

# Machine learning approach in predicting nanofluid viscosity of alumina, copper oxide, silicon dioxide, and titanium dioxide using physics constraint-based XGBoost model

Cite as: J. Appl. Phys. 138, 144701 (2025); doi: 10.1063/5.0240010

Submitted: 24 September 2024 · Accepted: 5 September 2025 ·

Published Online: 9 October 2025



View Online



Export Citation



CrossMark

Shekhar,<sup>1,2,a)</sup> Koj Sambyo,<sup>1,b)</sup> and Seema Tinker<sup>3,c)</sup>

## AFFILIATIONS

<sup>1</sup>Department of Computer Engineering, National Institute of Technology Arunachal Pradesh, Jote, Papum Pare District, Arunachal Pradesh 791113, India

<sup>2</sup>Department of Computer Applications, JECRC University, Jaipur, Rajasthan, India

<sup>3</sup>Department of Mathematics, JECRC University, Jaipur 303905, India

<sup>a)</sup>Author to whom correspondence should be addressed: shekharchndr@gmail.com

<sup>b)</sup>Email: sambyo.koj@gmail.com

<sup>c)</sup>Email: seematinker@gmail.com

## ABSTRACT

Nanofluids are essential colloidal suspensions composed of base fluids with suspended nanoparticles. They possess enhanced thermophysical properties, making them useful for various applications, including heat exchangers, solar collectors, and cooling systems. Although viscosity is a key property that affects heat transfer in nanofluids, it is difficult to predict accurately. Experimental methods provide precise viscosity values but are costly and time-consuming. Therefore, machine learning models have been developed to predict the viscosity of nanofluids more efficiently. The purpose of this research is to develop a Physics-Guided Extreme Gradient Boosting (XGBoost) model (PGXGB) for estimating nanofluid viscosity by incorporating physics-based relationships into the conventional loss functions of XGBoost algorithm. The model is applied to predict the viscosity of water-based  $\text{Al}_2\text{O}_3$ ,  $\text{TiO}_2$ ,  $\text{SiO}_2$ , and  $\text{CuO}$  nanofluids using 792 experimental data points. The results are compared with other machine learning models such as Gradient Boosting Regressor, Extra Tree Regressor, and Nu-Support Vector Regressor. The proposed PGXGB model accurately predicts all data points, demonstrating excellent accuracy and very low prediction error ( $R^2 = 0.992\ 248$ , RMSE = 0.0559 336). Furthermore, statistical and graphical error evaluations demonstrate that the PGXGB model outperforms widely cited empirical, theoretical, and soft computing models in terms of both accuracy and validity range. Sensitivity analysis is also conducted to determine the input factors that most significantly affect prediction accuracy. Among the individual variables, nanoparticle volume fraction is found to have the greatest influence.

© 2025 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/5.0240010>

18 October 2025 14:46:06

## I. INTRODUCTION

Nanofluids are fundamentally colloidal suspensions composed of nanoparticles mixed with base fluids such as ethanol, de-ionized water, or ethylene glycol.<sup>1</sup> The nanoparticles in nanofluids have diameters smaller than 100 nm and can consist of metallic elements, metallic oxides, or non-metallic oxides. The addition of

nano particles to base fluids is known to augment their thermophysical properties, resulting in enhanced energy absorption and increased heat transfer efficiency compared to the base fluids alone. It also significantly alters the thermal properties of base fluids, such as thermal conductivity, dynamic viscosity, and the heat transfer rate. In recent times, nanofluids have gained significant research interest, particularly in the context of heat transfer enhancement

techniques and sustainable, renewable energy systems. The study of nanofluids spans across various domains, including nuclear system cooling,<sup>2</sup> industrial system cooling, hydrodynamics with turbines, solar collectors,<sup>3</sup> hydroelectric rotors, power plant heat exchanger,<sup>4</sup> thermal systems,<sup>5</sup> and boiling processes.<sup>6</sup>

Viscosity is a crucial property of nanofluids that determines their resistance to flow between the adjacent layers of nanofluids. It affects various aspects such as pumping power, pressure drop, and heat transfer coefficient. Predicting nanofluid viscosity is crucial for applications in heat transfer. Experimentally measuring nanofluid viscosity yields the most precise results. However, conducting experiments takes time and is not practical when viscosity values are needed quickly for engineering calculations. This emphasizes the significance of theoretical and empirical models that can accurately estimate nanofluid viscosity.

### A. Theoretical and empirical models of viscosity prediction

The present section provides a concise overview of the existing empirical and theoretical models used in the prediction of nanofluid viscosity. In 1906, Einstein<sup>7</sup> first established the viscosity formula, assuming the viscous fluid contained a very low volume fraction ( $\phi < 0.02$ ) of spherical particles. Numerous theoretical and empirical models have been put forward by distinguished researchers such as Refs. 1 and 7–20. Empirical models are simplified mathematical correlations, but they suffer from limitations. For instance, they are dependent on parameters (e.g., aggregate size or inter-particle spacing between nanoparticles<sup>8,9</sup>) that are either experimentally unavailable or difficult to measure consistently.<sup>10</sup> Empirical models as shown in Table I are usually designed for particular nanofluid types or conditions, which limits their generalizing capability to other scenarios. Furthermore, under challenging situations, such as changing temperatures, nanoparticle concentrations, or different types of nanoparticles, their prediction accuracy tends to decline.

On the other hand, soft computing models do not depend on a priori presumptions about the relationships between parameters.

Rather, they learn patterns directly from data, particularly in combination with domain-specific constraints and generate more accurate and robust predictions. Overcoming the narrow range of empirical models, these soft computing techniques are adaptable to a wide variety of experimental settings including varied nanoparticle kinds (e.g.,  $\text{Al}_2\text{O}_3$ ,  $\text{TiO}_2$ ,  $\text{SiO}_2$ , and  $\text{CuO}$ ), sizes, and volume fractions. The transition from empirical models to soft computing techniques was motivated by specific gaps in the empirical models.

Precisely measuring viscosity is challenging because it depends on many factors. Important experimental results have indicated that both the volume fraction and size of nanoparticles impact nanofluid viscosity.<sup>18,19</sup> Additionally, some research has shown that temperature affects viscosity.<sup>20,21</sup> Mahbubul *et al.*<sup>22</sup> discovered that variables such as temperature, as well as the shape, size, and volume fraction of nanoparticles, all influence nanofluid viscosity. Research indicates that a rise in temperature causes a decrease in dynamic viscosity, whereas an increase in concentration leads to a rise in dynamic viscosity. Other aspects that impact thermophysical qualities like dynamic viscosity include the forms and synthesis procedures of the nanostructures.<sup>23</sup> Despite its importance, measuring viscosity experimentally in the lab can be laborious and challenging. As a result, existing models have limited use for predicting viscosity in common nanofluids such as  $\text{Al}_2\text{O}_3$ ,  $\text{TiO}_2$ ,  $\text{SiO}_2$ , and  $\text{CuO}$  in water. There is a need for more flexible and general models that can accurately estimate nanofluid viscosity based on basic nanoparticle characteristics and fluid properties. Improved predictive abilities will enhance heat transfer applications using these widely used nanofluids.

### B. Soft computing and artificial intelligence (AI) models

Soft computing approaches reveal complex relationships that are often missed by empirical models, and they efficiently manage multidimensional datasets with many variables such as nanoparticle size, base fluid viscosity, and temperature. Experiment-based empirical models rely heavily on resources. Conversely, soft

**TABLE I.** List of the most cited theoretical and empirical models for viscosity prediction.

Theoretical and empirical models	Correlation equation remarks
Pak and Cho <sup>1</sup>	$\mu_{\text{nf}} = \mu_{\text{bf}}(1 + 39.11\phi + 533.9\phi^2)$ .
Einstein <sup>7</sup>	$\frac{\mu_{\text{eff}}}{\mu_{\text{bf}}} = 1 + 2.5\phi \quad \phi < 0.02 \mu_{\text{eff}} = \text{Effective viscosity}$
Brinkman <sup>11</sup>	$\mu_{\text{eff}} = \frac{\mu_{\text{bf}}}{(1-\phi)^{2/3}}$
Ward <sup>12</sup>	$\frac{\mu_{\text{eff}}}{\mu_{\text{bf}}} = [1 + \eta(\phi_{\text{eff}} + 2.5\eta + (2.5\eta)^2 + \dots)]$
Lundgren <sup>13</sup>	$\mu_{\text{eff}} = [1 + 2.5\phi + \frac{25}{4}\phi^2 + f(\phi^3)]\mu_{\text{bf}}$
Batchelor <sup>14</sup>	$\mu_{\text{eff}} = (1 + 2.5\phi + 6.5\phi^2)\mu_{\text{bf}}$
Wang <i>et al.</i> <sup>15</sup>	$\mu_{\text{nf}} = \mu_{\text{bf}}(1 + 7.3\varphi + 123\varphi^2)$
Chen <i>et al.</i> <sup>16</sup>	$\mu_{\text{nf}} = \mu_{\text{bf}}(1 + 10.6\varphi + (10.6\varphi)^2)$
Abedian and Kachanov <sup>17</sup>	$\mu_{\text{nf}} = \frac{\mu_{\text{bf}}}{(1 - 2.5\varphi)}$

computing methods offer quick, reliable predictions across diverse experimental setups. In order to accurately predict the viscosity of certain kinds of nanofluids, artificial intelligence and soft computing approaches have been widely applied in conjunction with experimental studies in recent years. Abdollahi *et al.*<sup>24</sup> presented a novel analysis of heat transfer in hybrid nanofluid flow, incorporating graphene oxide and copper particles in pure water within a rotating system. They utilized the Radial Basis Function (RBF) method for the first time to solve coupled differential equations. The findings reveal a significant reduction in heat transfer as the Reynolds number increases, highlighting the complex interrelationships between key quantities such as Reynolds number, Nusselt number, and nanofluid concentration. Chiniforooshan Esfahani<sup>25</sup> introduced a novel multi-fidelity neural network (MFNN) to predict the viscosity of nanofluids by combining low-fidelity theoretical data with high-fidelity experimental measurements, thereby improving prediction accuracy over conventional machine learning models. This approach integrates physical laws into the model, enhancing its ability to capture the complex rheology of nanofluids in thermal applications. Table II provides a review of some earlier research on the use of soft computing and artificial intelligence to predict the viscosity of nanofluids.

The proposed physics-guided extreme gradient boosting (XGBoost) (PGXGB) model explicitly incorporates viscosity positivity and the inverse temperature–viscosity relationship into its loss function, ensuring that predictions adhere to fundamental physical principles. This novel soft computing approach integrates physics-constrained with the XGBoost algorithm to accurately predict nanofluid viscosity, capturing complex patterns within experimentally gathered datasets. To evaluate the model's performance, we conducted a comprehensive comparison with established machine learning techniques, including Gradient Boosting Regressor (GBR), Extra Tree Regressor (ETR), and Nu-Support Vector Regressor ( $\nu$ -SVR), as well as with widely cited empirical models. Validation is performed using statistical metrics, graphical analysis, and Taylor plots.

Additionally, sensitivity analysis was carried out to identify the most influential input parameters affecting viscosity prediction. To the best of our knowledge, this is the first application of a PGXGB model for predicting the viscosity of various nanofluids. The model achieved exceptional accuracy, with predictive performance exceeding 99.22%, demonstrating its effectiveness across diverse nanoparticle types and experimental conditions.

## II. COMPUTATIONAL METHODOLOGY

The proposed physics-guided XGBoost (PGXGB) model was developed using 792 experimental data points for water-based nanofluids containing  $\text{Al}_2\text{O}_3$ ,  $\text{TiO}_2$ ,  $\text{SiO}_2$ , and  $\text{CuO}$ . The model predicts nanofluid viscosity based on four key input features: temperature, base fluid viscosity, nanoparticle volume fraction, and particle size. Python was used for implementation, and training was conducted on Google Colab, which provided access to graphics processing units (GPUs) (e.g., NVIDIA Tesla T4/K80) and 12 GB RAM. Despite the added complexity from physics-constrained loss functions, model training remained efficient,

completing in approximately 25–30 min depending on hyperparameter configurations. The small variation in training time depended on the different hyperparameter combinations explored during tuning processes like GridSearchCV.

### A. Data collection

The dataset was compiled from 18 peer-reviewed sources,<sup>1,15,20,41–55</sup> ensuring a diverse range of experimental conditions. Nanoparticle sizes ranged from 10 to 150 nm, volume fractions from 0.03%–13.06%, and temperatures from 10 to 72 °C. Water was used as the base fluid, with its viscosity varying between 0.393 and 1.306 mPa s across studies.

This variability arising from differences in particle properties, fluid concentration, temperature, and experimental methodology enhances the model's generalizability but also introduces noise. The PGXGB model addresses this challenge by incorporating physics constrained that guide learning and improve reliability. Statistical summaries of the dataset are provided in Table III. The model achieved high predictive accuracy, with an  $R^2$  of 0.992, demonstrating its robustness across varying conditions.

### 1. Data preprocessing

The data collection and preprocessing steps were crucial to ensuring the reliability of the analysis. Handling missing data was not required, as the dataset has all the values of parameters for 792 data points.

Noise in the dataset was mitigated through several measures. First, the data were sourced from validated experimental studies, reducing the likelihood of noisy inputs. Second, a physics-guided loss function was integrated to enforce domain-specific physical constraints, thereby minimizing the influence of noise in predictions. For example, data that did not follow physical laws, such as negative viscosity values, were mitigated by the viscosity constraint loss function. Additionally, constraints for the common inverse relationship between viscosity and temperature were handled by the temperature–viscosity relationship loss function. Furthermore, standard preprocessing techniques such as normalization or scaling were applied to ensure that differing feature scales did not bias the model training.

### 2. Hyperparameter tuning

The PGXGB model's hyperparameters were tuned using GridSearchCV. For the current research, the learning rate, max depth, number of estimators, subsample ratio, and beta parameter (from temperature–viscosity relationship loss function) were taken into consideration. For every hyperparameter, a range of values were tested using a GridSearchCV technique, which cross-validated combinations to find the best-performing set.

The following range was set for input parameters:

**Learning Rate:** [0.01, 0.05, 0.1, 0.3], **Max Depth:** [3, 6, 9, 12], **Number of Estimators:** [100, 200, 500], **Subsample Ratio:** [0.6, 0.8, 1.0], **Beta Parameter:** [1.0, 0.1, 0.01, 0.001].

Using  $R^2$ , RMSE, or MAPE during cross-validation helped in choosing the best parameters for the model. PGXGB hyperparameter tuning was benefited also from the special physics-guided

TABLE II. Review of earlier research to predict the viscosity of nanofluids using soft computing methods.

Reference	Nanofluid used	Year	Soft computing method	Performance	Result
Karimipour <i>et al.</i> <sup>26</sup>	CuO/liquid paraffin	2018	Neural network	The data's correlational divergence from the experimental values was less than 5%.	Proposed a new correlation for estimating thermal conductivity and dynamic viscosity
Ajila <i>et al.</i> <sup>27</sup>	Iron oxide carbon dioxide	2024	Decision tree, K-nearest neighbors, and linear regression	$R^2 = 0.99$	The decision tree model performed the best in terms of nanofluid transport parameter prediction
Dai <i>et al.</i> <sup>28</sup>	SiO <sub>2</sub> /Ethylene glycol	2023	Gaussian process regression (GPR)	$R^2 = 0.996$ RMSE = 0.24 MAPE = 1.61	Predicted dynamic viscosity and torque
Chiniforooshan Esfahani <i>et al.</i> <sup>25</sup>	Al <sub>2</sub> O <sub>3</sub> , CuO, TiO <sub>2</sub> , SiO <sub>2</sub> , SiC	2023	Multi-fidelity neural network (MFNN)	$R^2 = 0.991$	MFNN surpasses Artificial Neural Network (ANN) for predicting the viscosity of nanofluids
Ahmadi <i>et al.</i> <sup>29</sup>	Al <sub>2</sub> O <sub>3</sub> /water, Al <sub>2</sub> O <sub>3</sub> /ethylene glycol	2018	GMDH artificial neural network	$R^2 = 0.9958$ RMSE = 0.0059	Modeled thermal conductivity
Heidari <i>et al.</i> <sup>30</sup>	Al <sub>2</sub> O <sub>3</sub> , CuO, TiO <sub>2</sub> , SiO <sub>2</sub>	2016	Multilayer Perceptron (MLP)-ANN	AARD = 0.41% ARD = 6.44%	Accurately predicted nanofluid viscosity
Zhao <i>et al.</i> <sup>31</sup>	Al <sub>2</sub> O <sub>3</sub> , CuO	2015	Radial basis function neural networks	Al <sub>2</sub> O <sub>3</sub> RMSE = $9.078 \times 10^{-5}$ , MAPE = 2.673%, SSE = 4.327 $\times 10^{-6}$ , R <sup>2</sup> = 0.9962 CuO RMSE = $3.623 \times 10^{-5}$ , MAPE = 1.067%, SSE = $2.572 \times 10^{-7}$ , R <sup>2</sup> = 0.9998	Modeled and predicted viscosity
Hemmat Esfe <i>et al.</i> <sup>32</sup>	TiO <sub>2</sub> /water	2016	ANN	R <sup>2</sup> = 0.9998	Predicted dynamic viscosity
Giwa SO <i>et al.</i> <sup>33</sup>	Alumina–ferrofluid hybrid	2020	Enhanced ANN, ANFIS, curve fitting	ANN and ANFIS models error less than 4.5% and 3.9% respectively	Modeled thermophysical properties
Meybodi <i>et al.</i> <sup>34</sup>	Al <sub>2</sub> O <sub>3</sub> , TiO <sub>2</sub> , SiO <sub>2</sub> , CuO/water	2015	Least Squares Support Vector Machine (LSSVM)	R <sup>2</sup> = 0.998	Predicted viscosity
Dadhich <i>et al.</i> <sup>6</sup>	TiO <sub>2</sub> , Al <sub>2</sub> O <sub>3</sub>	2023	Gradient Boosting Algorithm & Grid Search Optimization (GBR-GSO)	R <sup>2</sup> = 0.992	Nusselt's number prediction of nanofluids
Adun <i>et al.</i> <sup>35</sup>	Fe <sub>3</sub> O <sub>4</sub> –Al <sub>2</sub> O <sub>3</sub> –ZnO/ternary hybrid nanofluid of water	2021	ANN	R <sup>2</sup> (ANN) = 0.9555	A robust Specific Heat Capacity (SHC) model was developed.
Jamei <i>et al.</i> <sup>36</sup>	Carbon + Metallic oxide	2021	GPR, General Regression Neural Network (GRNN), and RF	R (GPR) = 0.9997, R (GRNN) = 0.9956, R (RF) = 0.9976	Assessed specific heat capacity for solar applications and compared to the other two ML techniques, GPR was better.
Alade <i>et al.</i> <sup>37</sup>	CuO/water	2020	ANN and SVR	RMSE (SVR) = 0.0023, RMSE (ANN) = 0.0025	Compared with ANN, SVR was better
Jamei <i>et al.</i> <sup>38</sup>	Various nanofluids	2021	Gene Expression Programming (GEP)	R = 0.9570, MAPE = 5.5134 RMSE = 0.065	Estimated specific heat capacity using GEP based SHC model
Shi <i>et al.</i> <sup>39</sup>	Carbon-based magnetic nanofluids	2021	ANN	R = 0.95	Predicted thermo-physical properties using ANN based SHC model
Hassan and Banerjee <sup>40</sup>	Molten salt-based nanofluids	2019	ANN	MAPE = 2%, R <sup>2</sup> = 0.9998,	Estimated specific heat capacity ANN based SHC model

loss function. Tuning clarified the penalty of physics constraints and offered precise predictions, which were adhered to physical laws. The model was trained on the whole training set using the best hyperparameters, which were found upon adjustment.

a. *Cross-validation.* The current research included cross-validation and an 80/20 data split for training and testing. Cross-validation reduced biases from a single split by splitting the data into several folds and alternately using these folds for training

**TABLE III.** Description of the statistical metrics used on a dataset.

Nanofluids	Al <sub>2</sub> O <sub>3</sub>	CuO	TiO <sub>2</sub>	SiO <sub>2</sub>	All
Count of data points	486	178	102	26	792
Min particle size (nm)	10	11	21	12	10
Max particle size (nm)	150	33	95	12	150
Min volume fraction (%)	0.03	0.15	0.2	0.45	0.03
Max volume fraction (%)	13.06	9	11.2	4	13.061
Min temperature (°C)	10	10	10	19.89	10
Max temperature (°C)	72	64.17	70	72	72
Min base fluid (water) viscosity (mPa s)	0.393	0.438	0.4	0.393	0.393
Max base fluid (water) viscosity (mPa s)	1.306	1.306	1.31	0.995	1.306
Min nanofluid viscosity (mPa s)	0.453	0.425	0.45	0.574	0.425
Max nanofluid viscosity (mPa s)	4.635	3.374	4.59	2.507	4.635

and validation. This technique ensured that the model's generalizability was continuously assessed across various data subsets, thereby enhancing its performance.

## B. PGXGB model overview

We explored the combination of XGBoost techniques with physics-driven models for predicting the viscosity of water-based nanofluids. Viscosity is influenced by many factors, such as temperature, base fluid viscosity, volume fraction, and nanoparticle size.

### 1. XGBoost algorithm

Although regression trees are excellent predictors, they can be time-consuming and very sensitive to variations in data. To overcome this difficulty, XGBoost utilizes a hybrid technique called Regression Tree and Classification (RTAC).<sup>56</sup> XGBoost is an ensemble learning technique that combines several base learners to achieve more accurate predictions. XGBoost performs best when these base learners are individually somewhat weak in certain aspects, allowing their errors to cancel out when combined, resulting in a more accurate overall prediction. The algorithm has two primary components: the objective function and the base learners. The objective function

itself comprises two parts: the loss function and a regularization term. The loss function also helps manage the model's complexity. If the model becomes too complicated, it can lead to overfitting, which then requires penalizing the model as it grows more complex and overfitted. Regularization techniques are employed to achieve this. Each tree group in XGBoost consists of three distinct components: the leaves, the internal nodes, and the root node.

### 2. Proposed physics-guided XGBoost model architecture

Machine learning models, such as XGBoost, can effectively capture complex patterns in data but may occasionally produce results that violate known physical laws. To address this, we integrate domain-specific physics constraints into the learning process. Our approach combines a physics-based model, which ensures consistency with physical laws, with the XGBoost algorithm. Physics-based outputs serve as auxiliary inputs, guiding the learning process toward physically plausible predictions. We have a physics-based model  $y_{\text{physics}}$  that generates simulated values of target variable  $y$  that follow the rules and constraints of physics laws, given as  $y_{\text{physics}} = f_{\text{physics}}(x, y)$ , where  $x$  is the input parameter that is physically connected to target variable  $y$ . In addition, the machine learning algorithm, i.e., the XGBoost model, is represented by the function  $\hat{y} = f_{\text{XGBoost}}(x, y)$ . A new physics-guided XGBoost (PGXGB) algorithm is developed by combining both the models. Therefore using  $y_{\text{physics}}$  instead of target variable  $y$ , we get

$$\hat{y} = f_{\text{PGXGB}}(x, y_{\text{physics}}). \quad (1)$$

Viscosity parameters are major area of concern when we integrate physics-based laws into the XGBoost algorithm learning process. This guarantees that the predictions made by the model follow the principles of physics. The current research utilizes two constraints in the development of our physics-based model. First and foremost, viscosity must be positive, indicating that it cannot be negative. Second, viscosity in Newtonian fluids often decreases with increasing temperature.

This hybrid framework, referred to as the physics-guided XGBoost (PGXGB) model, is illustrated in Fig. 1, which shows its architecture.

The following algorithm represents the physics-based restrictions on the XGBoost loss function:

18 October 2025 144606

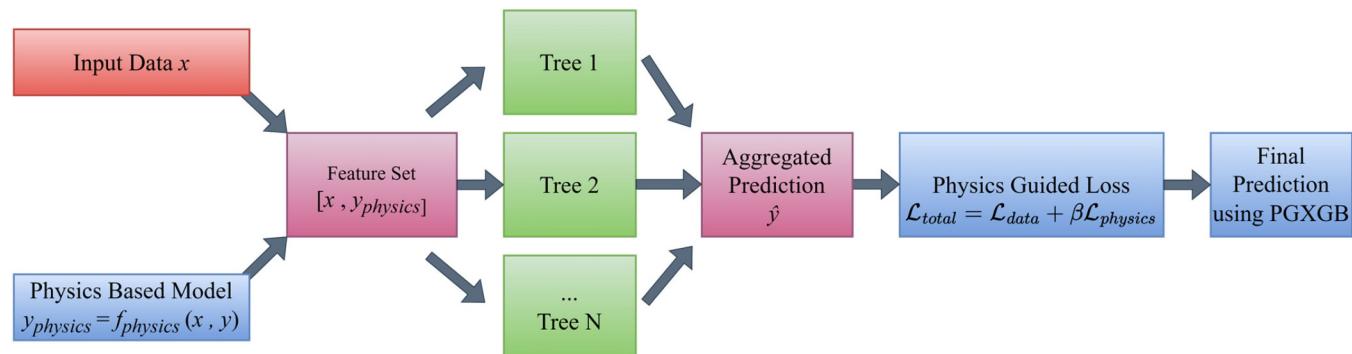


FIG. 1. Physics-based XGBoost model architecture.

**ALGORITHM (1) to Compute Physics-Constrained Loss Function.**

**Step 1:** Compute MSE loss  $L_{MSE}(\hat{y}, y)$ .

$$L_{MSE}(\hat{y}, y) = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2.$$

**Step 2:** Compute viscosity constraint loss  $L_{Viscosity}(\hat{y})$  with the constraint that viscosity is always positive and greater than zero.

$$L_{Viscosity}(\hat{y}) = \frac{1}{N} \sum_{i=1}^N (\max(0, -\hat{y}_i)).$$

**Step 3:** Compute temperature constraint loss function  $L_{Temperature}(\hat{y})$  using the constraint that viscosity usually decreases with increasing temperature for Newtonian fluids.

$$\Delta_t = \sum_{i=0}^{N-1} (T_{i+1} - T_i), \Delta_v = \sum_{i=0}^{N-1} (\hat{y}_{i+1} - \hat{y}_i), \Delta_{ev} = -\beta^* \Delta_t$$

$$L_{Temperature}(\hat{y}) = \frac{1}{N} \sum_{i=1}^N (\max(0, (\Delta_v - \Delta_{ev}))).$$

Here, temperature gradient  $\Delta_t$  is the sum of differences between successive temperatures, viscosity gradient  $\Delta_v$  is the sum of differences between successive predicted viscosities, expected viscosity gradient  $\Delta_{ev}$  is obtained by negating the  $\Delta_t$  with constant beta  $\beta$  term that represents the expected change in viscosity per unit temperature change.

**Step 4:** Compute Physics-Based Total loss  $L_{Physics}(\hat{y}, y)$  by combining all the losses with penalty weights.

$$L_{Physics}(\hat{y}, y) = L_{MSE}(\hat{y}, y) + \lambda_v * L_{Viscosity}(\hat{y}) + \lambda_t * L_{Temperature}(\hat{y}).$$

$\lambda_v$  and  $\lambda_t$  are penalty factors for violating each physical constraint.

**Step 5:** Output the final physics-constrained total loss  $L_{Physics}(\hat{y}, y)$ .

Literature reports indicate that nanofluid viscosity may deviate from strict monotonic decrease with temperature under certain conditions, such as the following:

- High particle concentrations lead to agglomeration and cluster formation.
- Non-Newtonian behavior at elevated shear rates.

We are not forcing the model to strictly obey the physics law trend that if concentration increases, then temperature will always increase at the same rate. As during training, the physics-based constraint is applied via a soft regularization parameter penalty  $\beta$  rather than a hard rule, this regularization parameter is treated as a tunable hyperparameter optimized via GridSearchCV and if the real-world data disagrees (because of physical phenomena such as particle agglomeration or non-Newtonian effects at high concentrations), the model is free to follow the data. This  $\beta$  controls how strongly the model is nudged toward the trend. For a larger value of  $\beta$ , the model strongly follows the inverse trend, and for a small  $\beta$ , the model relies more on the raw data patterns.

To account for variations in nanofluid behavior, both global and type-specific  $\beta$  values are considered. For instance,  $\text{Al}_2\text{O}_3$  nanofluids typically exhibit a stronger temperature-viscosity dependence than  $\text{SiO}_2$ , justifying different  $\beta$  values across materials. The

selection of  $\beta$  also depended on dataset characteristics, such as temperature range and distribution.

Future enhancements could involve adaptive  $\beta$  values that respond to local trends during training, along with sensitivity analyses to assess  $\beta$ 's influence on model performance. Proper calibration of  $\beta$  significantly enhances the physical validity and robustness of viscosity predictions.

The physics-guided loss function  $L_{Physics}(\hat{y}, y)$  measures the discrepancy between the predicted values  $\hat{y}$  and the true values  $y$  while incorporating physical constraints.

The following algorithm represents the physics-guided XGBoost (PGXGB) algorithm to calculate the final predicted value:

**ALGORITHM (2) PGXGB.**

Initialize the model with a constant value, typically the mean of the target variable:  $\hat{y}_i^{(0)} = \bar{y}$

For  $m = 1$  to  $M$ :

- (a) Compute the negative gradients of the physics-guided loss function with respect to the current predictions:

$$g_i = -\frac{\partial L_{Physics}(\hat{y}_i^{(m-1)}, y_i)}{\partial \hat{y}_i^{(m-1)}}.$$

- (b) Compute the second-order gradients (Hessians) of the physics-guided loss function:

$$h_i = \frac{\partial^2 L_{Physics}(\hat{y}_i^{(m-1)}, y_i)}{\partial (\hat{y}_i^{(m-1)})^2}.$$

- (c) Construct a new decision tree  $f_m(x)$  by optimizing the following objective function:

$$\begin{aligned} \mathcal{L}^{(m)} &= \sum_{i=1}^N \left[ g_i f_m(x_i) + \frac{1}{2} h_i f_m^2(x_i) \right] + \Omega(f_m), \\ &= \sum_{j=1}^{J_m} \left[ \left( \sum_{x_i \in R_{j,m}} g_i \right) w_{j,m} + \frac{1}{2} \left( \sum_{x_i \in R_{j,m}} h_i \right) w_{j,m}^2 \right] \\ &\quad + \gamma T_m + \frac{1}{2} \lambda \sum_{j=1}^{J_m} w_{j,m}^2. \end{aligned}$$

Here,  $R_{j,m}$  is the  $j^{\text{th}}$  leaf node of the  $m^{\text{th}}$  tree,  $w_{j,m}$  is the weight (score) of the  $j^{\text{th}}$  leaf node,  $T_m$  is the number of leaves in the  $j^{\text{th}}$  tree,  $\gamma$  and  $\lambda$  are regularization parameters, and  $J_m$  is the total number of leaf nodes in the  $m^{\text{th}}$  tree.

Update the predictions:  $\hat{y}_i^{(m)} = \hat{y}_i^{(m-1)} + \eta f_m(x_i)$

Output the final prediction:  $\hat{y}_i = \sum_{m=1}^M f_m(x_i)$

Here,  $\mathcal{D} = (x_i, y_i)_{i=1}^N$  is the training data, where  $x_i$  are the input features and  $y_i$  are the target values. The regularization term  $[\Omega(f)]$  for new decision tree  $f$  helps to control the complexity of the model and prevent overfitting. The learning rate ( $\eta$ ) controls the contribution of each individual tree to the final prediction. It is a

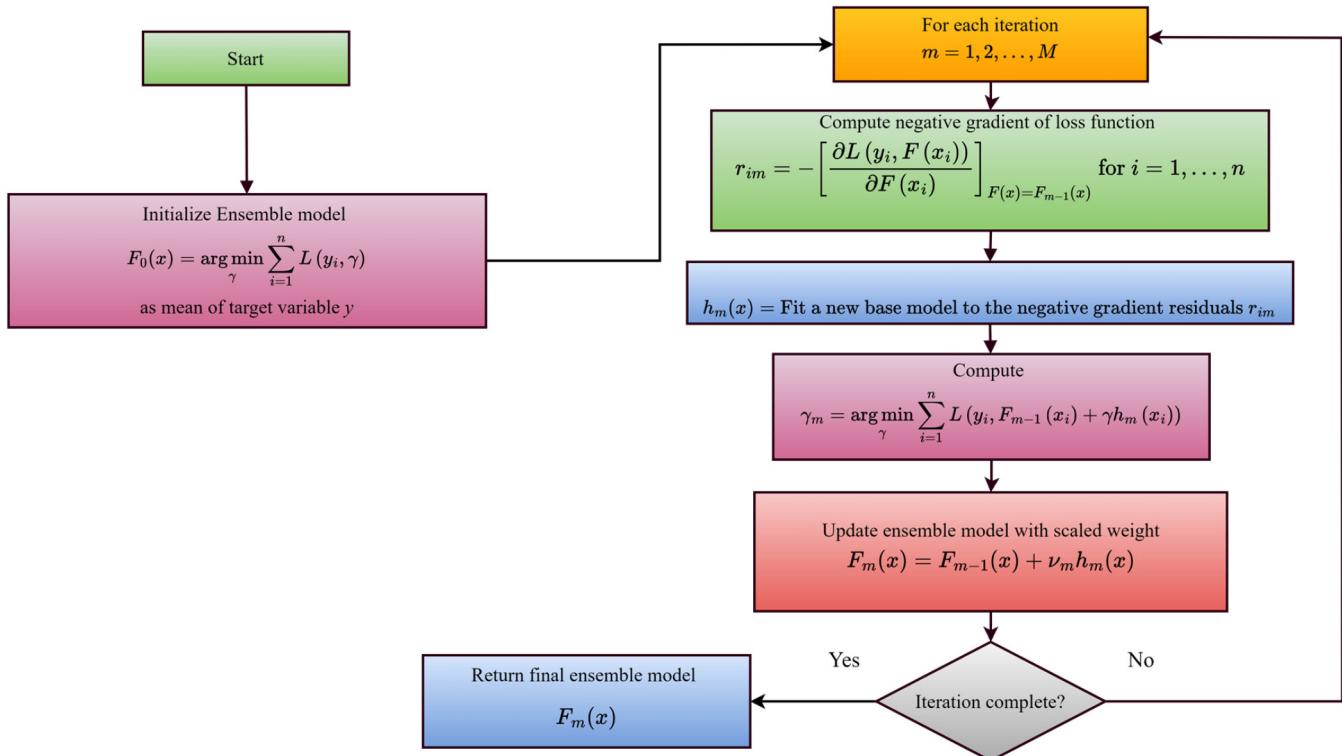


FIG. 2. The Gradient Boosting Regressor (GBR) algorithm.

tuning parameter for controlling the update in predictions. The maximum number of trees ( $M$ ) determines the depth of the ensemble model. This approach integrates the physics-guided loss function into the XGBoost framework by calculating its gradients and Hessians with respect to the predictions. These gradients and Hessians are then used to construct the objective function for training each tree in the ensemble. By incorporating physical constraints into the loss function, the PGXGB model aims to learn a model that not only fits the data accurately but also adheres to underlying physical laws. This process ultimately leads to more precise and interpretable models.

### C. Gradient Boosting Regressor model

The Gradient Boosting Regressor belongs to a class of techniques known as ensemble learning, which combines a number of weak models to create a powerful prediction model. In the Gradient Boosting Regressor, the ensemble method builds a strong predictive model by combining many weak regression trees.<sup>6,57</sup>

The accuracy of the model is progressively increased by training each new tree on the mistakes made by the ones that came before it; to keep the model from being too complicated and over fitting the data, it makes use of parameters such as the learning rate, maximum

depth, and maximum features; also, it can indicate the relative importance of each feature for forecasting; and it can handle input that has missing data. Figure 2 describes the steps of the GBR algorithm.

### D. $\nu$ -SVR: $\nu$ -Support vector regression

$\nu$ -SVR is a kernel-based regression approach that was motivated by Support Vector Machines (SVMs). It looks for a flat function with a small parameter ( $\nu$ ) that minimizes training mistakes and flatness while accommodating the majority of data points within a certain tolerance threshold. Schölkopf *et al.*<sup>58</sup> formulated  $\nu$ -SVR algorithm, which substitutes the parameter  $\nu$  for the C parameter used in the SVR technique. It establishes a lower bound on the proportion of support vectors and an upper bound on the proportion of training mistakes.

#### 1. An overview of $\nu$ -SVR mathematically

Here,  $x_i$ 's are the training data for  $i = 1, \dots, N$ ,  $O_i$  is the output value, and  $x_i$  is an input vector of dimension  $d$ ; the input data are mapped to a higher-dimensional feature space using the kernel function  $K(x_i, x_j)$ , and these kernel functions are used to handle nonlinear relationships,  $\nu$  ( $0 < \nu \leq 1$ ), a parameter that

strikes a balance between the function's flatness and the quantity of training mistakes.

## 2. $\nu$ -Tube loss term

The method for identifying outliers is defined by the ( $\nu$ )-tube loss term,

$$\frac{1}{N} \sum_{i=1}^N L_\nu(O_i - f(x_i)), \quad (2)$$

where  $L_\nu(O_i - f(x_i)) = \max(0, |O_i - f(x_i)| - \nu)$  is the  $\nu$  tube loss function that is insensitive to parameter epsilon ( $\varepsilon$ ). The parameter  $\nu$  in  $\nu$ -SVR controls the amount of support vectors and training mistakes within the flexible tube by automatically adjusting it to minimize  $\varepsilon$ .

## 3. Objective function of $\nu$ -SVR

The objective functions of  $\nu$ -SVR<sup>58</sup> can be restated as

$$\underset{w, b, \xi, \xi^*, \nu}{\text{Minimize}} \frac{1}{2} w^T w + C \left( \nu + \frac{1}{l} \sum_{i=1}^l (\xi_i + \xi_i^*) \right) \quad (3)$$

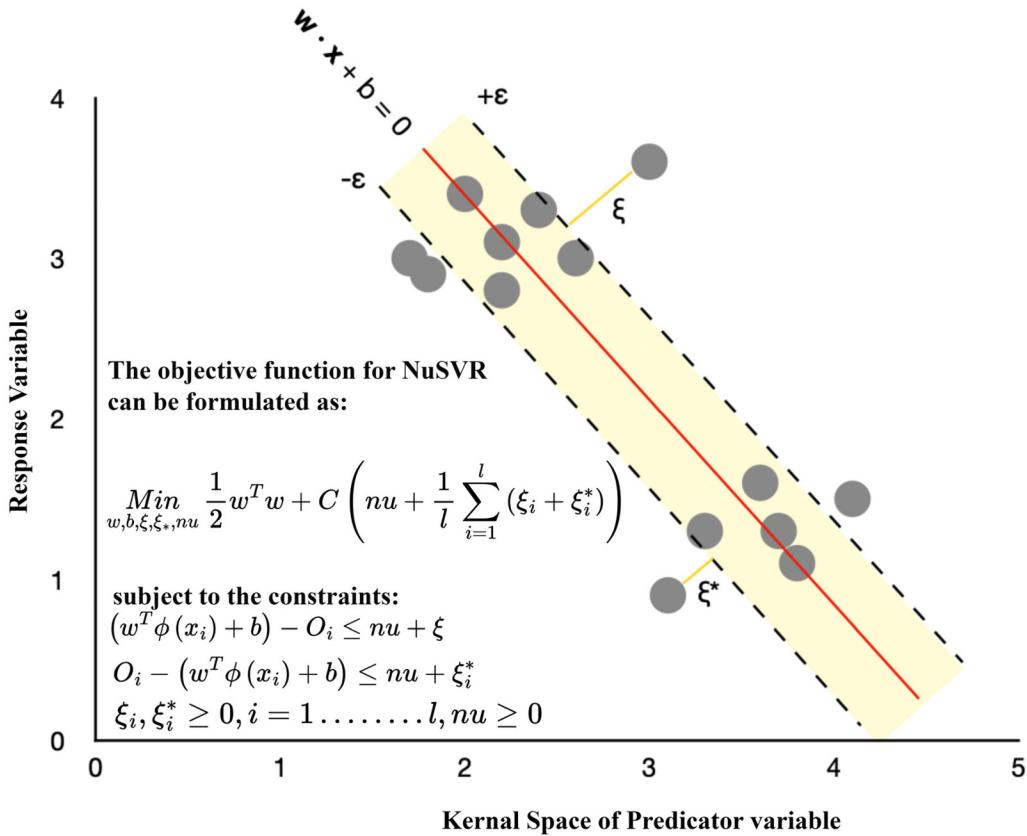
$$\text{Subject to} \begin{cases} (w^T \phi(x_i) + b) - O_i \leq \nu + \xi_i \\ O_i - (w^T \phi(x_i) + b) \leq \nu + \xi_i^* \\ \xi_i, \xi_i^* \geq 0, i = 1 \dots l, \nu \geq 0, \end{cases} \quad (4)$$

Lagrange multipliers  $\alpha_i^*$ ,  $\alpha_i$  are used in optimization to reduce the objective function while adhering to constraints, weight vector  $w$  value and bias  $b$  value are evaluated with the help of Karush–Kuhn–Tucker (KKT) equation. This leads to a dual optimization problem that the kernel trick solves in the feature space.

Equation (3) can be seen as the following convex quadratic problem:

$$\underset{\alpha_i^*, \alpha_i}{\text{Maximize}} \begin{cases} \sum_{i=1}^l (\alpha_i^* - \alpha_i) O_i = 0, \\ -\frac{1}{2} \sum_{j,i=1}^l (\alpha_j^* - \alpha_j)(\alpha_i^* - \alpha_i) k(x_i, x_j), \end{cases} \quad (5)$$

$$\text{Subject to} \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0, \sum_{i=1}^l (\alpha_i + \alpha_i^*) \leq C \cdot \nu \text{ and } \alpha_i^* \in [0, \frac{C}{l}]. \quad (6)$$



18 October 2025 14:46:06

FIG. 3. The  $\nu$ -SVR soft margin loss.

The final regression function and weight vector are expressed as follows:

$$f(x) = \sum_{i=1}^l (\alpha_i^* - \alpha_i) k(x_i, x) + b, \quad (7)$$

$$W = \sum_{i=1}^l (\alpha_i^* - \alpha_i) \Phi(x_i). \quad (8)$$

Equation (7) is used for predicting new data points ( $x$ ). Benefits of  $v$ -SVR include handling non-linear relationships through the kernel trick and offering flexibility in controlling the accuracy-complexity trade-off with the  $v$  parameter. By introducing the parameter epsilon ( $\epsilon$ ),  $v$ -SVR creates an insensitive zone surrounding the real target values. Penalties apply to predicted values that fall outside of this  $\epsilon$ -tube. The goal function is punished for predictions that are above or below the tube by  $\xi_i, \xi_i^*$ , correspondingly, encouraging a more robust and sparse solution (Fig. 3). In order to reduce the computational strain caused by high-dimensional data, it is useful to resolve the  $v$ -SVR problem's dual formulation. This conversion decreases the feature space from  $N$  to  $S$ , in which the number of support vectors is denoted by  $S$ . Computational complexity hinges on the support vectors

while operating in dual space, which results in more efficient computations, particularly when dealing with high-dimensional datasets.

### E. Extra Tree Regressor

Extra Tree Regression<sup>59</sup> is a tree ensemble learning approach that improves the accuracy and robustness of regression models by integrating the predictions of numerous decision trees. This technique, a variation in the random forest algorithm, builds trees differently. Every tree in a random forest is formed using a bootstrap sample of training data, with each node considering a random subset of features for splitting. This method enhances generalization efficiency by lowering individual tree correlation. Extra Tree Regression, on the other hand, selects input samples and characteristics at random for splitting at each node to develop trees. This replacement-free randomization means that there is only one selection of each sample and feature per tree. Moreover, trees develop to their full depth without pruning, which can allow them to capture complex data patterns. In some circumstances, the randomization and lack of pruning in Extra Tree Regression can reduce bias and boost tree variety, improving overall performance compared to conventional decision tree models or random forests.<sup>60</sup> Figure 4 shows the steps used in the Extra Tree Regressor algorithm.

18 October 2025 14:46:06

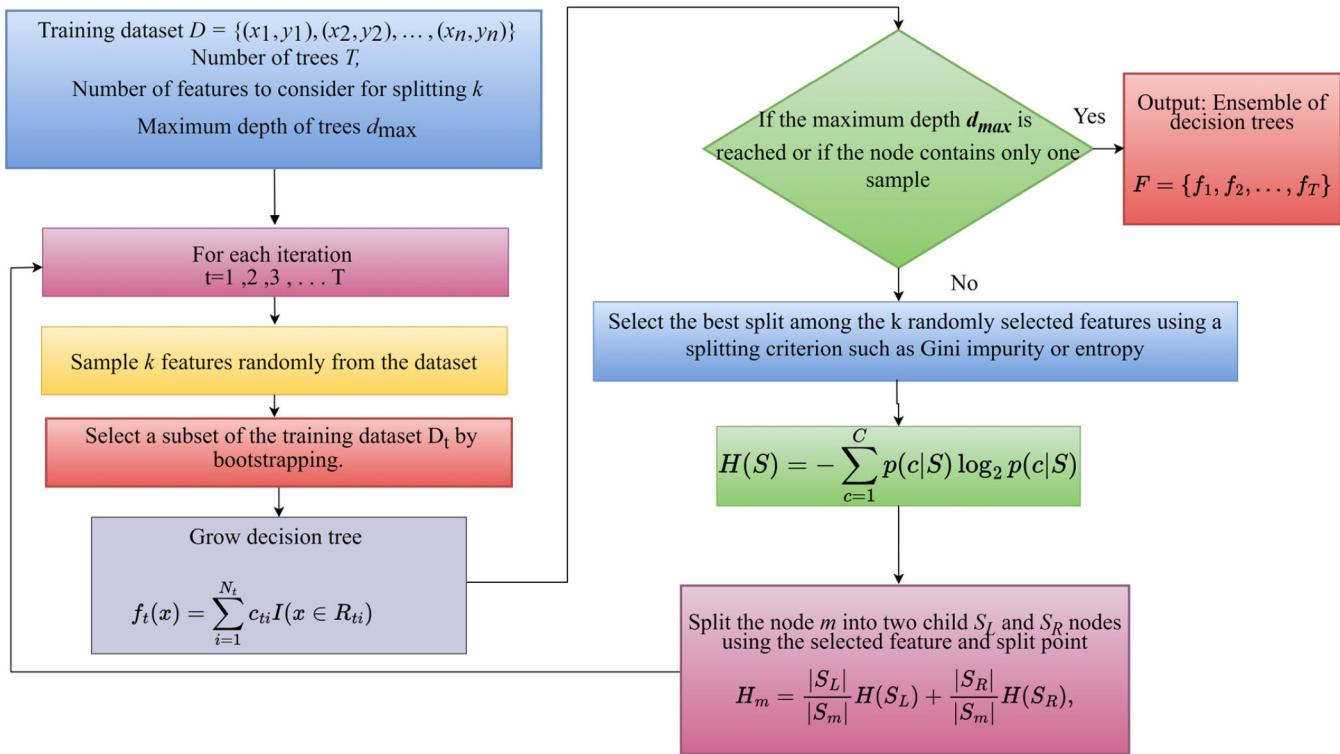


FIG. 4. The Extra Tree Regressor algorithm.

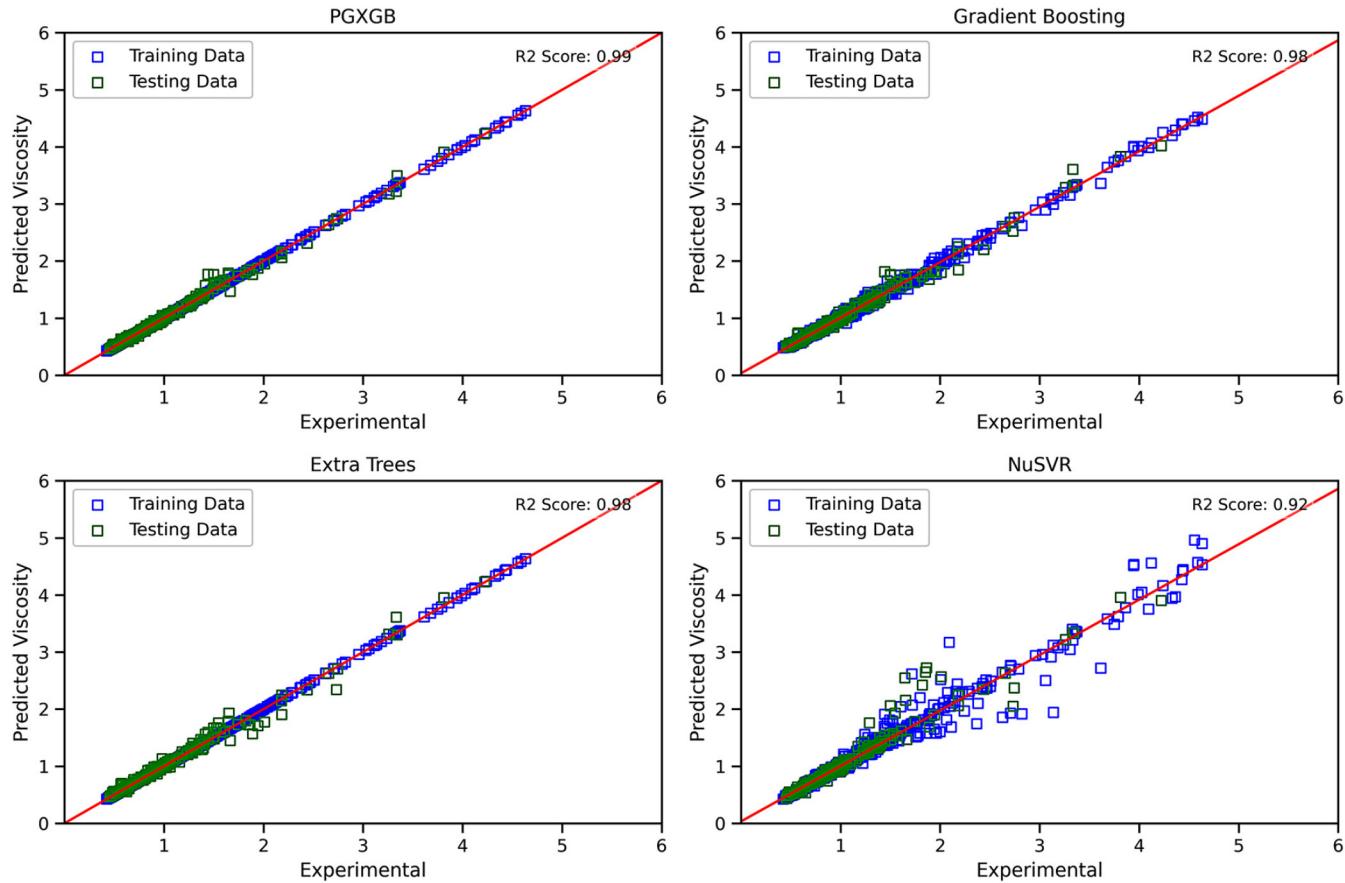


FIG. 5. Scattered plot of testing and training dataset for PGXGB, Gradient Boosting Regressor, Extra Tree Regressor, and  $\nu$ -SVR models.

18 October 2025 144606

### III. RESULTS AND DISCUSSION

This section describes how the previously described PGXGB model is applied to determine the viscosity of nanofluids and discusses the significant impacts of the related parameters. A variety of graphical and statistical analyses were carried out to confirm the validity and precision of the proposed PGXGB model for nanofluid viscosity prediction.

#### A. Performance evaluation and analysis of PGXGB model accuracy

The graphical comparison between predicted and actual experimental viscosities using the proposed model is shown in Fig. 5. With a coefficient of determination ( $R^2$ ) of 0.992248 for the PGXGB model, Fig. 5 shows that the predicted and actual experimental viscosities of PGXGB model match quite well.

The graphs in Fig. 6 show the relative error as a function of experimental viscosity and offer additional confirmation. Remarkably, the majority of relative errors are concentrated near the zero red line, indicating that the difference between predicted

and actual values is very small. This demonstrates the outstanding predictive power of the PGXGB model. Larger deviations in the model's predictions, particularly in the relative error plot, stem from extreme experimental conditions, dataset characteristics, and unmodeled interactions. High nanoparticle concentrations or low temperatures introduce nonlinear effects on viscosity, making accurate modeling challenging. Outliers in experimental data, often due to measurement errors, calibration issues, or inconsistencies in nanoparticle synthesis, also contribute to these deviations.

Data imbalance is another factor; some nanoparticle types (e.g.,  $\text{SiO}_2$  with 26 points) are underrepresented compared to others (e.g.,  $\text{Al}_2\text{O}_3$  with 486 points), reducing accuracy for less frequent conditions. Predictions near dataset boundaries (extremes in particle size, volume fraction, or temperature) tend to be less accurate due to limited training data in these regions. Furthermore, unmodeled interactions such as nanoparticle aggregation and electroviscous effects may not be fully captured, especially under extreme conditions. Specifically, the model may underperform where nonlinear effects become significant, such as at very small or large nanoparticle sizes or at higher temperatures where the

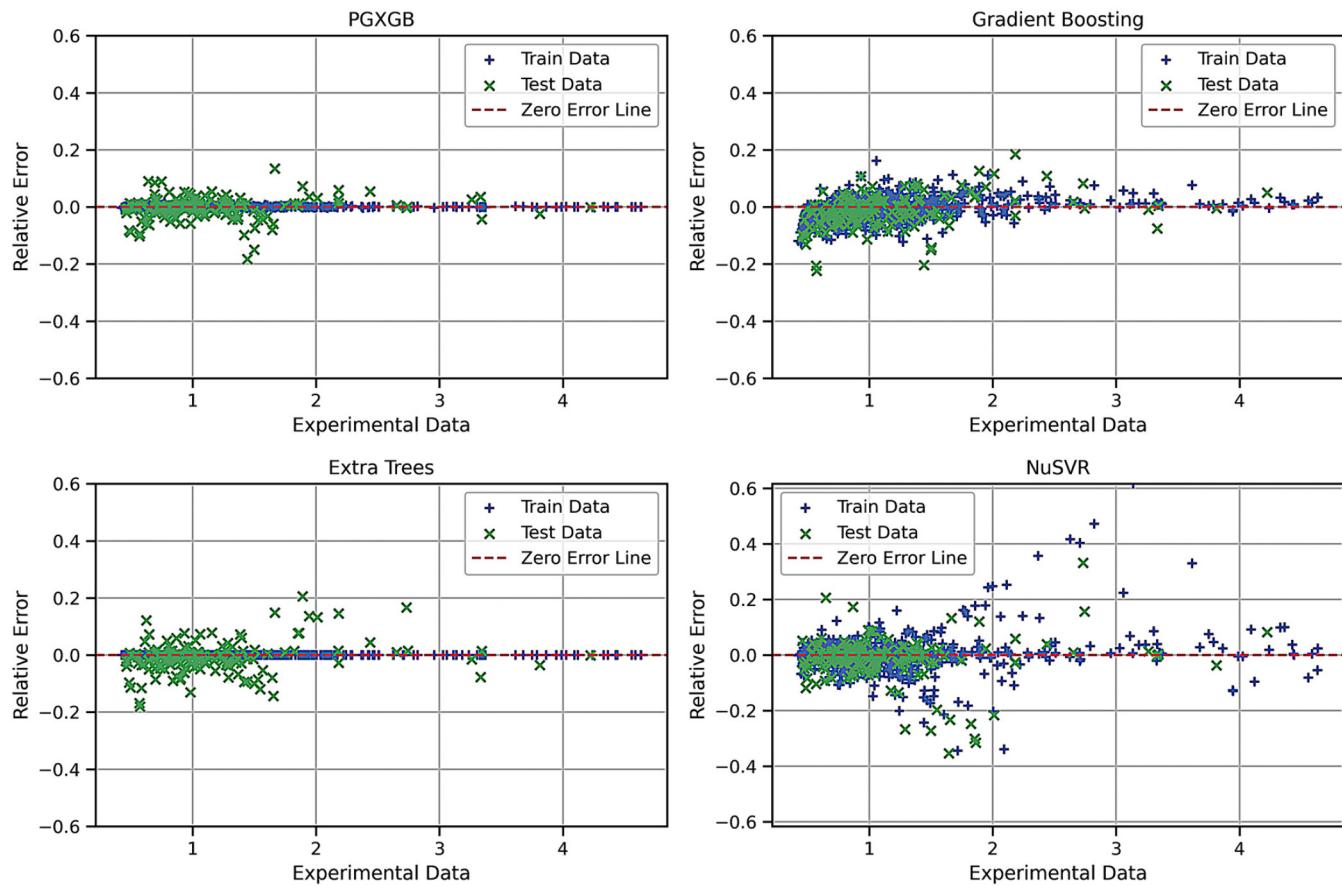


FIG. 6. Relative error vs training and testing data plot for PGXGB, Gradient Boosting Regressor, Extra Tree Regressor, and  $\nu$ -SVR models.

18 October 2025 144606

viscosity–temperature relationship deviates from Newtonian assumptions. Nanoparticle types with fewer data points, like CuO, also challenge the model's generalization. To address these issues, future work could involve data augmentation to balance the dataset, incorporating additional features (e.g., pH, particle shape, synthesis methods) to capture more complex effects and exploring ensemble or hybrid models that combine PGXGB with domain-specific corrections for extreme cases. These approaches could significantly enhance model accuracy and robustness.

A more comprehensive comparison of the models is shown in Fig. 7, which plotted cumulative frequency vs absolute relative error; PGXGB outperformed all other models, correctly predicting 95% of the data points with an absolute relative error of less than 0.1%. Furthermore, Fig. 7 also shows that gradient boosting model performed consistently and satisfactorily in all ranges of absolute relative inaccuracy.

#### B. Performance evaluation using Taylor diagrams

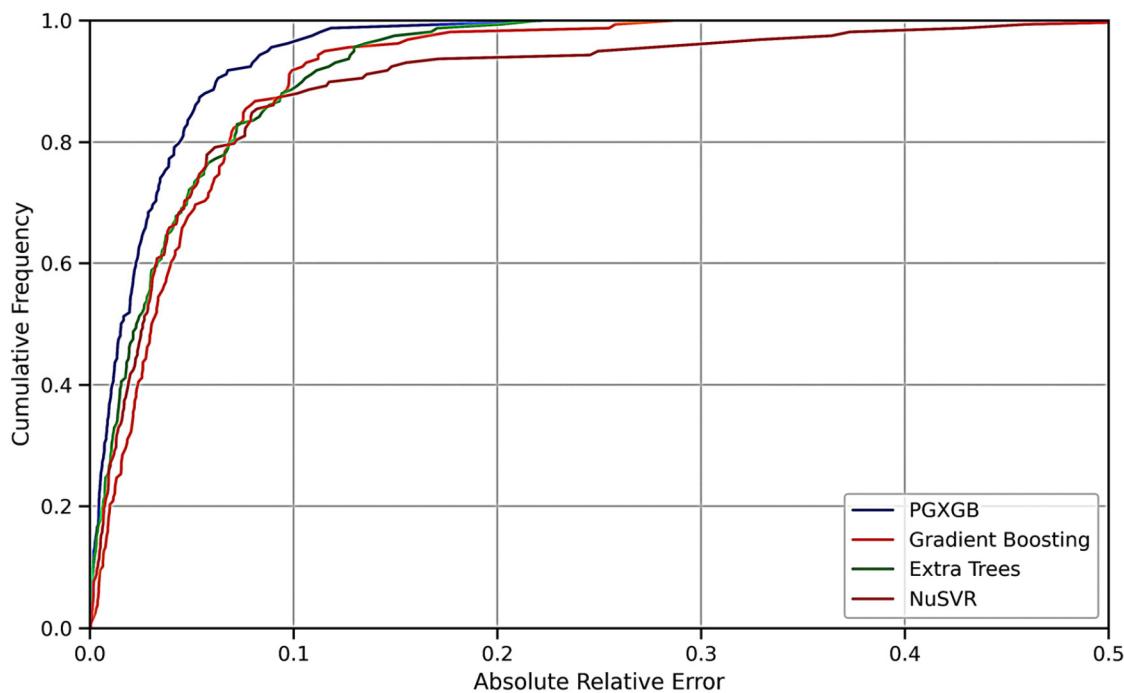
The results of this research are also shown in the Taylor diagram in Fig. 8. The PGXGB model value is closest to the

reference data point, which indicates a strong correlation and minimal deviation from the actual values. This position proves that when compared to GBR, ET, and  $\nu$ -SVR models, PGXGB performs better in terms of reliability and accuracy.

#### C. Comparison and analysis of the proposed PGXGB model's prediction performance against several existing models

The PGXGB model's accuracy in predicting nanofluid viscosity is displayed in Fig. 9. In theoretical models, viscosity is typically correlated with the volume fraction in the nanofluid. We evaluated the predictions of the PGXGB model in several scenarios with actual testing and other models. In contrast to previous models that have trouble with predicting viscosity across particle volume fractions, Fig. 9 demonstrates that the PGXGB model accurately matches actual experiments. The fluid viscosity increases as the concentration of nanoparticles increases.

The PGXGB model demonstrates superior performance compared to empirical models for viscosity prediction, as seen in Fig. 9. This advantage is attributed to the model's capacity to predict the



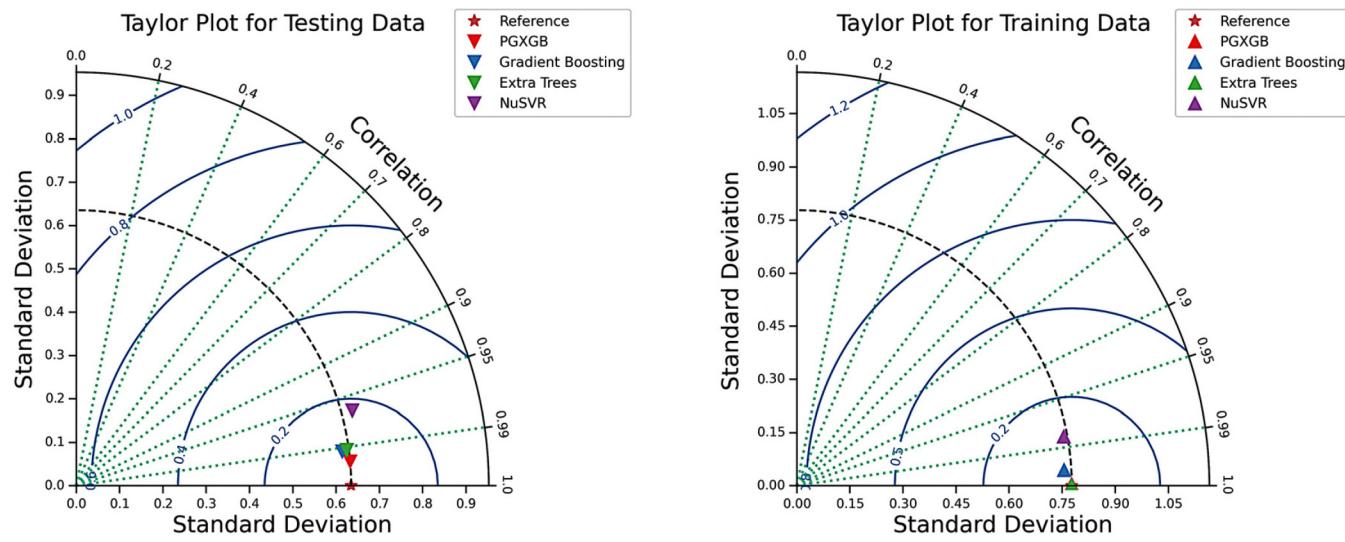
**FIG. 7.** Cumulative frequency vs absolute relative error plot for PGXGB, Gradient Boosting Regressor, Extra Tree Regressor, and  $\nu$ -SVR models.

18 October 2025 14:06:06

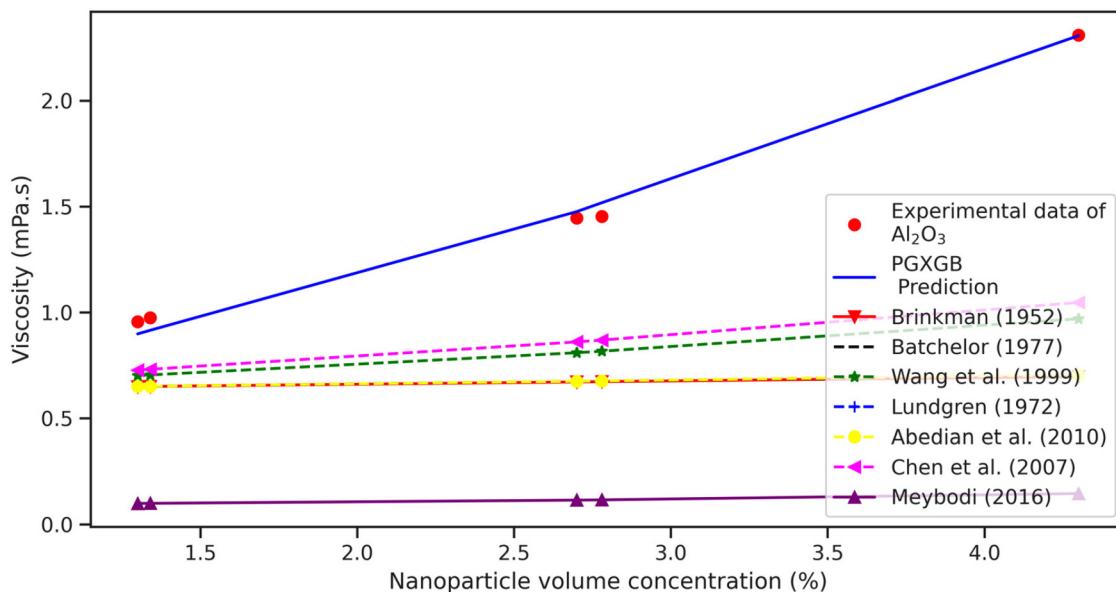
nonlinear relationships between nanoparticle volume fractions, temperature, and viscosity using gradient boosting, a feature most empirical models lack. The addition of a physics-guided loss function further improves the model by enforcing predictions to adhere

to the laws of physics and domain-specific principles, thus extending data-based learning with theoretical constraints.

Unlike empirical models, which employ specific assumptions such as linear approximations or power laws, the PGXGB model



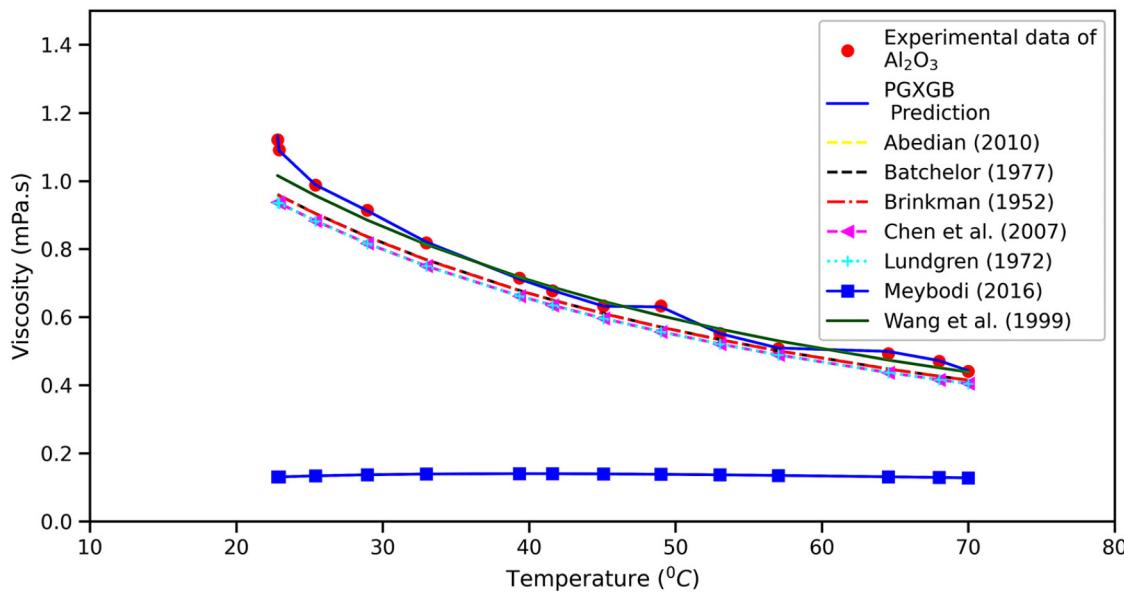
**FIG. 8.** Taylor diagram illustration showing how important parameters affect the relative viscosity prediction.



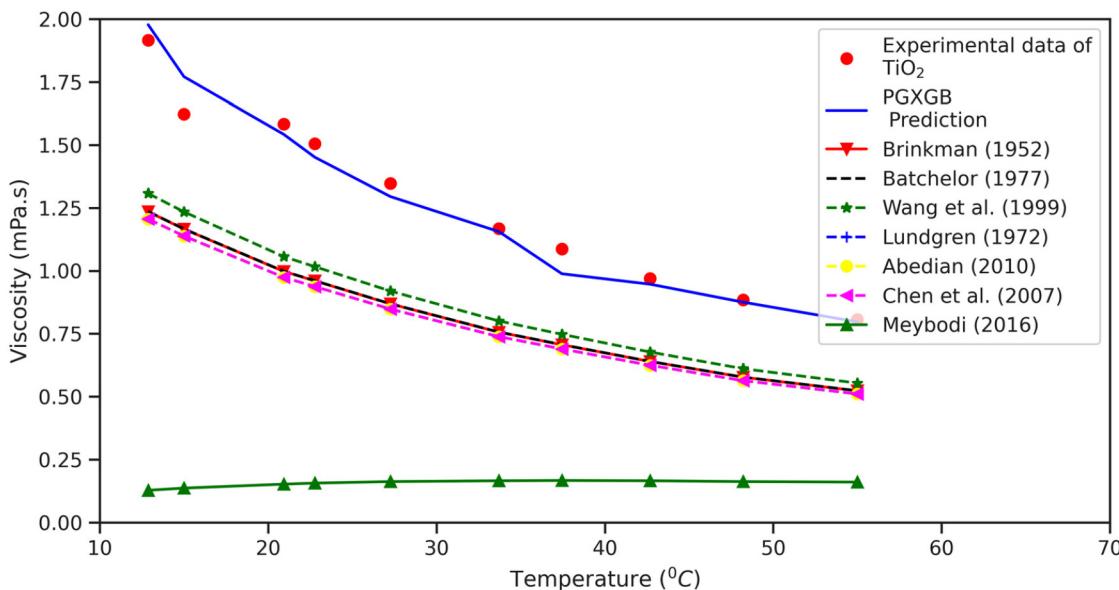
**FIG. 9.** Analysis of predicted viscosity vs volume concentration by the PGXGB model in comparison with experimental data and a few useful empirical and theoretical models: Al<sub>2</sub>O<sub>3</sub>: size 13 nm and temperature 42 °C.

utilizes a robust feature set, including nanoparticle size, type, and base fluid viscosity, and then learns the complex interactions among these features for improved prediction. This flexibility across experimental conditions also contributes to the PGXGB

model's outperformance. While standard empirical models are parameterized for specific conditions and struggle with generalization, the PGXGB model is easily adaptable to a broad range of nanoparticle volume fractions and extreme temperatures.



**FIG. 10.** Analysis of predicted viscosity vs temperature by the PGXGB model in comparison with experimental data and a few useful empirical and theoretical models: Al<sub>2</sub>O<sub>3</sub> with  $d_p = 36$  nm and  $\varphi = 1\%$ .

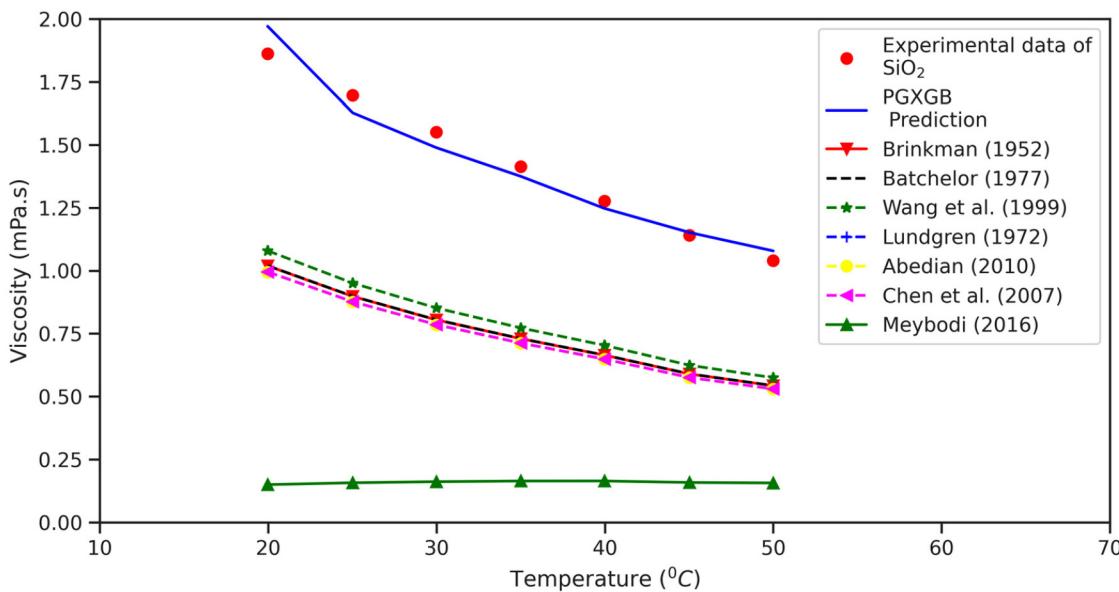


**FIG. 11.** Analysis of predicted viscosity vs temperature by the PGXGB model in comparison with experimental data and a few useful empirical and theoretical models:  $\text{TiO}_2$  with  $d_p = 21 \text{ nm}$  and  $\varphi = 2\%$ .

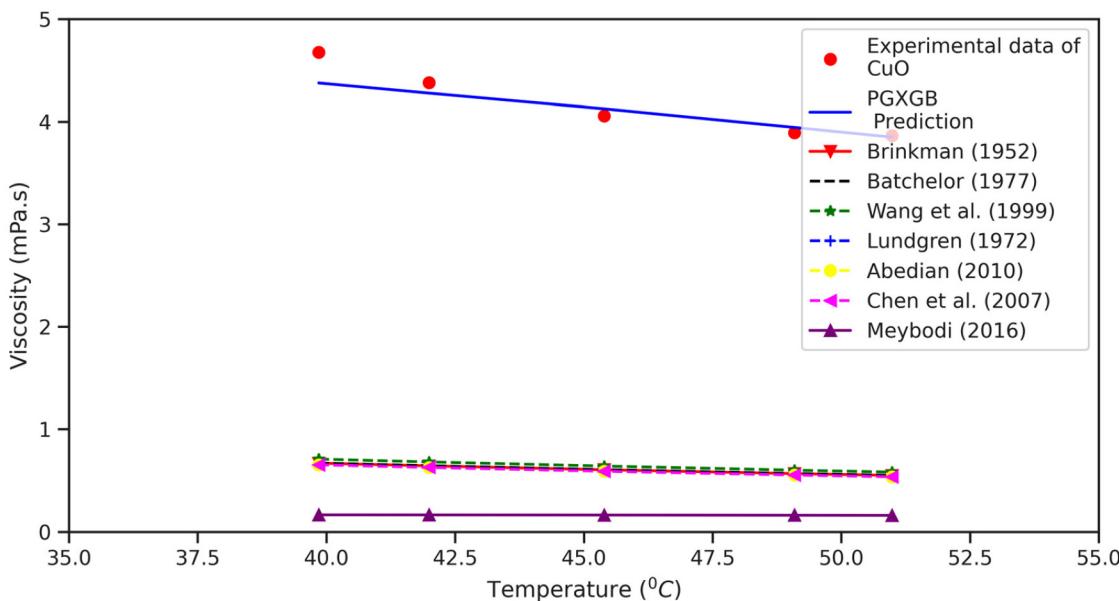
Furthermore, existing literature models, including Brinkman<sup>11</sup> and Batchelor<sup>14</sup>, used static equations derived under specific circumstances, rendering them rigid and less capable of handling variation compared to the PGXGB model, which leverages

machine learning to optimize predictions using a diverse dataset, thereby improving its ability to account for variability. Another advantage of the PGXGB model is its robustness to noise and outliers, inherent in the gradient boosting framework, which

18 October 2025 144606



**FIG. 12.** Analysis of predicted viscosity vs temperature by the PGXGB model in comparison with experimental data and a few useful empirical and theoretical models:  $\text{SiO}_2$  with  $d_p = 12 \text{ nm}$  and  $\varphi = 1.85\%$ .



**FIG. 13.** Analysis of predicted viscosity vs temperature by the PGXGB model in comparison with experimental data and a few useful empirical and theoretical models: CuO with  $d_p = 29$  nm and  $\varphi = 9\%$ .

allows it to handle experimental inconsistencies effectively. Moreover, while empirical models often focus on a single nanoparticle or base fluid, the PGXGB model is versatile, generalizing across multiple nanoparticles including  $\text{Al}_2\text{O}_3$ ,  $\text{TiO}_2$ ,  $\text{SiO}_2$ , and CuO. By combining advanced machine learning techniques, data-driven insights, and physics-based constraints, the PGXGB model offers accurate, versatile, and generalizable algorithms for viscosity prediction across diverse conditions, surpassing empirical methods. Figure 10 illustrates how well the PGXGB model fits actual experimental data for a specific scenario: an  $\text{Al}_2\text{O}_3$ -water solution containing 1% of small 36 nm nanoparticles at varying temperatures. As expected from the experimental results, the PGXGB model accurately predicts that fluid viscosity decreases with increasing temperature. Additionally, Fig. 10 demonstrates the PGXGB model's accuracy in estimating viscosity compared to previous models. Earlier models did not account for temperature or nanoparticle size.

Even the Meybodi model,<sup>34</sup> despite considering the fluid's relationship with temperature, deviated from actual experiments. The PGXGB model, however, matched up nicely. In many situations, the PGXGB model performs better than other models. It is significantly closer to experimental data as shown in Figs. 10–13, which illustrate how it predicts fluid viscosity for a variety of combinations, including  $\text{Al}_2\text{O}_3$ ,  $\text{TiO}_2$ ,  $\text{SiO}_2$ , and CuO. When dealing with varied combinations, temperatures, or sizes of nanoparticles, other models such as Brinkman,<sup>11</sup> Batchelor,<sup>14</sup> Wang *et al.*,<sup>15</sup> and Abedian and Kachanov<sup>17</sup> perform less well. Although the model by Chen *et al.*<sup>16</sup> performs better, it is not perfect. The PGXGB model is a useful tool for estimating fluid viscosity in many scenarios since it performs well in each of these scenarios.

#### D. Statistical metrics comparison of PGXGB model with other soft computing model

A brief description of the many statistical metrics used to assess the prediction model's performance is given in Table IV.

**TABLE IV.** Statistical metrics for evaluating various models.

S. no.	Statistical metrics	Equations
1	AARD	$\frac{1}{n} \sum_{i=1}^n \left  \frac{y_i - \hat{y}_i}{y_i} \right  \times 100\%$
2	$R^2$	$1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$
3	RMSE	$\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$
4	MSE	$\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$
5	RMSD	$\sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}$
6	MAPE	$\frac{1}{n} \sum_{i=1}^n \left  \frac{y_i - \hat{y}_i}{y_i} \right  \times 100\%$

Here,  $n$  is the number of samples,  $y_i$  are the actual values, and  $\hat{y}_i$  are the predicted values.

**TABLE V.** Comparisons of various soft computing models using statistical metrics. Boldface is used to highlight the best and optimal values.

Model	R <sup>2</sup>	ARD (%)	AARD (%)	RMSE	RMSD (%)	MAPE
Present study (PGXGB)	<b>0.992 248</b>	<b>0.032 14</b>	<b>0.027 049</b>	<b>0.055 934</b>	<b>0.055 934</b>	<b>2.704 91</b>
Gradient Boosting	0.983 74	0.051 599	0.044 929	0.081 008	0.081 008	4.492 88
Extra Trees	0.982 744	0.049 044	0.039 79	0.083 451	0.083 451	3.978 99
v-SVR	0.923 893	0.076 083	0.054 942	0.175 255	0.175 255	5.494 22

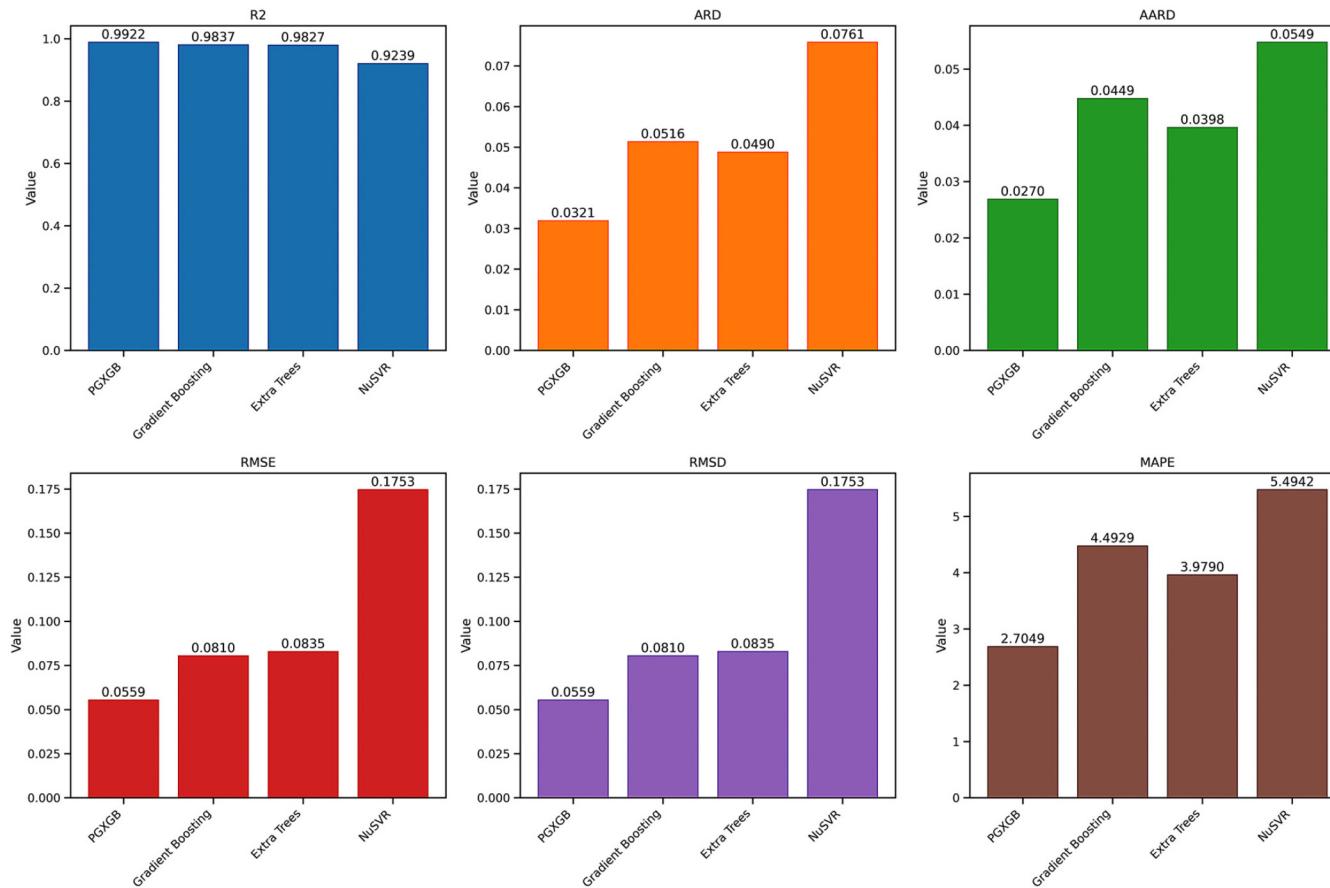
The identification of the best model among the choices is greatly aided by these statistical metrics.

Important statistical metrics including R<sup>2</sup>, ARD, AARD, RMSE, RMSD, and MAPE are compared among the proposed PGXGB model and a number of other soft computing models, i.e., Gradient Boosting, Extra Tree, and v-SVR, which are shown in **Table V**, with the highest R<sup>2</sup> (0.992 248) value and the lowest AARD and RMSE values. The table demonstrates the current model's higher efficiency. These results demonstrate that the suggested PGXGB model performs more accurately than other current machine learning models. **Figure 14** shows the statistical indices

comparison R<sup>2</sup>, ARD, AARD, RMSE, RMSD, and MAPE for soft computing algorithm.

#### E. Statistical metrics comparisons of PGXGB model with other theoretical and empirical models

Important statistical metrics including R<sup>2</sup>, ARD, AARD, and RMSD are compared between the proposed PGXGB model and a number of other theoretical and empirical models as shown in **Table VI**. With the highest R<sup>2</sup> value and the lowest AARD and RMSD values, the table demonstrates the PGXGB model's higher

**FIG. 14.** Comparison of statistical metrics R<sup>2</sup>, ARD, AARD, RMSE, RMSD, MAPE for soft computing algorithm.

18 October 2025 14:46:06

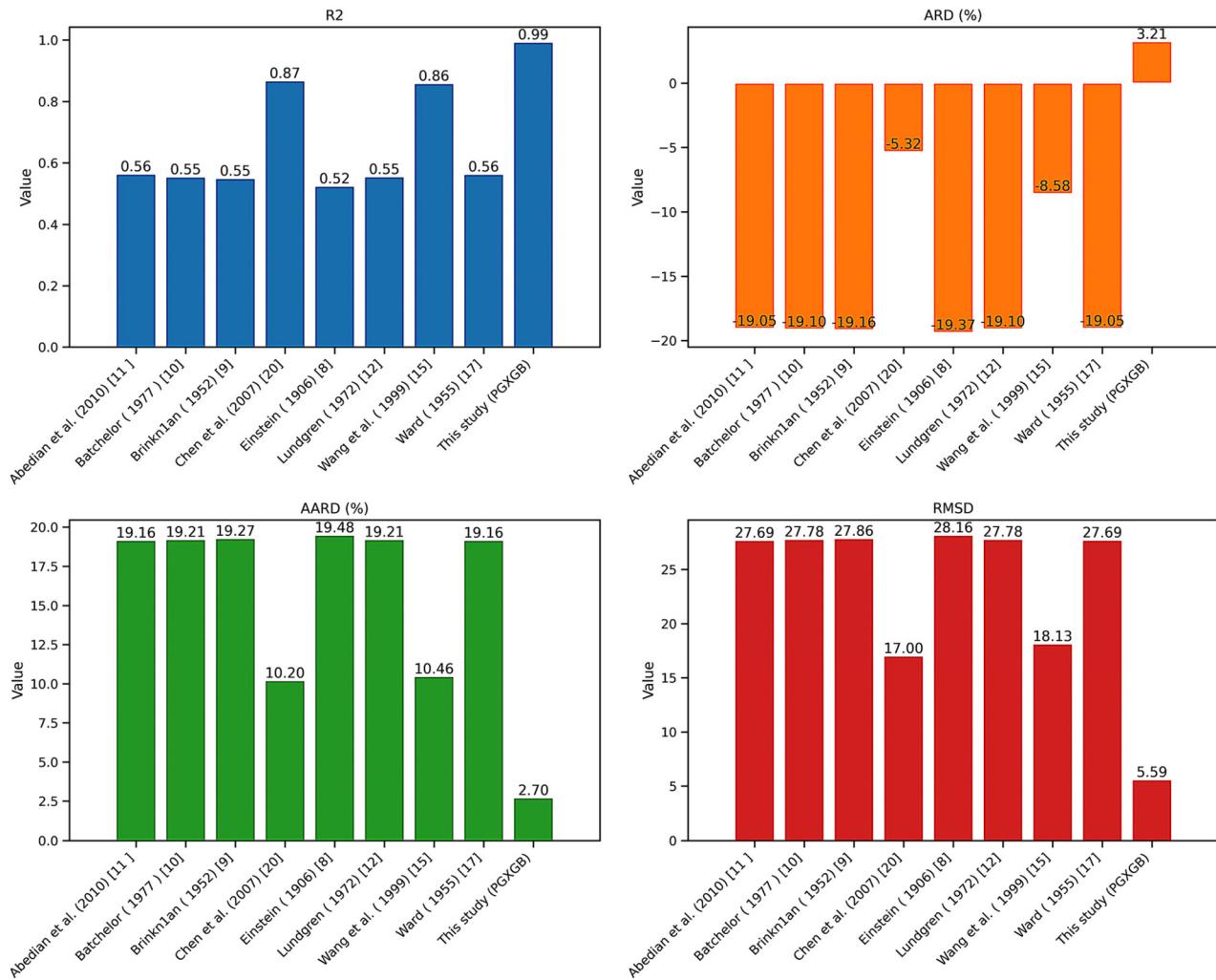
**TABLE VI.** Comparisons of various theoretical and empirical models using statistical metrics.

Model	R <sup>2</sup>	ARD (%)	AARD (%)	RMSD
Abedian and Kachanov <sup>17</sup>	0.5636	-19.05	19.157	27.6893
Batchelor <sup>14</sup>	0.5542	-19.1035	19.2104	27.7781
Brinkman <sup>11</sup>	0.549	-19.1639	19.2707	27.8565
Chen <i>et al.</i> <sup>16</sup>	0.8667	-5.315	10.2031	17.0028
Einstein <sup>7</sup>	0.5242	-19.3745	19.4811	28.1551
Lundgren <sup>13</sup>	0.5544	-19.1013	19.2082	27.7751
Wang <i>et al.</i> <sup>15</sup>	0.8581	-8.5764	10.464	18.1291
Ward <sup>12</sup>	0.563	-19.0522	19.1592	27.6936
Present study (PGXGB)	<b>0.992 248</b>	<b>3.213 96</b>	<b>2.704 91</b>	<b>5.593 36</b>

efficiency. As a result, **Table VI** shows that the proposed PGXGB model performs more accurately than other current models. **Figure 15** graphically displays this comparison of statistical indices R<sup>2</sup>, ARD, AARD, and RMSD for both theoretical and empirical models.

#### IV. SENSITIVITY ANALYSIS

Various studies have shown that different factors affect the prediction of nanofluid viscosity. A sensitivity analysis was conducted using the proposed PGXGB model to identify that the particular input variables have the most significant impact on prediction accuracy. The results in **Table VII** indicate that nanoparticle volume fraction ( $\varphi$ ) has the highest influence among single variables, with the highest R<sup>2</sup> (0.665 028) and lowest RMSE (0.367 675) and AARD (26.8727). For two variable combinations,

**FIG. 15.** Comparison of statistical metrics R<sup>2</sup>, ARD, AARD, RMSD for theoretical and empirical models.

**TABLE VII.** Sensitivity analysis using a combination of variables.

Test no.	Variables combination	Input variables	RMSE	R-Squared	AARD
1		d	0.601 238	0.104 277	44.2945
2		<b>Phi</b>	<b>0.367 675</b>	<b>0.665 028</b>	<b>26.8727</b>
3	Single variable	Temp	0.604 996	0.093 045	39.014
4		Base viscosity	0.602 31	0.101 079	38.1475
5		d, phi	0.345 661	0.703 938	25.7843
6		d, temp	0.581 146	0.163 143	32.5694
7	Combination of two variables	d, base viscosity	0.577 668	0.173 13	32.6766
8		<b>Phi, temp</b>	<b>0.181 727</b>	<b>0.918 169</b>	<b>8.161 39</b>
9		Phi, base viscosity	0.202 558	0.898 333	8.289 62
10		Temp, base viscosity	0.607 007	0.087 005	38.6951
11		d, phi, temp	0.058 521	0.991 514	2.745 73
12	Combination of three variables	<b>d, phi, base viscosity</b>	<b>0.058 235</b>	<b>0.991 597</b>	<b>2.607 63</b>
13		d, temp, base viscosity	0.582 118	0.160 342	32.825
14		Phi, temp, base viscosity	0.200 91	0.899 981	7.819 62
15	Combination of all the variables	<b>d, phi, temp, base viscosity</b>	<b>0.055 934</b>	<b>0.992 248</b>	<b>2.704 91</b>

the highest  $R^2$  (0.918 169) was achieved with phi and temperature. Combining three variables (nanoparticle size, volume fraction, and base fluid viscosity) resulted in  $R^2$  (0.991 597), RMSE (0.058 235 1), and AARD (2.607 63). Adding temperature as a fourth variable increased  $R^2$  to (0.992 248). Figure 16 shows the graphical outlook of sensitivity analysis.

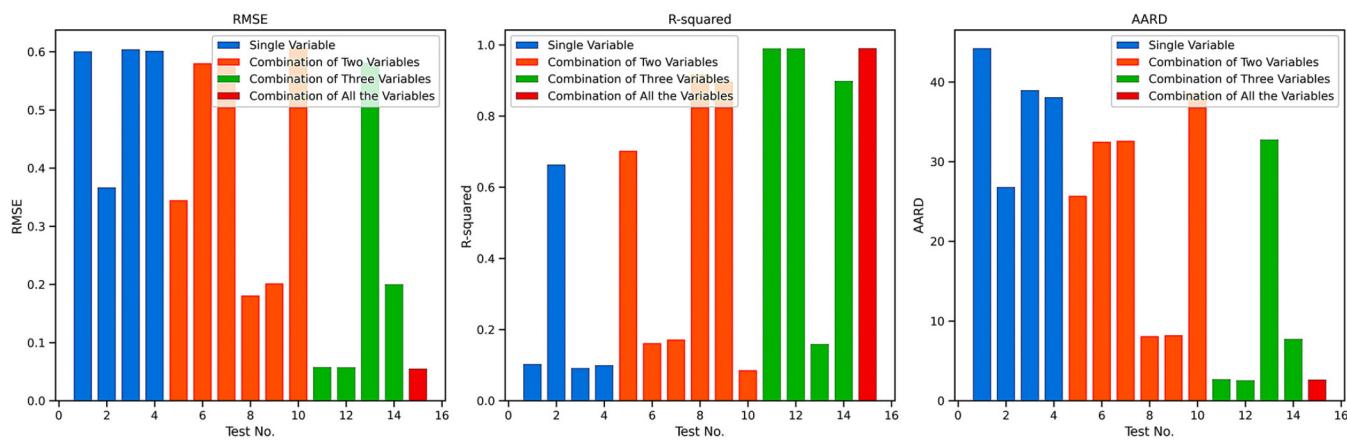
In the sensitivity analysis, volume concentration was identified as the most critical parameter influencing nanofluid viscosity due to its direct, nonlinear, and dominant impact on fluid behavior. An increase in the nanoparticle volume fraction directly adds more solid particles, enhancing particle-particle and particle-fluid interactions. This raises internal flow resistance, leading to higher viscosity due to increased shear forces. The relationship is nonlinear because higher concentrations promote particle agglomeration, which disproportionately increases viscosity. This variable also significantly impacts other thermophysical properties, including

density, thermal conductivity, and specific heat, playing a crucial role in nanofluid modeling. While temperature and nanoparticle size also affect viscosity, their influence is constrained by volume fraction. For example, temperature's ability to reduce viscosity diminishes at low volume fractions, and the effects of nanoparticle size (where smaller particles increase the surface area for interaction) remain secondary to volume fraction itself. Thus, nanoparticle volume fraction was found to be the most influential variable affecting viscosity variability among the 792 data points analyzed.

18 October 2025 144606

### A. Applications for this insight

This understanding has significant practical implications. In nanofluid optimization, engineers can fine-tune nanoparticle volume fractions to balance viscosity with thermal performance and pumping power requirements. For heat transfer systems,

**FIG. 16.** Sensitivity analysis using the combination of variables.

optimizing volume fractions enhances thermal conductivity while avoiding excessive viscosity that could hinder flow efficiency. Industrial applications in fields such as lubrication, printing, and biomedical engineering can leverage this insight to adjust fluid properties for specific requirements, such as achieving shear-thinning behavior or stability at high concentrations. Additionally, these findings guide the development of predictive models and empirical correlations that prioritize nanoparticle volume fraction, improving accuracy for diverse nanofluid formulations. In scale-up studies, this knowledge ensures that laboratory results can be effectively translated to industrial scales with reproducible fluid characteristics.

### B. Non-linear relationships in nanoparticle size and temperature

The relationship between nanoparticle size and viscosity is nonlinear, influenced by factors such as surface area-to-volume ratio, boundary layer effects, and cluster formation. Smaller particles typically increase viscosity due to enhanced interactions with the base fluid, although aggregation and Brownian motion can alter this effect. Larger nanoparticles can also aggregate, adding further complexity. Similarly, temperature affects viscosity non-linearly. Viscosity decreases exponentially with increasing temperature, but this effect saturates at higher temperatures. At extreme temperatures, thermal expansion and particle dynamics further contribute to this nonlinear behavior. While the PGXGB model captures many complex relationships, it may struggle with data coverage at extreme sizes or temperatures and assumes additive interactions between variables, potentially missing higher-order nonlinearities. To better explore these nonlinearities, future studies could conduct sensitivity analyses, incorporate interaction terms, apply transformations, and combine PGXGB with physics-informed corrections. Expanding the dataset would also improve generalization.

## V. CONCLUSION

The current research presents a new method for predicting water-based nanofluid viscosity, which includes alumina, copper oxide, silicon dioxide, and titanium dioxide. It does this by employing a physics-guided XGBoost model (PGXGB). The loss function, which is determined based on physical constraints, is integrated into the XGBoost algorithm to develop the PGXGB model. The proposed PGXGB model provides accurate predictions. The PGXGB predicts nanofluid viscosity across a range of parameters with high accuracy from base fluid viscosity, temperature, size, and nanoparticle volume fraction as inputs. The incorporation of physical concepts yields a solution that is both efficient and consistent. The present research allows for the establishment of the following conclusions:

1. Using a variety of operating processes, the viscosity of several water-based nanofluids ( $\text{Al}_2\text{O}_3$ ,  $\text{CuO}$ ,  $\text{TiO}_2$ , and  $\text{SiO}_2$ ) is estimated using the suggested PGXGB model. With a prediction accuracy of over 99%, it surpasses both empirical and theoretical correlations as well as all conventional computer-aided models in predicting the viscosity of nanofluids.

2. From a statistical perspective, the PGXGB model's prediction validation for the dataset is justified. The data can be easily predicted by the suggested PGXGB model, and it agrees extremely well with the original experimental results.
3. According to Sec. III D, this suggested PGXGB model outperforms all other machine learning models, including  $\nu$ -SVM, Gradient Boosting Regressor, and Extra Tree Regressor, in terms of accuracy in predicting the viscosity of nanofluids. These comparison findings show that the suggested PGXGB model produces more accurate predictions results with ( $\text{RMSE} = 0.0559$  336,  $R^2 = 0.992$  248)
4. This model is more practical than any other conventional model and also has benefits over classical techniques since it is guided by physics laws. Additionally, it offers a powerful model for viscosity prediction with improved generalization abilities.
5. The PGXGB model uses the regularization parameter to address the overfitting problem during training.
6. Sensitivity analysis is also adopted for determining the input factors that have the greatest impact on prediction accuracy. Among single variables, nanoparticle volume fraction has the largest influence.
7. The validity of the experimental results is confirmed by both statistical and graphical analysis. It works better than earlier models, especially under difficult circumstances such as high temperatures, concentrations of nanoparticles, and viscosity levels.

## VI. LIMITATIONS AND FUTURE SCOPE

The PGXGB framework developed in this study is intended as a data-driven predictive tool for nanofluid viscosity within the experimental range considered. While the model incorporates soft physics-based constraints to maintain physically consistent trends, it remains a black-box machine learning method and does not explicitly resolve the underlying particle-scale mechanisms. As such, detailed microstructural phenomena, such as nanoparticle clustering, agglomeration dynamics, or base-fluid molecular interactions, cannot be directly inferred. Any physical insights from PGXGB outputs are, therefore, limited to observing broad trends rather than providing mechanistic explanations.

The present framework treats nanoparticle type primarily as a categorical variable and does not explicitly decompose its influence in terms of measurable physical characteristics such as zeta potential, surface energy, or synthesis method. While these factors can significantly affect particle-fluid interactions and, thus, viscosity, incorporating them would require detailed and consistent experimental reporting across datasets, which was not available for the materials considered in this work. As such, the current model captures nanoparticle effects indirectly through the available descriptors, and more explicit physical decomposition remains a direction for future research.

This work is restricted to four oxide nanoparticles ( $\text{Al}_2\text{O}_3$ ,  $\text{TiO}_2$ ,  $\text{SiO}_2$ , and  $\text{CuO}$ ) dispersed in water, chosen for their practical relevance and the availability of high-quality data. Application of the framework to other base fluids (e.g., ethylene glycol or oils) or to different nanoparticle chemistries would require retraining with

datasets that capture the thermophysical properties and particle-fluid interaction characteristics of those systems. While the PGXGB approach is adaptable, its predictive accuracy depends on the representativeness and quality of the training data. These considerations are identified as areas for future research.

For current research, the dataset consisted predominantly of Newtonian measurements at low shear rates (where viscosity is shear-independent); we have taken a dataset that is not considering variables like the shear rate or nanoparticle size, so we have not discussed these variable physical behaviors; we may consider and use these variables for future work, and also in future work, we will incorporate an **explicit physical decomposition of nanoparticle type** (e.g., incorporating zeta potential, surface energy, or synthesis method) that would enrich the model.

We have taken datasets that do not consider variables like the shear rate or nanoparticle size, so we have not discussed these variable physical behaviors; we may consider and use these variables for future work.

## NOMENCLATURE

### General characters

- d = particle size (nm)  
 $\mu_{bf}$  = base fluid viscosity (mPa s)  
 $\mu_{nf}$  = nanofluid viscosity (mPa s)  
 $\varphi$  = volume fraction (%)  
T = temperature (°C)  
 $v$  = in SVR controls number of support vectors and margin of error

### Short forms

- ARD = absolute relative deviation  
AARD = average absolute relative

### Deviation

- $Al_2O_3$  = aluminum oxide  
 $CuO$  = copper oxide  
ET = Extra Tree  
GBR = Gradient Boost Regressor  
KKT = Karush–Kuhn–Tucker  
MAE = mean absolute error  
MAPE = mean absolute percentage error  
MSE = mean square error  
PGXGB = physics-guided XGBoost  
 $R^2$  = coefficient of determination  
RMSD = root mean square deviation  
RMSE = root mean square error  
 $SiO_2$  = silicon dioxide  
 $TiO_2$  = titanium dioxide  
 $v$ -SVR: =  $v$ -Support Vector Regression

### Subscripts

- bf = base fluid  
nf = nanofluids  
np = nanoparticle

## AUTHOR DECLARATIONS

### Conflict of Interest

The authors have no conflicts to disclose.

### Author Contributions

**Shekhar:** Conceptualization (equal); Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Resources (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Koj Sambyo:** Project administration (equal); Supervision (equal). **Seema Tinker:** Validation (equal).

## DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

## REFERENCES

- <sup>1</sup>B. C. Pak and Y. I. Cho, "Hydrodynamic and heat transfer study of dispersed fluids with submicron metallic oxide particles," *Exp. Heat Transfer* **11**, 151–170 (1998).  
<sup>2</sup>T. R. Barrett, S. Robinson, K. Flinders, A. Sergis, and Y. Hardalupas, "Investigating the use of nanofluids to improve high heat flux cooling systems," *Fusion Eng. Des.* **88**, 2594–2597 (2013).  
<sup>3</sup>A. Kasaeian, A. T. Eshghi, and M. Sameti, "A review on the applications of nanofluids in solar energy systems," *Renewable Sustainable Energy Rev.* **43**, 584–598 (2015).  
<sup>4</sup>H. D. Koca, S. Doganay, A. Turgut, I. H. Tavman, R. Saidur, and I. M. Mahbubul, "Effect of particle size on the viscosity of nanofluids: A review," *Renewable Sustainable Energy Rev.* **82**, 1664–1674 (2018).  
<sup>5</sup>M. H. Ahmadi, M. Sadeghzadeh, H. Maddah, A. Solouk, R. Kumar, and K. wing Chau, "Precise smart model for estimating dynamic viscosity of  $SiO_2$ /ethylene glycol–water nanofluid," *Eng. Appl. Comput. Fluid Mech.* **13**, 1095–1105 (2019).  
<sup>6</sup>M. Dadhich, Shekhar, K. Sambyo, V. Sharma, and G. Jain, "Nusselt number estimation using a GBR-GSO-based machine learning predictive model in alumina and titania nanofluids in a boiling process," *J. Therm. Anal. Calorim.* **148**, 14225–14242 (2023).  
<sup>7</sup>A. Einstein, "Eine neue Bestimmung der Moleküldimensionen," *Ann. Phys.* **324**, 289–306 (1906).  
<sup>8</sup>N. A. Frankel and A. Acrivos, "On the viscosity of a concentrated suspension of solid spheres," *Chem. Eng. Sci.* **22**, 847–853 (1967).  
<sup>9</sup>I. M. Krieger and T. J. Dougherty, "A mechanism for Non-Newtonian flow in suspensions of rigid spheres," *Trans. Soc. Rheol.* **3**, 137–152 (1959).  
<sup>10</sup>D. P. Kulkarni, D. K. Das, and G. A. Chukwu, "Temperature dependent rheological property of copper oxide nanoparticles suspension (nanofluid)," *J. Nanosci. Nanotechnol.* **6**, 1150–1154 (2006).  
<sup>11</sup>H. C. Brinkman, "The viscosity of concentrated suspensions and solutions," *J. Chem. Phys.* **20**, 571 (1952).  
<sup>12</sup>S. G. Ward, "Oil and colour chemists' association," *Nature* **175**, 240 (1955).  
<sup>13</sup>T. S. Lundgren, "Slow flow through stationary random beds and suspensions of spheres," *J. Fluid Mech.* **51**, 273–299 (1972).  
<sup>14</sup>G. K. Batchelor, "The effect of Brownian motion on the bulk stress in a suspension of spherical particles," *J. Fluid Mech.* **83**, 97–117 (1977).  
<sup>15</sup>X. Wang, X. Xu, and S. U. S. Choi, "Thermal conductivity of nanoparticle–fluid mixture," *J. Thermophys. Heat Transfer* **13**, 474–480 (1999).  
<sup>16</sup>H. Chen, Y. Ding, Y. He, and C. Tan, "Rheological behaviour of ethylene glycol based titania nanofluids," *Chem. Phys. Lett.* **444**, 333–337 (2007).

- <sup>17</sup>B. Abedian and M. Kachanov, "On the effective viscosity of suspensions," *Int. J. Eng. Sci.* **48**, 962–965 (2010).
- <sup>18</sup>L. Godson, B. Raja, D. M. Lal, and S. Wongwises, "Experimental investigation on the thermal conductivity and viscosity of silver-deionized water nanofluid," *Exp. Heat Transfer* **23**, 317–332 (2010).
- <sup>19</sup>M. Gholizadeh, M. Jamei, I. Ahmadianfar, and R. Pourrajab, "Prediction of nanofluids viscosity using random forest (RF) approach," *Chemom. Intell. Lab. Syst.* **201**, 104010 (2020).
- <sup>20</sup>M. J. Pastoriza-Gallego, C. Casanova, J. L. Legido, and M. M. Piñeiro, "CuO in water nanofluid: Influence of particle size and polydispersity on volumetric behaviour and viscosity," *Fluid Phase Equilib.* **300**, 188–196 (2011).
- <sup>21</sup>A. Baghban, A. Jalali, M. Shafiee, M. H. Ahmadi, and K. wing Chau, "Developing an ANFIS-based swarm concept model for estimating the relative viscosity of nanofluids," *Eng. Appl. Comput. Fluid Mech.* **13**, 26–39 (2019).
- <sup>22</sup>I. M. Mahbubul, R. Saidur, and M. A. Amalina, "Latest developments on the viscosity of nanofluids," *Int. J. Heat Mass Transfer* **55**, 874–885 (2012).
- <sup>23</sup>M. H. Ahmadi, M. A. Ahmadi, M. A. Nazari, O. Mahian, and R. Ghasempour, "A proposed model to predict thermal conductivity ratio of Al<sub>2</sub>O<sub>3</sub>/EG nanofluid by applying least squares support vector machine (LSSVM) and genetic algorithm as a connectionist approach," *J. Therm. Anal. Calorim.* **135**, 271–281 (2019).
- <sup>24</sup>S. A. Abdollahi, A. Alizadeh, i. C. Esfahani, M. Zarinfar, and P. Pasha, "Investigating heat transfer and fluid flow betwixt parallel surfaces under the influence of hybrid nanofluid suction and injection with numerical analytical technique," *Alexandria Eng. J.* **70**, 423–439 (2023).
- <sup>25</sup>I. Chiniforooshan Esfahani, "A data-driven physics-informed neural network for predicting the viscosity of nanofluids," *AIP Adv.* **13**, 025206 (2023).
- <sup>26</sup>A. Karimpour, S. Ghasemi, M. H. K. Darvanooghi, and A. Abdollahi, "A new correlation for estimating the thermal conductivity and dynamic viscosity of CuO/liquid paraffin nanofluid using neural network method," *Int. Commun. Heat Mass Transfer* **92**, 90–99 (2018).
- <sup>27</sup>F. Ajila, S. Manokaran, K. Ramaswamy, D. Thiagarajan, P. Pappula, A. Shaik, S. Dillibabu, U. Kasi, and M. Selvaraju, "Prediction of nanofluid thermal conductivity and viscosity with machine learning and molecular dynamics," *Therm. Sci.* **28**, 717–729 (2024).
- <sup>28</sup>X. Dai, H. T. Andani, A. Alizadeh, A. M. Abed, G. F. Smaisim, S. K. Hadrawi, M. Karimi, M. Shamsborhan, and D. Toghraie, "Using Gaussian process regression (GPR) models with the Matérn covariance function to predict the dynamic viscosity and torque of SiO<sub>2</sub>/ethylene glycol nanofluid: A machine learning approach," *Eng. Appl. Artif. Intell.* **122**, 106107 (2023).
- <sup>29</sup>M. H. Ahmadi, F. Hajizadeh, M. Rahimzadeh, M. B. Shafii, A. J. Chamkha, G. Lorenzini, and R. Ghasempour, "Application GMDH artificial neural network for modeling of Al<sub>2</sub>O<sub>3</sub>/water and Al<sub>2</sub>O<sub>3</sub>/ethylene glycol thermal conductivity," *Int. J. Heat Technol.* **36**, 773–782 (2018).
- <sup>30</sup>E. Heidari, M. A. Sobati, and S. Movahedirad, "Accurate prediction of nanofluid viscosity using a multilayer perceptron artificial neural network (MLP-ANN)," *Chemom. Intell. Lab. Syst.* **155**, 73–85 (2016).
- <sup>31</sup>N. Zhao, X. Wen, J. Yang, S. Li, and Z. Wang, "Modeling and prediction of viscosity of water-based nanofluids by radial basis function neural networks," *Powder Technol.* **281**, 173–183 (2015).
- <sup>32</sup>M. Hemmat Esfe, M. R. Hassani Ahangar, M. Rejvani, D. Toghraie, and M. H. Hajmohammad, "Designing an artificial neural network to predict dynamic viscosity of aqueous nanofluid of TiO<sub>2</sub> using experimental data," *Int. Commun. Heat Mass Transfer* **75**, 192–196 (2016).
- <sup>33</sup>S. O. Giwa, M. Sharifpur, M. Goodarzi, H. Alsulami, and J. P. Meyer, "Influence of base fluid, temperature, and concentration on the thermophysical properties of hybrid nanofluids of alumina–ferrofluid: Experimental data, modeling through enhanced ANN, ANFIS, and curve fitting," *J. Therm. Anal. Calorim.* **143**, 4149–4167 (2021).
- <sup>34</sup>M. K. Meybodi, S. Naseri, A. Shokrollahi, and A. Daryasafar, "Prediction of viscosity of water-based Al<sub>2</sub>O<sub>3</sub>, TiO<sub>2</sub>, SiO<sub>2</sub>, and CuO nanofluids using a reliable approach," *Chemom. Intell. Lab. Syst.* **149**, 60–69 (2015).
- <sup>35</sup>H. Adun, D. Kavaz, I. Wole-Osho, and M. Dagbasi, "Synthesis of Fe<sub>3</sub>O<sub>4</sub>-Al<sub>2</sub>O<sub>3</sub>-ZnO/water ternary hybrid nanofluid: Investigating the effects of temperature, volume concentration and mixture ratio on specific heat capacity, and development of hybrid machine learning for prediction," *J. Energy Storage* **41**, 102947 (2021).
- <sup>36</sup>M. Jamei, I. Ahmadianfar, I. A. Olumegbon, M. Karbasi, and A. Asadi, "On the assessment of specific heat capacity of nanofluids for solar energy applications: Application of Gaussian process regression (GPR) approach," *J. Energy Storage* **33**, 102067 (2021).
- <sup>37</sup>I. O. Alade, M. A. Abd Rahman, Z. Abbas, Y. Yaakob, and T. A. Saleh, "Application of support vector regression and artificial neural network for prediction of specific heat capacity of aqueous nanofluids of copper oxide," *Sol. Energy* **197**, 485–490 (2020).
- <sup>38</sup>M. Jamei, I. Ahmadianfar, I. A. Olumegbon, A. Asadi, M. Karbasi, Z. Said, M. Sharifpur, and J. P. Meyer, "On the specific heat capacity estimation of metal oxide-based nanofluid for energy perspective—A comprehensive assessment of data analysis techniques," *Int. Commun. Heat Mass Transfer* **123**, 105217 (2021).
- <sup>39</sup>L. Shi, S. Zhang, A. Arshad, Y. Hu, Y. He, and Y. Yan, "Thermo-physical properties prediction of carbon-based magnetic nanofluids based on an artificial neural network," *Renewable Sustainable Energy Rev.* **149**, 111341 (2021).
- <sup>40</sup>M. A. Hassan and D. Banerjee, "A soft computing approach for estimating the specific heat capacity of molten salt-based nanofluids," *J. Mol. Liq.* **281**, 365–375 (2019).
- <sup>41</sup>M. Chandrasekar, S. Suresh, and A. Chandra Bose, "Experimental investigations and theoretical determination of thermal conductivity and viscosity of Al<sub>2</sub>O<sub>3</sub>/water nanofluid," *Exp. Therm. Fluid Sci.* **34**, 210–216 (2010).
- <sup>42</sup>L. Fedele, L. Colla, and S. Bobbo, "Viscosity and thermal conductivity measurements of water-based nanofluids containing titanium oxide nanoparticles," *Int. J. Refrig.* **35**, 1359–1366 (2012).
- <sup>43</sup>K. B. Anoop, S. Kabelac, T. Sundararajan, and S. K. Das, "Rheological and flow characteristics of nanofluids: Influence of electroviscous effects and particle agglomeration," *J. Appl. Phys.* **106**, 034909 (2009).
- <sup>44</sup>C. T. Nguyen, F. Desgranges, N. Galanis, G. Roy, T. Maré, S. Boucher, and H. Angue Mintsa, "Viscosity data for Al<sub>2</sub>O<sub>3</sub>-water nanofluid-hysteresis: Is heat transfer enhancement using nanofluids reliable?," *Int. J. Therm. Sci.* **47**, 103–111 (2008).
- <sup>45</sup>J. H. Lee, K. S. Hwang, S. P. Jang, B. H. Lee, J. H. Kim, S. U. S. Choi, and C. J. Choi, "Effective viscosities and thermal conductivities of aqueous nanofluids containing low volume concentrations of Al<sub>2</sub>O<sub>3</sub> nanoparticles," *Int. J. Heat Mass Transfer* **51**, 2651–2656 (2008).
- <sup>46</sup>H. Masuda, A. Ebata, K. Teramae, and N. Hishinuma, "Alteration of thermal conductivity and viscosity of liquid by dispersing ultra-fine particles. Dispersion of Al<sub>2</sub>O<sub>3</sub>, SiO<sub>2</sub> and TiO<sub>2</sub> ultra-fine particles," *Netsu Bussei* **7**, 227–233 (1993).
- <sup>47</sup>A. Turgut, I. Tavman, M. Chirtoc, H. P. Schuchmann, C. Sauter, and S. Tavman, "Thermal conductivity and viscosity measurements of water-based TiO<sub>2</sub> nanofluids," *Int. J. Thermophys.* **30**, 1213–1226 (2009).
- <sup>48</sup>I. Tavman, A. Turgut, M. Chirtoc, H. Karbstein, and S. Tavman, "Experimental investigation of viscosity and thermal conductivity of suspensions containing nanosized ceramic particles," *Arch. Mater. Sci. Eng.* **34**, 99–104 (2008).
- <sup>49</sup>W. Duangthongsuk and S. Wongwises, "Measurement of temperature-dependent thermal conductivity and viscosity of TiO<sub>2</sub>-water nanofluids," *Exp. Therm. Fluid Sci.* **33**, 706–714 (2009).
- <sup>50</sup>E. V. Timofeeva, A. N. Gavrilov, J. M. McCloskey, Y. V. Tolmachev, S. Sprunt, L. M. Lopatina, and J. V. Selinger, "Thermal conductivity and particle agglomeration in alumina nanofluids: Experiment and theory," *Phys. Rev. E* **76**, 061203 (2007).
- <sup>51</sup>W. Duangthongsuk and S. Wongwises, "An experimental study on the heat transfer performance and pressure drop of TiO<sub>2</sub>-water nanofluids flowing under a turbulent flow regime," *Int. J. Heat Mass Transfer* **53**, 334–344 (2010).
- <sup>52</sup>M. J. Pastoriza-Gallego, C. Casanova, R. Páramo, B. Barbés, J. L. Legido, and M. M. Piñeiro, "A study on stability and thermophysical properties (density and viscosity) of Al<sub>2</sub>O<sub>3</sub> in water nanofluid," *J. Appl. Phys.* **106**, 064301 (2009).

- <sup>53</sup>Y. He, Y. Jin, H. Chen, Y. Ding, D. Cang, and H. Lu, "Heat transfer and flow behaviour of aqueous suspensions of TiO<sub>2</sub> nanoparticles (nanofluids) flowing upward through a vertical pipe," *Int. J. Heat Mass Transfer* **50**, 2272–2281 (2007).
- <sup>54</sup>D. Kwek, A. Crivoi, and F. Duan, "Effects of temperature and particle size on the thermal property measurements of Al<sub>2</sub>O<sub>3</sub>—water nanofluids," *J. Chem. Eng. Data* **55**, 5690–5695 (2010).
- <sup>55</sup>C. T. Nguyen, F. Desgranges, G. Roy, N. Galanis, T. Maré, S. Boucher, and H. Angue Mintsas, "Temperature and particle-size dependent viscosity data for water-based nanofluids - hysteresis phenomenon," *Int. J. Heat Fluid Flow* **28**, 1492–1506 (2007).
- <sup>56</sup>T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Barcelona, Spain, 25–19 August 2016 (ACM, 2016), pp. 785–794.
- <sup>57</sup>Shekhar, K. Sambyo, R. P. Sharma, and S. R. Mishra, "Predicting nanofluid density in ethylene glycol-based oxide nanoparticles using machine learning approach: GBR-GSO models," *J. Therm. Anal. Calorim.* **150**, 4441–4458 (2025).
- <sup>58</sup>B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New support vector algorithms," *Neural Comput.* **12**, 1207–1245 (2000).
- <sup>59</sup>Shekhar, K. Sambyo, and S. K. Gupta, "Extra tree regressor and tree-structured Parzen estimator based machine learning model for predicting nanofluid's Nusselt number," *Eng. Res. Express* **7**, 015284 (2025).
- <sup>60</sup>Shekhar and K. Sambyo, "Predicting viscosity of multi-walled carbon nanotube/water nanofluids using Gaussian process regression and emperor penguin optimizer algorithm," *Eng. Res. Express* **7**, 015281 (2025).