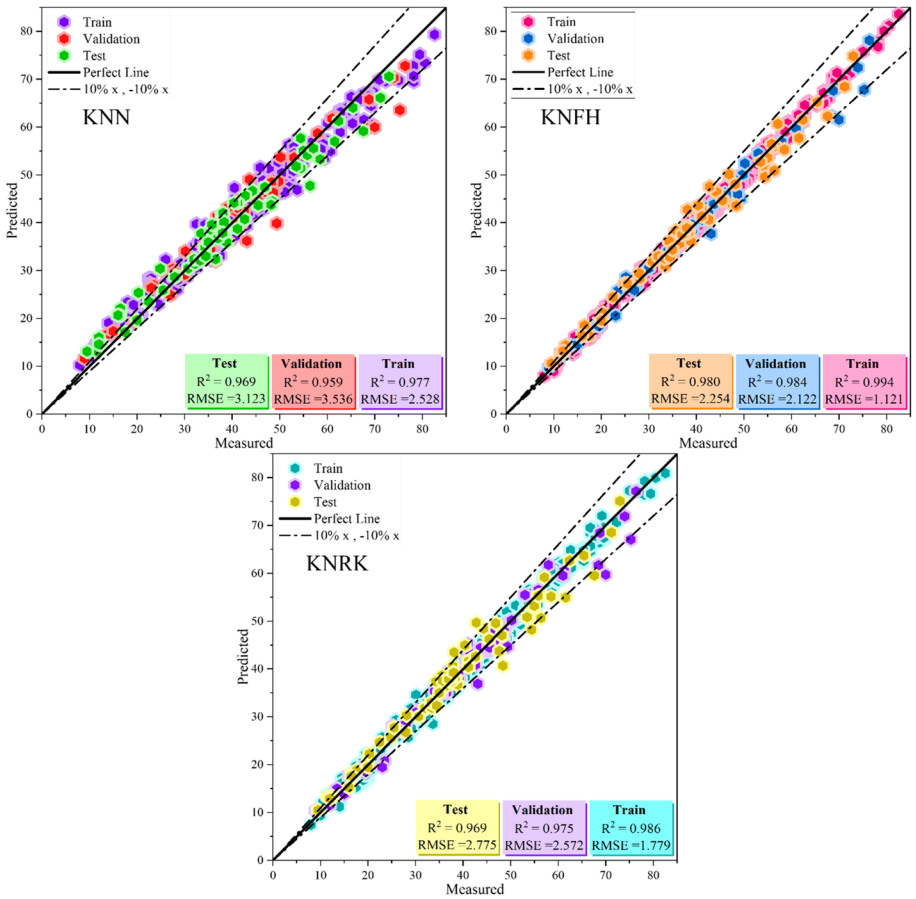

Fig. 3 Plotting the dispersion of evolved hybrid models

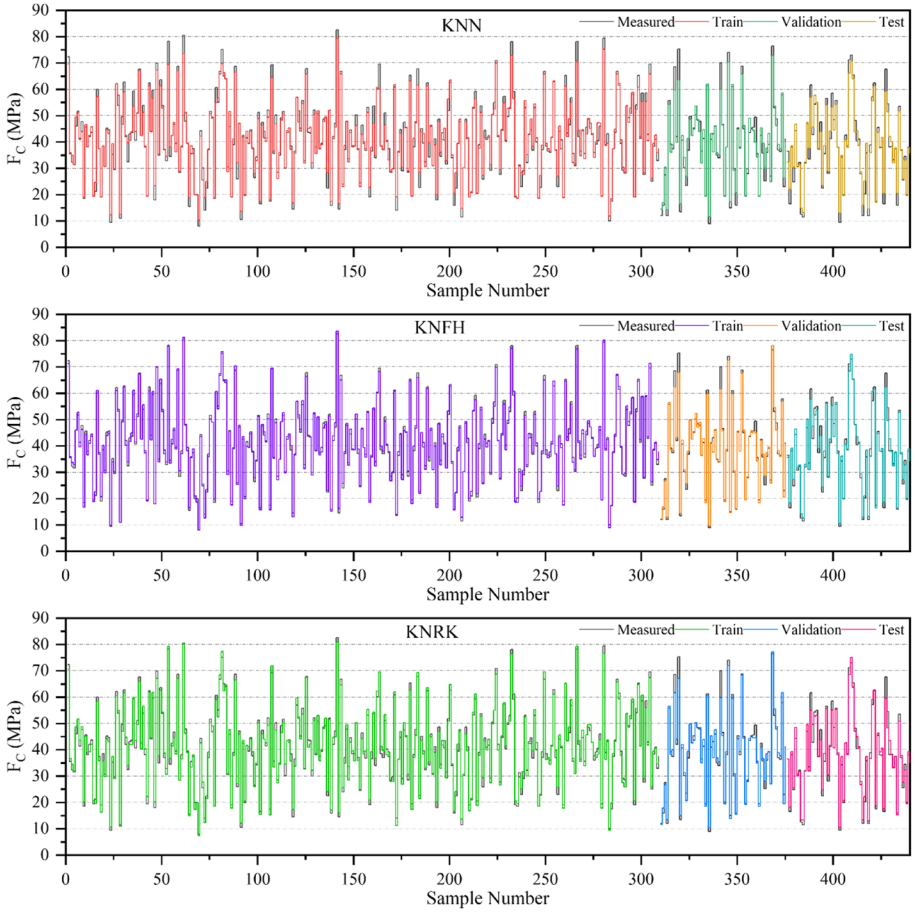

Fig. 4 The comparison of predicted and measured values

and measured points but are less precise than the KNFH model, with a significant gap between projected and measured points.

Figure 5 presents a drop-line plot depicting the error percentages of the models developed in this study. The majority of data points cluster around the 14.96% mark, underscoring KNFH as the model with the lowest error rate. In contrast, both KNN and KNRK exhibit a broader range of error percentages, with a substantial number of values surpassing 37.94% and 19.13%. Notably, the right-skewed distributions of KNN and KNRK highlight data points with significantly higher error percentages. This observation underscores KNFH's superior accuracy and serves as a visual representation of the error percentage distributions for the developed models.

Figure 6 presents a scatter interval plot that effectively illustrates the error percentages associated with the models examined in this study. Notably, KNFH emerges as the top performer, boasting an outstanding mean error rate of 0%. Its error distribution consistently remains below the 10% threshold, and the data displays minimal dispersion, closely resembling a normal distribution curve. In contrast, KNN's performance is characterized by dispersion across all phases. This model exhibits a more symmetrical and uniform normal distribution, with error percentages not exceeding 25%. The behavior of KNRK stands out due to its unique characteristics. This model showcases the most

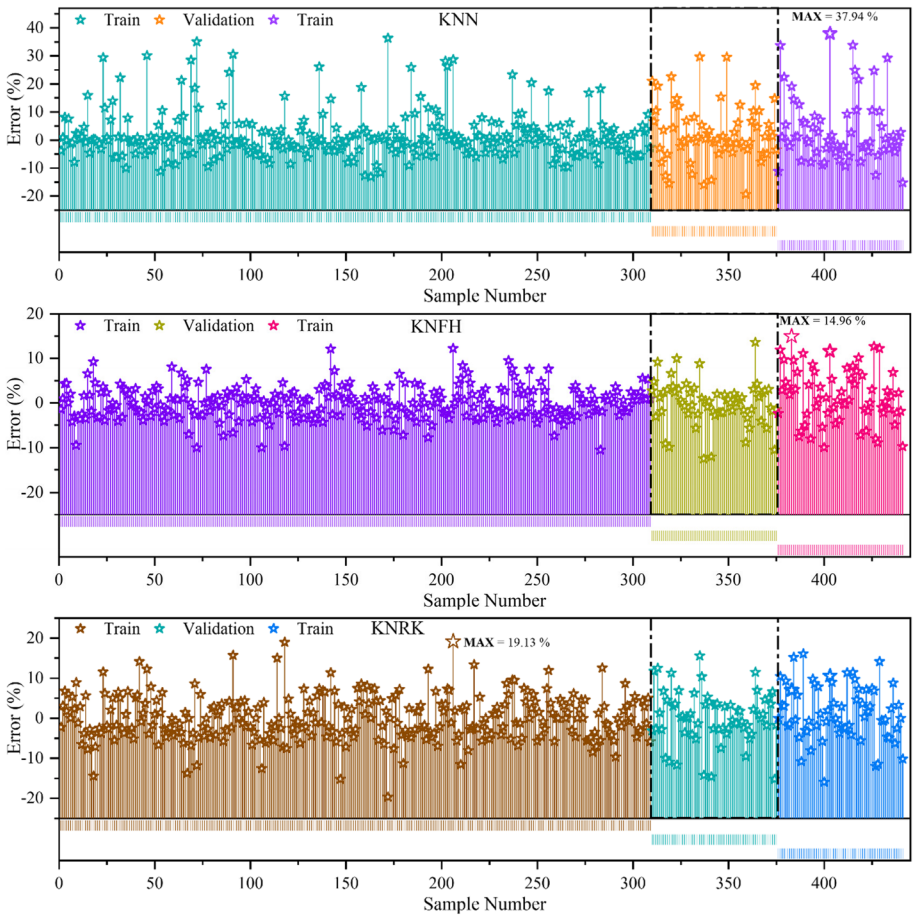

Fig. 5 The error rate percentage for the models is based on the vertical drop line plot

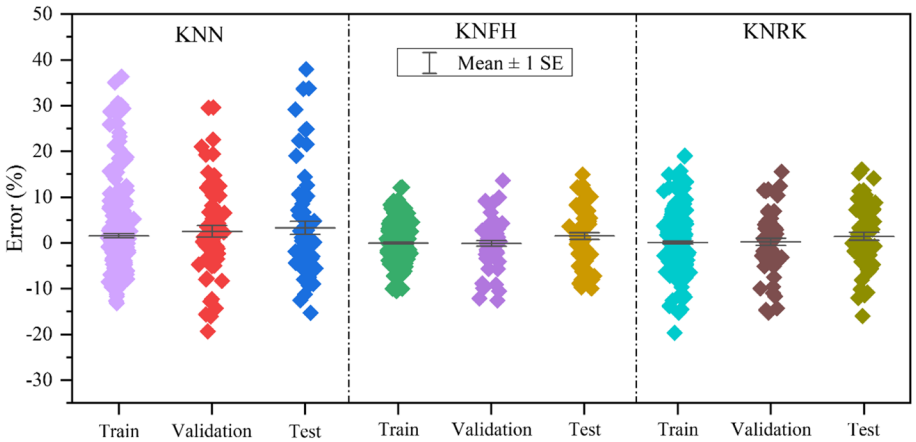

Fig. 6 The scatter interval plot of errors comparison of proposed models

pronounced and diverse discrepancies among the three. Interestingly, a single outlier datum contributes to over 15% of the dataset, an unusual occurrence in statistical analysis. This further emphasizes the distinct nature of KNFH's performance.

Conclusions

Experimental studies aimed at comprehending the distinct properties of compressive strength (F_c) of recycled aggregate concrete (RAC) has significantly increased in recent years. Due to its complex and nonlinear nature, it has been challenging to establish a precise correlation between the composition variables and F_c using conventional statistical methods. The solution to this problem requires a robust and sophisticated methodology that can glean valuable information from the vast amount of experimental data. Such a strategy ought to offer precise estimation methods and perceptions of the complex issues involved in nonlinear materials science. Machine learning (ML), a potent tool capable of revealing hidden patterns within complex datasets, plays a crucial role. With these considerations in mind, this study is dedicated to harnessing the cutting-edge capabilities of ML, particularly the K-nearest neighbors (KNN) model, to predict F_c of RAC. The foundation of this endeavor rests upon a meticulously curated dataset comprising 441 test experiments and 6 input parameters extracted from an extensive compilation of published literature. To enhance the predictive potential of the KNN model, two meta-heuristic algorithms, namely the Fire Hawk optimizer (FHO) and the Runge–Kutta optimization (RUK), have been seamlessly integrated. The effectiveness and predictive prowess of these models in estimating F_c of RAC properties are quantified through a range of performance evaluation metrics, which are elaborated upon in a dedicated section. The following vital outcomes emerge from this comprehensive evaluation:

- Among the proposed models, the KNFH variants demonstrate remarkable outcomes, yielding the highest R^2 values. Although the KNN model had a slightly lower R^2 score, the difference was negligible. Regarding error rates, KNFH outperforms KNN and KNRK, exhibiting a significant 1.7% reduction. The elevated R^2 values and reduced error rates underscore the impressive predictive capabilities of KNFH.
- Notably, the KNFH model consistently displays the lowest RMSE values across all phases, highlighting its remarkable dependability and accuracy in forecasting F_c. KNFH's RMSE is noticeably 77% lower than that of the KNN model, clearly demonstrating the model's improved prediction accuracy.

The findings unequivocally establish KNFH as the superior performer, outshining KNN and earning the top model accolade in this study due to its exceptional performance.

Acknowledgements
I would like to take this opportunity to acknowledge that there are no individuals or organizations that require acknowledgment for their contributions to this work.

Authors' contributions
All authors contributed to the study's conception and design. Data collection, simulation, and analysis were performed by "Min Duan".

Funding
This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Availability of data and materials
Data can be shared upon request.

Declarations

Competing interests
The authors declare no competing interests.

Received: 13 November 2023 Accepted: 19 December 2023
Published online: 10 January 2024

References

1. Shah HA et al (2022) Application of machine learning techniques for predicting compressive, splitting tensile, and flexural strengths of concrete with metakaolin. Materials 15(15):5435. https://doi.org/10.3390/ma15155435
2. Shi H, Xu B, Zhou X (2009) Influence of mineral admixtures on compressive strength, gas permeability and carbonation of high performance concrete. Constr Build Mater 23(5):1980–1985. https://doi.org/10.1016/j.conbuildmat.2008.08.021
3. Morel J-C, Pkla A, Walker P (2007) Compressive strength testing of compressed earth blocks. Constr Build Mater 21(2):303–309
4. Moutassem F, Chidiac SE (2016) Assessment of concrete compressive strength prediction models. KSCE J Civ Eng 20:343–358
5. Ni H-G, Wang J-Z (2000) Prediction of compressive strength of concrete by neural networks. Cem Concr Res 30(8):1245–1250. https://doi.org/10.1016/S0008-8846(00)00345-8
6. Sadowski Ł, Nikoo M, Nikoo M (2018) Concrete compressive strength prediction using the imperialist competitive algorithm. Computers and Concrete, An International Journal 22(4):355–363
7. Nikoo M, Torabian Moghadam F, and Sadowski L (2015) Prediction of concrete compressive strength by evolutionary artificial neural networks, Adv Mat Sci Eng, vol. 2015.
8. Asteris PG, Skentou AD, Bardhan A, Samui P, Pilakoutas K (2021) Predicting concrete compressive strength using hybrid ensembling of surrogate machine learning models. Cem Concr Res 145:106449
9. Duan Z-H, Kou S-C, Poon CS (2013) Prediction of compressive strength of recycled aggregate concrete using artificial neural networks. Constr Build Mater 40:1200–1206. https://doi.org/10.1016/j.conbuildmat.2012.04.063
10. Mousavi SM, Aminian P, Gandomi AH, Alavi AH, Bolandi H (2012) A new predictive model for compressive strength of HPC using gene expression programming. Adv Eng Softw 45(1):105–114
11. Folino P, Xargay H (2014) Recycled aggregate concrete–mechanical behavior under uniaxial and triaxial compression. Constr Build Mater 56:21–31. https://doi.org/10.1016/j.conbuildmat.2014.01.073
12. Shi C, Li Y, Zhang J, Li W, Chong L, Xie Z (2016) Performance enhancement of recycled concrete aggregate–a review. J Clean Prod 112:466–472
13. Wardeh G, Ghorbel E, Gomart H (2015) Mix design and properties of recycled aggregate concretes: applicability of Eurocode 2. Int J Concr Struct Mater 9:1–20
14. Lovato PS, Possan E, Dal Molin DCC, Masuero ÂB, Ribeiro JLD (2012) Modeling of mechanical properties and durability of recycled aggregate concretes. Constr Build Mater 26(1):437–447
15. Duan ZH, Poon CS (2014) Properties of recycled aggregate concrete made with recycled aggregates with different amounts of old adhered mortars. Mater Des 58:19–29. https://doi.org/10.1016/j.matdes.2014.01.044
16. Xu JJ, Zhao XY, Chen ZP, Liu JC, Xue JY, Elchalakani M (2019) Novel prediction models for composite elastic modulus of circular recycled aggregate concrete-filled steel tubes. Thin-Walled Structures 144:106317
17. Zhou ZH (2021) Machine learning. Springer Nature.
18. Wang H, Lei Z, Zhang X, Zhou B (2016) J. Peng, Machine learning basics, Deep learning, pp 98–164
19. Ceryan N, Okkan U, Kesimal A (2013) Prediction of unconfined compressive strength of carbonate rocks using artificial neural networks. Environ Earth Sci 68:807–819
20. Akbulut S, Kalkan E, and Celik S (2003) Artificial neural networks to estimate the shear strength of compacted soil samples, in Int Conf New Dev Soil Mech Geotech Eng pp. 285–290.
21. Sahoo K, Sarkar P, and Robin Davis P (2016) Artificial neural networks for prediction of compressive strength of recycled aggregate concrete.
22. Golafshani EM, Behnood A (2018) Automatic regression methods for formulation of elastic modulus of recycled aggregate concrete. Appl Soft Comput 64:377–400. https://doi.org/10.1016/j.asoc.2017.12.030
23. Xiong L, Yao Y (2021) Study on an adaptive thermal comfort model with K-nearest-neighbors (KNN) algorithm. Build Environ 202:108026
24. Uddin S, Haque I, Lu H, Moni MA, Gide E (2022) Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction. Sci Rep 12(1):6256
25. Abu Alfeilat HA, et al. (2019) Effects of distance measure choice on k-nearest neighbor classifier performance: a review, Big Data 7:221–248.
26. Azizi M, Talatahari S, Gandomi AH (2023) Fire Hawk optimizer: a novel metaheuristic algorithm. Artif Intell Rev 56(1):287–363
27. Shishehgarkhaneh MB, Azizi M, Basiri M, Moehler RC (2022) BIM-based resource tradeoff in project scheduling using fire hawk optimizer (FHO). Buildings 12(9):1472
28. Hosseinzadeh M et al (2023) A cluster-based trusted routing method using fire hawk optimizer (FHO) in wireless sensor networks (WSNs). Sci Rep 13(1):13046
29. Chen H, Ahmadianfar I, Liang G, Bakhsizadeh H, Azad B, Chu X (2022) A successful candidate strategy with Runge-Kutta optimization for multi-hydropower reservoir optimization. Expert Syst Appl 209:118383
30. Ahmadianfar I, Heidari AA, Gandomi AH, Chu X, Chen H (2021) RUN beyond the metaphor: an efficient optimization algorithm based on Runge Kutta method. Expert Syst Appl 181:115079
31. Yousri D et al (2022) Modified interactive algorithm based on Runge Kutta optimizer for photovoltaic modeling: justification under partial shading and varied temperature conditions. IEEE Access 10:20793–20815

Publisher's Note