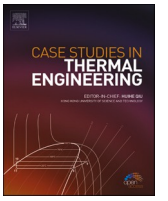
Case Studies in Thermal Engineering 73 (2025) 106505



Contents lists available at ScienceDirect

Case Studies in Thermal Engineering

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





Hybrid computational model for analysis of drug release from biomaterial matrix: Machine learning aided material design

Haoyu Wang [a,b], Zenan Tian [b], Long Wang [a], Haifan Wang [b], Yuxing Zhang [a,*], Wenrui Ban [b,**]

[a] Biomedical Research Center of Xijing University, Xi'an, Shaanxi, 710123, China
[b] Department of Orthopedics, The Second Affiliated Hospital, Xi'an Jiaotong University, Xi'an, Shaanxi, 710004, China

ARTICLE INFO

Keywords:
Drug release
Biomaterials
Gradient boosting decision trees
Deep neural networks
Neural oblivious decision ensembles

ABSTRACT

Prediction of drug release from novel drug delivery systems is of great importance to enhance the efficiency of therapeutics. This study evaluates the predictive capabilities of three machine learning models, Gradient Boosting Decision Trees (GBDT), Deep Neural Networks (DNN), and Neural Oblivious Decision Ensembles (NODE) for modeling complex nonlinear relationships in a dataset containing drug concentration. The data is extracted from mass transfer simulation of drug release from a biomaterial matrix, and the main output is spatial concentration distribution of the drug inside the matrix. Key input features (r and z) ranged from 0.001 to 0.003 m and 0–0.006 m, respectively, predicting the output concentration (C) within 0.00038–0.000831 mol/m^3. The models were optimized using the Stochastic Fractal Search (SFS) algorithm, which systematically adjusted hyperparameters to maximize cross-validation R^2 scores. NODE outperformed other models, achieving R^2 scores of 0.99881 (train), 0.99776 ± 0.00003 (validation), and 0.99829 (test), with minimal error metrics: RMSE of 0.00000344 (train) and 0.00000421 (test). In contrast, GBDT and DNN achieved test R^2 scores of 0.97117 and 0.97165, respectively. NODE's predictions were highly consistent across folds, demonstrating robustness and generalizability. SHAP analysis indicated the dominant influence of input feature r, while z contributed negatively to the output. The NODE model is the most appropriate method for handling complex interactions in tabular data thanks to its unique combination of interpretability, scalability, and high predictive performance.

1. Introduction

Efficient delivery of drugs to the target sites is one of challenges in pharmaceutical area which has significant effect on the treatment efficiency of various diseases such as cancer treatment. The delivery of desired dosage and sustained release of drug to the tumor site is the goal for development of advanced cancer treatment methods based on drug delivery method [1–3]. Controlled release (CR) is an efficient strategy to enhance the therapeutic effects of medicines for patients. CR can be carried out via various techniques such as nanoparticles and porous materials [4]. In this technique, the drug is released from the carrier in a highly controlled manner. The rate of drug release can be controlled via molecular interaction manipulation, mass transfer rate, and other phenomena [5]. In

* Corresponding author.
** Corresponding author.
E-mail addresses: prozhangyuxing@163.com (Y. Zhang), banwenrui@163.com (W. Ban).

https://doi.org/10.1016/j.csite.2025.106505
Received 29 January 2025; Received in revised form 3 May 2025; Accepted 10 June 2025
Available online 10 June 2025
2214-157X/© 2025 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

order to prepare an efficient formulation for CR of medicines, all underlying phenomena should be well understood which can be achieved via modeling the release of drug. Drug release modeling such as mass transfer modeling has been successfully carried out for different formulations, and the results are promising to analyze the rate of drug release from CR formulations [6].

Recently, biomaterial matrices have been used as the host for CR of medicines, due to their biological properties such as biodegradability [4,5,7]. For understanding the release rate, mass transfer was analyzed in the biomaterial matrix to find out the concentration distribution entire the domain. The model based on mass transfer and machine learning (ML) can be explored for this problem which is a new paradigm in evaluation of drug release via hybrid mass transfer-ML models. This strategy has been already evaluated for different processes and showed promising results in terms of accuracy and versatility [8,9]. However, a large dataset is required for training ML models which can be generated via mass transfer simulation of CR system.

Machine Learning (ML) has grown increasingly important as it provides effective techniques for modeling complex data relationships to solve critical problems [10]. Among its advancements, deep learning excels in capturing hierarchical patterns, while tree-based methods like Gradient Boosting Decision Trees (GBDT) offer robust performance in tabular data tasks. This study explores the capabilities of three methods: GBDT, Deep Neural Networks (DNNs), and Neural Oblivious Decision Ensembles (NODE), leveraging their strengths to model nonlinear relationships in data effectively. The models are employed for prediction of dataset derived from drug release from a biomaterial matrix. The ML models are combined with mass transfer simulation of drug release for finding the concentration distribution in the system.

The selection of these methods is motivated by their unique strengths. GBDT is renowned for its reliability and interpretability in structured data problems, DNNs excel in high-dimensional and complex datasets, and NODE bridges the strengths of neural networks and tree-based models, providing scalability and interpretability. These models are particularly suited to capturing the intricate interactions between features in datasets characterized by nonlinear dependencies.

This paper presents a thorough methodology that combines advanced pre-processing techniques, precise hyperparameter tuning using Stochastic Fractal Search (SFS), and robust cross-validation for performance evaluation. Through a systematic comparison of these methods, the study aims to offer valuable insights into their effectiveness and applicability for predictive tasks, thus furthering the use of ML in data modeling for biological applications such as drug delivery.

2. Data analytics

The dataset used in this study contains more than 6000 data points, offering a solid basis for both model training and evaluation. The inputs include the variables r (ranging between 0.001 and 0.003 m) and z (ranging between 0 and 0.006 m), while the output variable C represents the concentration (in mol/m^3) with values spanning from 0.00038 to 0.000831. The methodology is taken from the previous studies on evaluation of drug concentration in biomaterial matrix via mass transfer and machine learning analyses [4,5,7]. Table 1 summarizes the key statistics of the dataset, offering insights into the range, mean, and variability of the input and output parameters.

Computational fluid dynamic methods can be used for simulation of fluid flow and have shown great performance for different geometries and conditions including heat and mass transfer [11,12]. For the mass transfer modeling system in this study, only biomaterial matrix was considered, and the solute concentration variations were obtained by solution of diffusional mass transfer model:

$$\nabla.(-D\nabla C_d) = R$$

where D is drug diffusivity in the biomaterial matrix (m^2/s), C_d is the drug concentration (mol/m^3), and R is the chemical reaction rate (mol/m^3.s). The equation of mass transfer was solved in the process via finite element technique using *COMSOL Multiphysics* package to obtain the concentration distribution in the biomaterial matrix containing drug [5]. The results were then employed for training the ML algorithms. Thus, the data framework for ML modeling is constructed for concentration of drug (C) in the matrix versus coordinates.

Fig. 1 illustrates the pair plot of the dataset, showcasing the relationships and distributions among the variables r, z, and C in the drug release dataset. This visualization highlights the data's structure and any potential trends or correlations that might inform subsequent modeling efforts.

The dataset underwent a thorough pre-processing pipeline to ensure high-quality inputs for modeling. In this study, outlier detection performed based on Isolation Forests. This method isolates outliers by constructing random decision trees and detecting points with short path lengths, which are indicative of potential anomalies [13,14]. By removing these outliers, we ensured the integrity of the dataset and minimized the impact of noisy data on model performance.

After outlier detection, the data was normalized using the Min-Max Scaler. This method normalizes the input features r and z to a

Table 1
The statistics of the variables.

Variable	Minimum	Maximum	Mean	Standard Deviation
r(m)	0.001	0.003	0.001990	0.000727
z(m)	0	0.006	0.003284	0.001794
C(mol/m3)	0.00038	0.000831	0.000707	0.0001

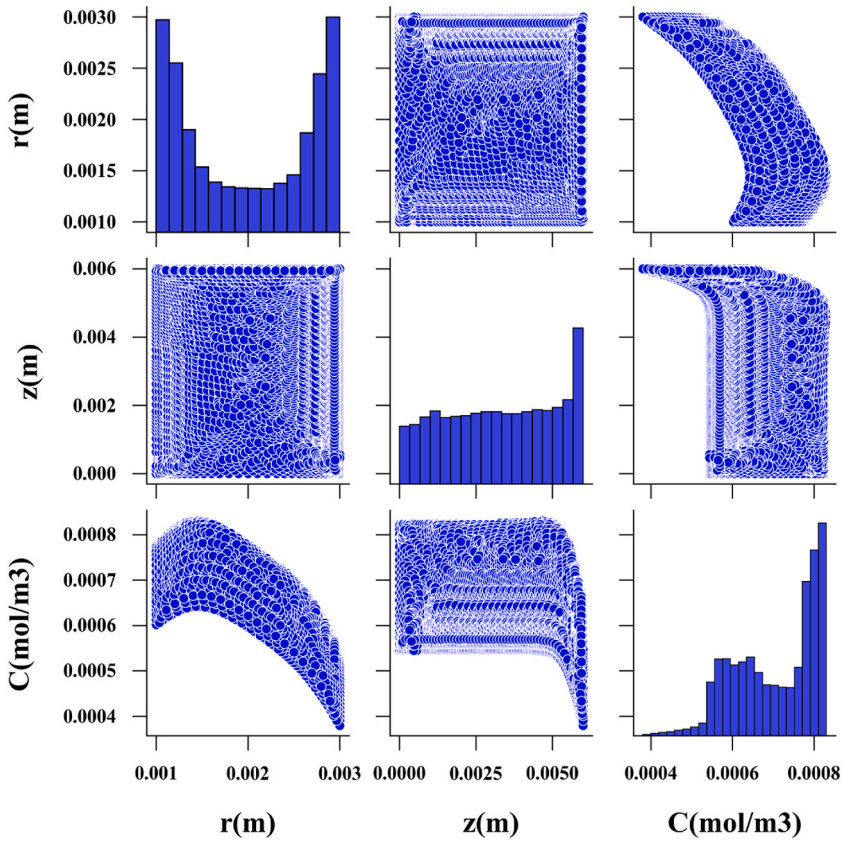


Fig. 1. Pair plot analysis for r, z, and C.

range between 0 and 1, maintaining their relative relationships while ensuring compatibility with ML models sensitive to the input data's scale. Normalization accelerates model convergence and improves prediction accuracy by maintaining consistency across features [5].

3. Methodology

The methodology of this study (Fig. 2) integrates pre-processing, deep and tree-based models, and optimization techniques to build accurate regression models for predicting the output variable C. In the first place, Pre-processing involved using the Isolation Forest for finding the outliers and the Min-Max Scaler for feature normalization.

Three modeling approaches were employed: GBDT, DNN, and NODE. Each method was meticulously optimized to model the intricate non-linear relationships between the input features (r and z) and the output (C). Hyperparameter tuning was conducted using the SFS algorithm, a metaheuristic technique designed for efficient exploration of high-dimensional parameter spaces. This

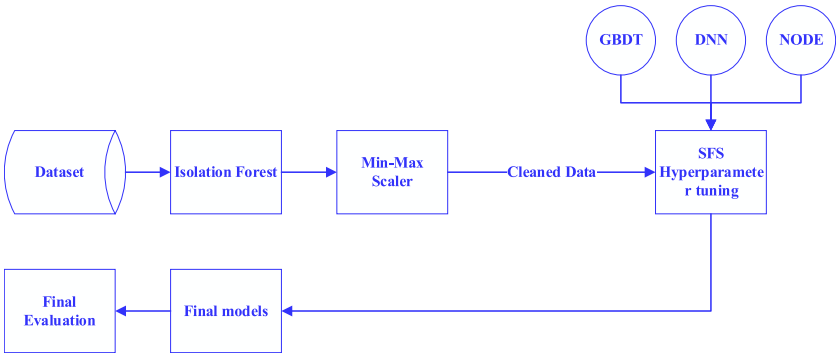


Fig. 2. Overall modeling process designed for prediction of drug release.

comprehensive methodology ensured robust model training and validation, delivering high predictive accuracy across multiple validation folds.

3.1. Gradient Boosting Decision Trees (GBDT)

GBDT is a robust ensemble learning method designed to solve regression problems by iteratively minimizing a specified loss function. GBDT combines the outputs of multiple weak models—shallow decision trees—into a robust predictive model, where each base model (tree) is trained to correct the residual errors of the ensemble so far [15]. During each iteration, a new tree is added, and its predictions are weighted by a learning rate, ν, to control its contribution to the overall model. The updated predictions at each stage are expressed as $F_m(x) = F_{m-1}(x) + \nu \cdot h_m(x)$, where $h_m(x)$ stands for the output of the new tree. This sequential method allows GBDT to reliably capture complex non-linear patterns [16].

The performance of GBDT in regression tasks is highly dependent on the tuning of key hyperparameters. The learning rate (ν) manages the contribution of each tree to the final model, with smaller values requiring more trees but providing greater stability. The number of trees controls the ensemble's size, striking a balance between underfitting and overfitting. Tree depth regulates the complexity of individual trees, capturing higher-order interactions when set appropriately. Subsampling specifies the fraction of the dataset used for training each tree, introducing randomness to enhance generalization. Additional regularization techniques, such as penalizing leaf weights or using L1/L2 regularization, further mitigate overfitting. Hyperparameter optimization, typically performed via grid search or Bayesian methods, ensures the model achieves optimal predictive performance [17].

By leveraging its ability to iteratively refine predictions and model complex feature interactions, GBDT achieves exceptional accuracy in regression tasks, making it a preferred choice for tabular data problems. Advanced implementations such as XGBoost, LightGBM, and CatBoost enhance scalability, computational efficiency, and stability, further solidifying GBDT's position as a cornerstone of regression modeling.

3.2. Deep Neural Networks (DNNs)

DNNs are powerful ML-based models designed to learn relationships in the dataset with great robustness and reliability [18]. For regression tasks, DNNs are configured to predict continuous outputs by adapting their architecture, activation functions, and optimization methods to suit the problem's requirements [19].

A standard DNN for regression has the structure including input, hidden, and output layers in its design for data analytics. The input layer is attributed to the dataset's features, with one neuron for each input feature. The hidden layers are composed of neurons that apply transformations to the data, capturing hierarchical patterns and relationships [20]. These transformations are performed through weighted connections, biases, and non-linear activation functions. The output layer, which produces the final predictions, typically consists of one neuron (for single-output regression).

The transformations within each layer of a DNN can be described mathematically. For a given layer l, $z^{(l)}$ are computed using [21]:

$$z^{(l)} = W^{(l)} a^{(l-1)} + b^{(l)}$$

where the matrix of weights is represented by $W^{(l)}$, while the vector of biases is denoted as $b^{(l)}$. The output activation from the layer before is given by $a^{(l-1)}$. A non-linear transformation is applied using the activation function $g(\cdot)$, producing the following result:

$$a^{(l)} = g\left(z^{(l)}\right)$$

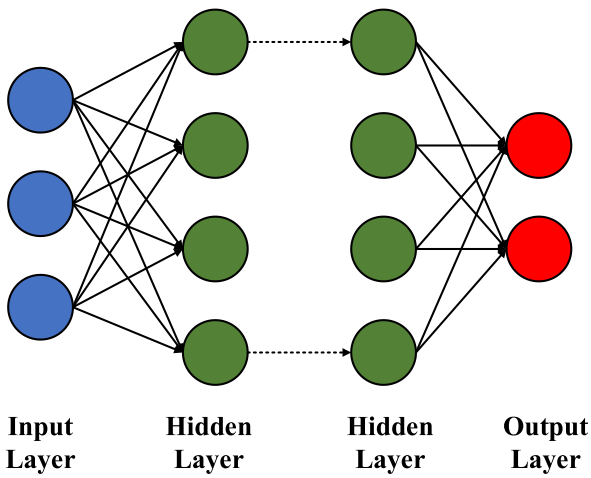


Fig. 3. Deep neural network architecture.

Well-known activation functions include ReLU (Rectified Linear Unit) for its simplicity and efficiency, tanh for smooth transformations, and sigmoid for certain specialized use cases. ReLU is most commonly used in hidden layers due to its ability to handle vanishing gradient issues effectively.

The performance of a DNN in regression tasks depends heavily on the choice of hyperparameters, which can be optimized for better accuracy and generalization. Critical hyperparameters include the network's depth, defined by the number of layers, and its capacity to model complex relationships, determined by the count of neurons within each layer. The learning rate, which regulates the adjustment magnitude during optimization, and the batch size, specifying the dataset size processed in each training iteration, are equally important. Additionally, methods such as dropout, which temporarily disables a subset of neurons during training, are employed to mitigate the risk of overfitting. Proper weight initialization is essential for stable and efficient training, as it impacts how optimization begins.

DNNs are trained through end-to-end processes using gradient-based optimization methods such as stochastic gradient descent (SGD). Employing their capacity to learn non-linear and hierarchical relationships, DNNs perform exceptionally well in regression tasks, particularly with complex and high-dimensional datasets.

Fig. 3 shows the neural network architecture employed in the hybrid computational model. The network consists of an input layer that receives spatial coordinates, one or more hidden layers that learn complex relationships in the data, and an output layer that produces a prediction. Together, these layers form a feedforward neural network that processes the input data through successive transformations to provide accurate predictions.

3.3. Neural Oblivious Decision Ensembles (NODE)

NODE is a deep learning method for tabular data, which combines the decision tree structure with the flexibility of neural networks. NODE generalizes Oblivious Decision Trees (ODTs), which split data based on fixed features and thresholds at each depth, into differentiable models trained end-to-end via backpropagation. This mechanism enables NODE to leverage hierarchical representation learning while retaining the interpretability of tree-based models [22,23].

At its core, a NODE layer comprises mm differentiable ODTs of depth dd, each receiving the same input vector $x \in R^n$. The trees use learnable splitting features $f \in R^d$, thresholds $b \in R^d$, and a response tensor $R \in R^{2^d}$. Differentiability is achieved through the α-entmax transformation, which generalizes softmax to produce sparse probability distributions. Splitting features are computed as [22]:

$$\widehat{f}_i(x) = \sum_{j=1}^{n} x_j \cdot \mathrm{entmax}_\alpha(F_{ij})$$

where $F \in R^{d\times n}$ is a learnable feature selection matrix. Thresholding is softened as:

$$\sigma_\alpha(x) = \mathrm{entmax}_\alpha([x, 0])$$

allowing smooth decision boundaries. The tree's output combines weighted responses:

$$\widehat{h}(x) = \sum_{i_1,\ldots,i_d \in [0,1]^d} R_{i_1,\ldots,i_d} \cdot C_{i_1,\ldots,i_d}(x)$$

where *C(x)* encodes routing probabilities at each split.

NODE adopts a multi-layer structure inspired by DenseNet, where outputs of earlier layers are concatenated as inputs for deeper layers. This design captures both shallow and deep patterns, with final predictions averaged across all trees and layers. Important hyperparameters of this method include the count of layers, quantity of trees per layer, tree depth, α in entmax, and the dimensionality of tree outputs [24].

Training is conducted via stochastic gradient descent with standard losses, such as mean squared error for regression, using data preprocessing like quantile normalization to ensure stability. Initializations for feature selectors and thresholds are data-aware for faster convergence. By integrating tree structures with neural networks, NODE achieves state-of-the-art results on tabular data while offering interpretability, scalability, and adaptability.

3.4. Stochastic Fractal Search (SFS)

For hyperparameter tunning, we employed the SFS algorithm, an advanced metaheuristic method. Inspired by the principles of organic growth in natural environments, it integrates fractal patterns to boost the effectiveness of exploring the solution [25–27]. Below steps outline the procedure of this algorithm:

- Diffusion Step: The SFS algorithm starts with a diffusion phase, utilizing two independent Gaussian walks. A random choice is made between the two walks, governed by a tunable parameter that influences the decision mechanism. The primary Gaussian walk (*GW1)*, initiates from the Best Point (μ_{BP}, BP) determined by the optimization algorithm. The diffusion process commences at point P_i, where two randomly generated numbers ε and ε_0, uniformly distributed in the range [0, 1], impact the trajectory. The expression for *GW1* is [25]:

$$GW_1 = \text{Gaussian}(\mu_{BP}, \sigma) + (\varepsilon \cdot BP - \varepsilon_0 \cdot Pi)$$

The second Gaussian walk, indicated as GW2, starts at the point μ_P and is described by the following equation:

$$GW_2 = \text{Gaussian}(\mu_P, \sigma)$$

To improve localized exploration around promising candidates, the step size σ for both Gaussian walks is modified according to Ref. [28]:

$$\left|\sigma = \frac{\log(g)}{g} \cdot (Pi - BP)\right|$$

- Initialization: Each new particle is checked against boundary constraints for the specified dimension j during the initialization phase. For a specific particle P, with UB representing the upper bounds as a vector, LB indicating the vector of lower bounds, and ϵ being a random number between 0 and 1, this formula is applied [27]:

$$P_j = LB + \varepsilon \cdot (UB - LB)$$

- Linear Ranking Function: SFS uses a linear ranking approach on two separate occasions to determine the probability assigned to each particle, based on how well they perform in terms of fitness. The probability P_{ai} for particle P_i is determined by Ref. [28]:

$$P_{ai} = \frac{\text{rank}(P_i)}{N}$$

- Updating Processes: The first update adjusts each element corresponding to component j and observation i individually. The modified point, represented as P'_i, is obtained after the first update step, with the update defined by the following equation [26]:

$$P'_i(j) = P_r(j) - \varepsilon \cdot (P_t(j) - P_i(j))$$

In the subsequent update phase, P'_i undergoes further modification by integrating two randomly chosen points, P'_t and P'_r. A uniformly distributed random variable, $\widehat{\varepsilon}$, controls the substitution of the vector point. The updated point, denoted as P''_i, is then computed as follows [26]:

$$P''_i = P'_i - \widehat{\varepsilon} \cdot \left|P'_t - BP\right| \quad \text{if} \quad \widehat{\varepsilon_0} \le \frac{1}{2}$$

$$P''_i = P'_i + \widehat{\varepsilon} \cdot \left|P'_t - P'_r\right| \quad \text{if} \quad \widehat{\varepsilon_0} > \frac{1}{2}$$

The optimization process focused on maximizing the mean R^2 obtained through 5-fold cross-validation (CV), ensuring robust evaluation across different data partitions. The SFS algorithm began with an initial sample size of 80, facilitating comprehensive exploration of the parameter space. Key settings included a diffusion rate of 0.6 and an attraction rate of 0.25, which provided a balance between local refinement and global exploration. These configurations enabled the algorithm to systematically adjust hyperparameters, iteratively improving the mean R^2 score from CV. This approach allowed the SFS method to efficiently traverse the search space and enhance model performance without requiring exhaustive computational resources.

4. Results and discussion

The results were obtained using a comprehensive evaluation process that involved a random split of the dataset, with 80 % allocated for training and validation and the remaining 20 % reserved for testing. The validation of the models was conducted through 5-fold cross-validation (CV) to ensure robustness and minimize bias in performance assessment.

The performance metrics for the models, including the R^2 scores in the training, validation, and test phases, are listed in Table 2. Moreover, Table 3 displays the error metrics—specifically the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE)—offering a comprehensive assessment of model accuracy throughout the various phases.

The GBDT model's performance in predicting drug concentration (C) is depicted in Fig. 4 through a scatter plot of predicted versus actual values. The points cluster closely along the diagonal line, reflecting strong predictive accuracy with a test R^2 of 0.97117.

Table 2
R^2 scores of optimized models in Train, Validation, and Test Phases.

Models	Train R^2	CV R^2	Test R^2
GBDT	0.98750	0.98033 ± 0.00549	0.97117
DNN	0.97280	0.96955 ± 0.00137	0.97165
NODE	0.99881	0.99776 ± 0.00003	0.99829

Table 3
The Error Metrics calculated for the optimized models.

Models	RMSE		MAE	
	Train	Test	Train	Test
GBDT	1.11244E-05	1.72763E-05	5.95663E-06	7.38688E-06
DNN	1.70027E-05	1.71292E-05	1.32025E-05	1.31267E-05
NODE	3.43650E-06	4.20774E-06	2.09625E-06	2.40711E-06

However, at higher concentrations (above 0.0007 mol/m^3), the model underpredicts, as points deviate below the line, indicating limitations in capturing highly nonlinear relationships in complex regions of the dataset.

Fig. 5 showcases the DNN model's ability to predict drug concentrations, with a scatter plot comparing predicted and actual values. The points exhibit tighter clustering around the diagonal than the GBDT model, achieving a slightly higher test R^2 of 0.97165. Even at elevated concentrations, deviations are minimal, underscoring DNN's superior capacity to model nonlinear patterns thanks to its multi-layer architecture.

The NODE model's performance is illustrated in Fig. 6, where predicted drug concentrations align almost perfectly with actual values across the entire range. The scatter plot shows minimal deviations, reflected in an impressive test R^2 of 0.99829. This exceptional accuracy highlights NODE's strength in integrating neural network flexibility with decision tree robustness, making it highly effective for complex tabular data.

Among the models evaluated here, NODE demonstrated the best performance, as shown in Tables 2 and 3 and Figs. 4–6.

NODE achieved the highest R^2 scores across all phases: 0.99881 in training, 0.99776 ± 0.00003 in cross-validation (CV), and 0.99829 in testing. These findings highlight NODE's capability to effectively model and forecast intricate nonlinear patterns within the dataset. Additionally, NODE achieved the smallest error measurements, with training and testing RMSE values of 0.00000344 and 0.00000421, respectively, along with MAE values of 0.00000210 for training and 0.00000241 for testing. These metrics underscore the model's accuracy and dependability in predictions.

The minimal variance in the CV R^2 score (±0.00003) underscores NODE's robustness and ability to generalize effectively across different data folds. This consistency highlights the absence of overfitting, which is a significant concern in high-capacity models. NODE's balanced performance across training, validation, and testing phases ensures reliable application in unseen data scenarios.

Visual comparisons of predicted versus actual outputs, as illustrated in Figs. 4–6, further validate NODE's superior performance. The NODE model demonstrated the closest alignment to actual data points, with minimal deviations compared to the GBDT and DNN models.

In conclusion, NODE is found to be the best model in this study, achieving the highest accuracy and generalization robustness. Therefore, it is the most suitable model for modeling and prediction in this research.

Fig. 7 presents the predicted drug concentration (C) as a function of radial coordinate (r) at a fixed axial position (z = 0.003 m). The plot reveals a smooth decline in concentration from the center outward, consistent with a radial gradient driven by diffusive mass transfer. This physically realistic trend validates the model's utility in simulating drug release behavior, aiding the design of controlled release systems.

In Fig. 8, the predicted drug concentration (C) is plotted along the axial coordinate (z) at a fixed radial position (r = 0.002 m). The concentration decreases steadily with increasing z, mirroring an axial gradient typical of diffusive transport. This predictable pattern reinforces the model's relevance and accuracy, offering valuable insights for optimizing drug delivery kinetics.

Figs. 7 and 8 illustrate the effects of the input variables r and z on the predicted concentration C, providing insights into the model's response behavior. Both inputs clearly show an overall negative relationship with output. Fig. 9 presents a contour plot of the predicted

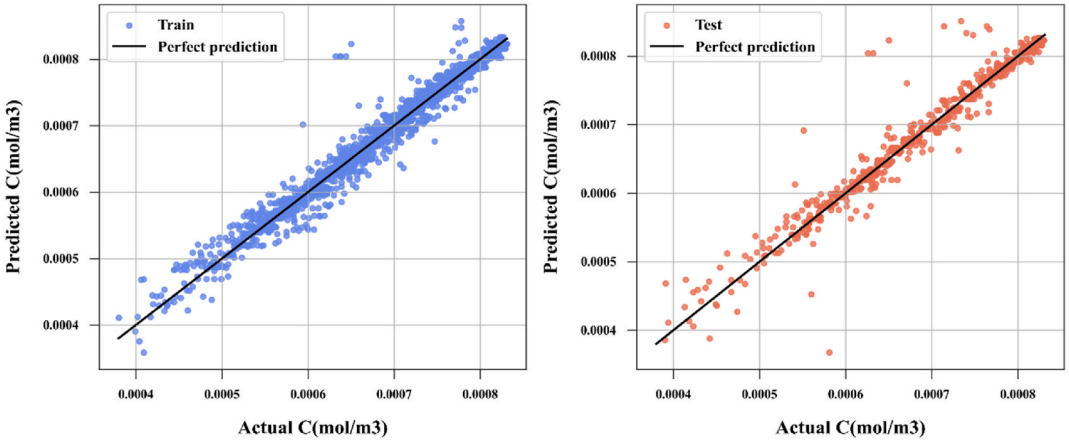


Fig. 4. Comparison of Predicted and Actual outputs (GBDT Model).

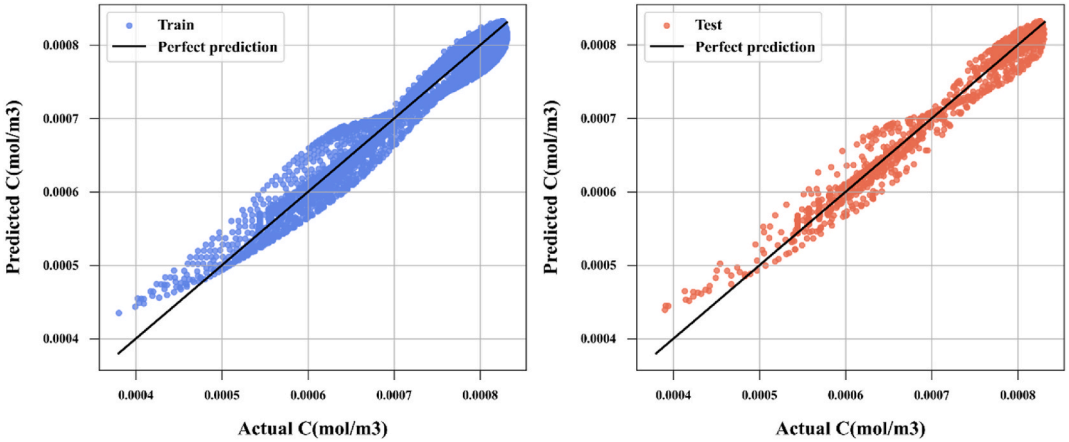


Fig. 5. Comparison of Predicted and Actual outputs (DNN Model).

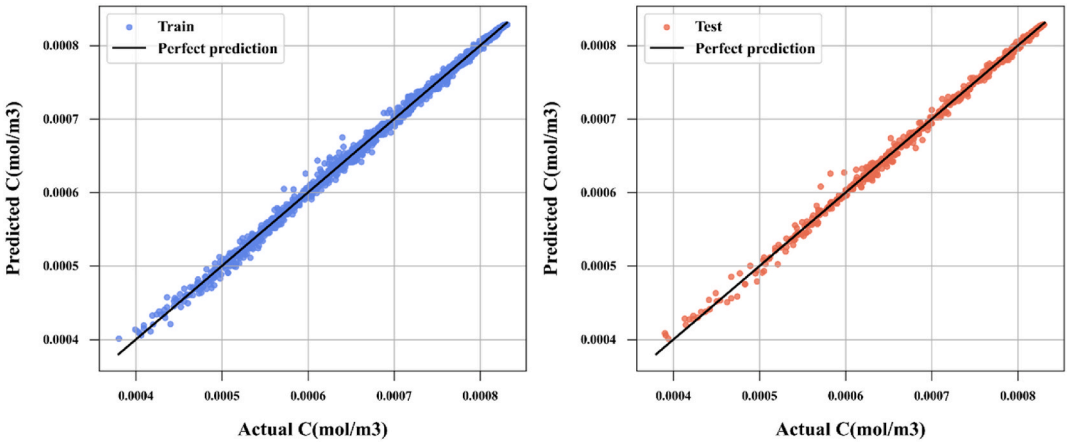


Fig. 6. Comparison of Predicted and Actual outputs (NODE Model).

drug concentration (C) as a function of the radial coordinate (r) and axial coordinate (z), clearly visualizing their combined influence on the concentration distribution within the biomaterial matrix. The plot shows that concentration is highest at the matrix's center (lower r and z values) and decreases steadily toward the boundaries (higher r and z values). As r increases from 0.001 to 0.003 m, the concentration drops significantly, indicating a strong negative relationship. Similarly, as z increases from 0 to 0.006 m, the concentration also declines, though less markedly. This gradient results from diffusive mass transfer, where the drug diffuses outward from the matrix's core to the surrounding tissue, consistent with the dynamics of controlled release systems. The obtained surface concentration plot obtained from mass transfer modeling is shown in Fig. 10 to compare the ML structured concentration distribution with the mass transfer results and great agreement has been obtained which confirms the accuracy of the optimized ML models in prediction of drug concentration in the biomatrix.

SHAP (SHapley Additive exPlanations) is a widely used interpretability framework designed to explain the impact of individual features on a model's predictions. Fig. 11 provides a SHAP summary plot, which visualizes the impact of each input variable on the predicted concentration C. The plot highlights that r has the most significant influence, with higher SHAP values correlating to increased C, while z exhibits a smaller yet consistent negative impact. The spread of points along the horizontal axis represents the variability in feature contributions across the dataset, with denser clusters indicating more frequent impacts. This analysis underscores the model's reliance on r and z in making predictions, offering valuable insights into the underlying relationships and the model's interpretability.

The NODE model developed for predicting drug release showed strong performance, but it has some limitations. The dataset used was based entirely on simulations, which may not fully reflect the complexities of real-world drug release systems. Additionally, while SHAP analysis improved interpretability, the model could still offer clearer insights for designing biomaterials.

For future work, validating the model with experimental data will be crucial to confirm its real-world applicability. Adding features like matrix porosity or drug solubility could boost prediction accuracy. Exploring hybrid models that combine physical constraints with machine learning could also improve both accuracy and interpretability, enhancing the model's potential for practical drug

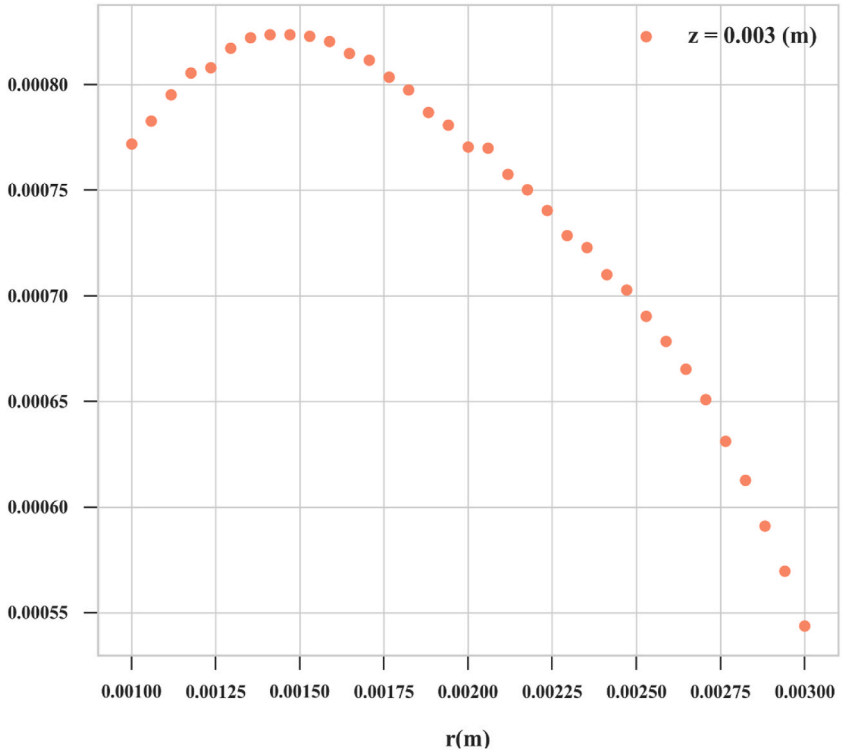


Fig. 7. Radial profile on predicted C (z = 0.003(m)).

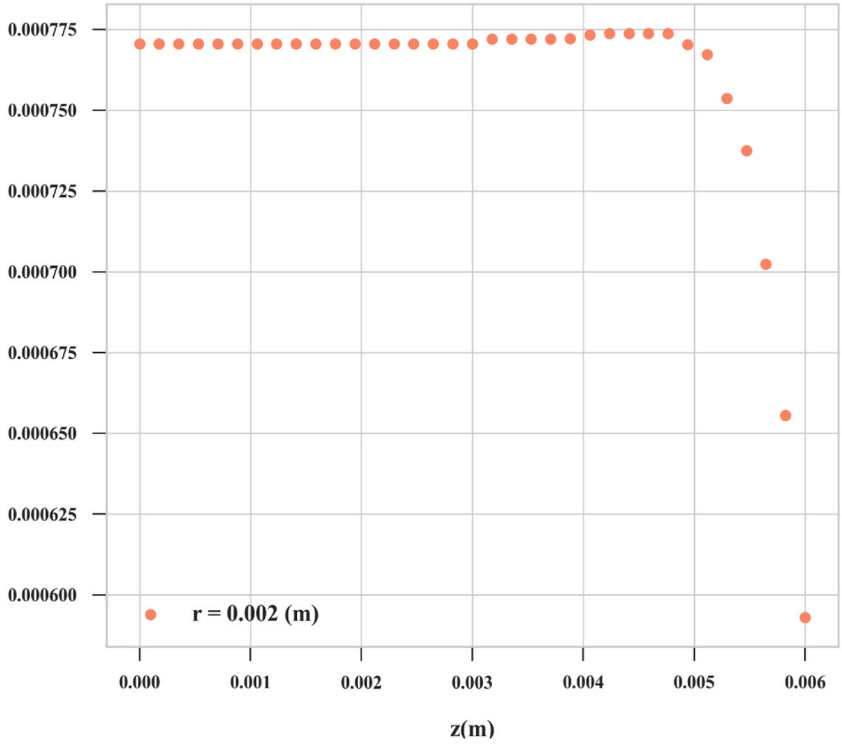


Fig. 8. Axial profile on predicted C (r = 0.002(m)).

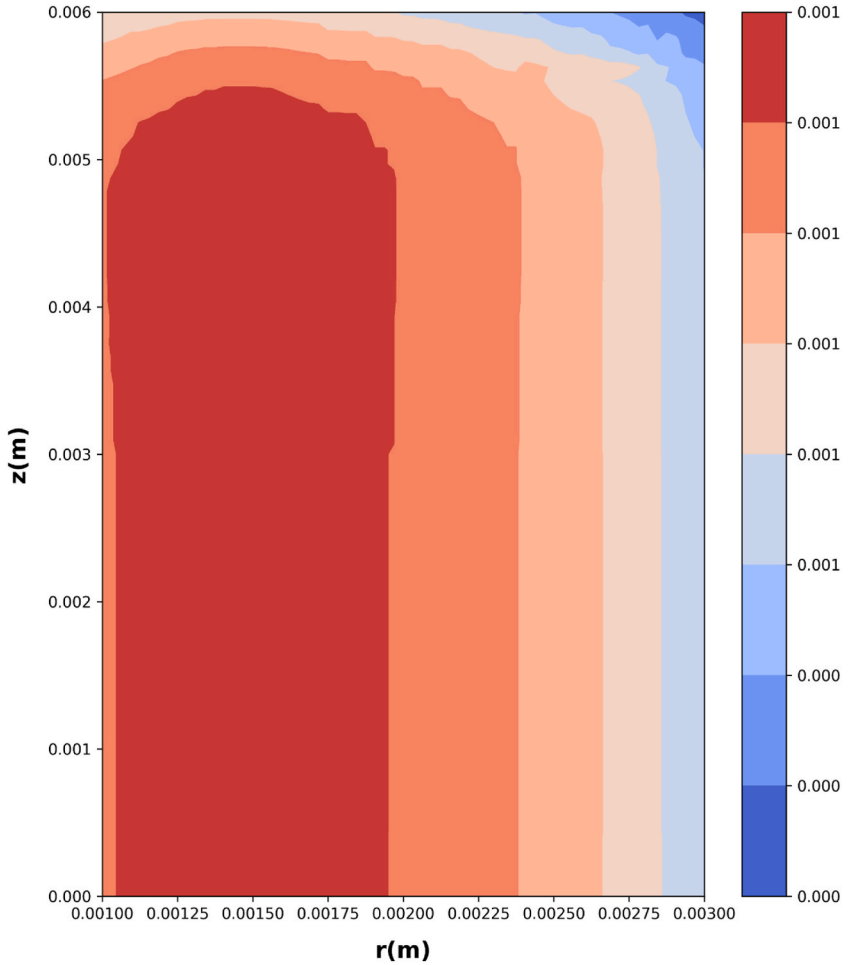


Fig. 9. Contour Plot of predicted C as a function of r and z (ML modeling).

delivery applications.

5. Conclusion

We analyzed the utilization of three ML models—GBDT, DNN, and NODE—in predicting nonlinear relationships in a dataset of over 6000 points. Among the models, NODE demonstrated superior accuracy in fitting, with the highest calculated R^2 (0.99881 for training, 0.99776 $\pm$ 0.00003 for cross-validation, and 0.99829 for testing) and the lowest error metrics (RMSE: 0.00000344 for training and 0.00000421 for testing). NODE's ability to capture intricate interactions between input features (r and z) and the output variable (C) was evident, with minimal overfitting and exceptional generalizability.

Furthermore, SHAP analysis highlighted the significant positive impact of r and the consistent negative effect of z on the predicted output. The visualization of the NODE model's predictions underscored its precision and alignment with the dataset's underlying physical relationships. These results validate NODE as a robust, interpretable, and scalable model for predictive tasks in tabular datasets, outperforming traditional GBDT and DNN approaches.

Overall, integrating advanced pre-processing, systematic hyperparameter tuning using the SFS algorithm, and rigorous evaluation has established NODE as a highly effective method for regression problems, particularly in cases involving nonlinear dependencies. This opens up the possibility for future investigation of NODE and other similar hybrid architectures in domains where high accuracy and reliability are crucial.

CRediT authorship contribution statement

Haoyu Wang: Writing – original draft, Investigation, Methodology, Validation. **Zenan Tian:** Formal analysis, Resources, Software, Writing – original draft. **Long Wang:** Writing – original draft, Methodology, Visualization. **Haifan Wang:** Writing – review & editing, Resources, Conceptualization. **Yuxing Zhang:** Writing – original draft, Visualization, Validation, Supervision, Conceptualization.

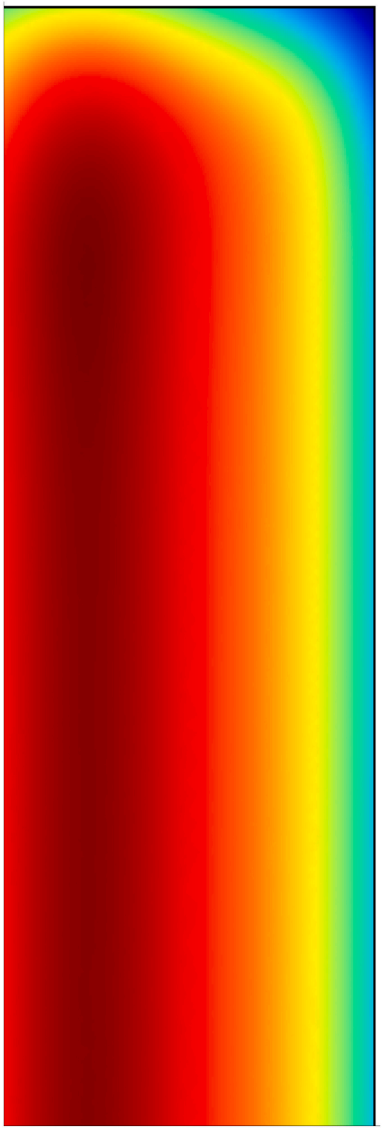


Fig. 10. Drug concentration distribution inside the biomatrix obtained via mass transfer simulation.

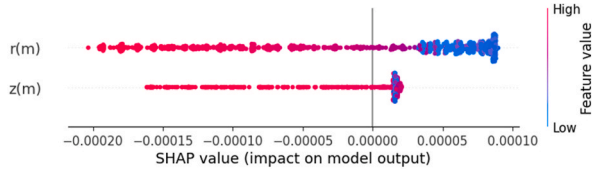


Fig. 11. SHAP summary plot.

Wenrui Ban: Conceptualization, Writing – review & editing, Formal analysis, Validation.

Declaration of competing interest

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

Data availability

Data will be made available on request.

References

[1] L.A. Jadhav, S.K. Mandlik, Nanocarriers in skin cancer treatment: emerging drug delivery approaches and innovations, Nano TransMed 4 (2025) 100068.
[2] Z.S. Sayed, et al., Cancer treatment approaches within the frame of hyperthermia, drug delivery systems, and biosensors: concepts and future potentials, RSC Adv. 14 (53) (2024) 39297–39324.
[3] H. Shao, et al., Polysaccharide-based drug delivery targeted approach for Colon cancer treatment: a comprehensive review, Int. J. Biol. Macromol. (2025) 139177.
[4] S.M. Alshahrani, H.F. Alotaibi, M.Y. Begum, Computational analysis of controlled drug release from porous polymeric carrier with the aid of mass transfer and artificial intelligence modeling, Sci. Rep. 14 (1) (2024) 28422.
[5] M. Alqarni, et al., Model development using hybrid method for prediction of drug release from biomaterial matrix, Chemometr. Intell. Lab. Syst. 253 (2024) 105216.
[6] C. Mircioiu, et al., Mathematical modeling of release kinetics from supramolecular drug delivery systems, Pharmaceutics 11 (3) (2019) 140.
[7] S.M. Alshahrani, H.F. Alotaibi, M. Alqarni, Modeling and validation of drug release kinetics using hybrid method for prediction of drug efficiency and novel formulations, Front. Chem. 12 (2024).
[8] K.Y. Thajudeen, M.M. Ahmed, S.A. Alshehri, Integration of machine learning and CFD for modeling mass transfer in water treatment using membrane separation process, Sci. Rep. 14 (1) (2024) 23970.
[9] Y. Ding, Y. Jin, Development of advanced hybrid computational model for description of molecular separation in liquid phase via polymeric membranes, J. Mol. Liq. 396 (2024) 123999.
[10] I.H. Sarker, Machine learning: Algorithms, real-world applications and research directions, SN Computer Sci. 2 (3) (2021) 160.
[11] N. Abbas, W. Shatanawi, T.A.M. Shatnawi, Thermodynamic analysis of micropolar-casson fluid flow with PST and PHF heating condition over a curved stretching surface, Ain Shams Eng. J. 15 (7) (2024) 102778.
[12] N. Khan, et al., Models based analysis of radiative induced MHD hybrid nanofluid flow over an exponentially stretching sheet, Int. J. Mod. Phys. B 39 (3) (2025) 2550033.
[13] Z. Cheng, C. Zou, J. Dong, Outlier detection using isolation forest and local outlier factor, in: Proceedings of the Conference on Research in Adaptive and Convergent Systems, 2019.
[14] Liu, F.T., K.M. Ting, and Z.-H. Zhou. Isolation forest. in *2008 Eighth Ieee International Conference on Data Mining*. 2008. IEEE..
[15] C. Li, A gentle introduction to gradient boosting, URL: http://www.ccs.neu.edu/home/vip/teach/MLcourse/4_boosting/slides/gradient_boosting.pdf, 2016, 30.
[16] A. Natekin, A. Knoll, Gradient boosting machines, a tutorial, Front. Neurorob. 7 (2013) 21.
[17] P. Prettenhofer, G. Louppe, Gradient boosted regression trees in scikit-learn, in: PyData 2014, 2014.
[18] R.M. Cichy, D. Kaiser, Deep neural networks as scientific models, Trends Cognit. Sci. 23 (4) (2019) 305–317.
[19] C. Krauss, X.A. Do, N. Huck, Deep neural networks, gradient-boosted trees, random forests: statistical arbitrage on the S&P 500, Eur. J. Oper. Res. 259 (2) (2017) 689–702.
[20] W. Liu, et al., A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11–26.
[21] A.L. Caterini, D.E. Chang, Deep Neural Networks in a Mathematical Framework, Springer, 2018.
[22] S. Popov, S. Morozov, A. Babenko, Neural oblivious decision ensembles for deep learning on tabular data, arXiv preprint arXiv:1909.06312 (2019).
[23] J. Luo, S. Xu, NCART: neural classification and regression tree for tabular data, Pattern Recogn. 154 (2024) 110578.
[24] P. Wielopolski, O. Furman, M. Zięba, NodeFlow: towards End-to-End flexible probabilistic regression on tabular data, Entropy 26 (7) (2024) 593.
[25] M. ElKomy, A survey on (stochastic fractal search) algorithm, arXiv preprint arXiv:2102.01503 (2021).
[26] H. Salimi, Stochastic fractal search: a powerful metaheuristic algorithm, Knowl. Base Syst. 75 (2015) 1–18.
[27] M. Alqarni, A. Alqarni, Computational intelligence investigations on the correlation of pharmaceutical solubility in mixtures of binary solvents: effect of composition and temperature, Chin. J. Phys. 93 (2025) 503–514.
[28] S. Aras, E. Gedikli, H.T. Kahraman, A novel stochastic fractal search algorithm with fitness-distance balance for global numerical optimization, Swarm Evol. Comput. 61 (2021) 100821.