PAPER • **OPEN ACCESS**

Enhancing thermal management systems: a machine learning and metaheuristic approach for predicting thermophysical properties of nanofluids

To cite this article: Aritra Saha *et al* 2024 *Eng. Res. Express* **6** 045537

Engineering Research Express

PAPER

Enhancing thermal management systems: a machine learning and metaheuristic approach for predicting thermophysical properties of nanofluids

Aritra Saha[1,*], **Ankan Basu**[2] and **Sumanta Banerjee**[3]

[1] Fakultät für Elektrotechnik und Informatik, Leibniz Universität Hannover, Hannover, Germany
[2] Department of Computer Science and Engineering, Jadavpur University, Kolkata, India
[3] Department of Mechanical Engineering, Heritage Institute of Technology, Kolkata, India
[*] Author to whom any correspondence should be addressed.

E-mail: aritra.saha@stud.uni-hannover.de, ankanb.cse.pg@jadavpuruniversity.in and sumanta.banerjee@heritageit.edu

Keywords: nanofluids, machine learning, heat transfer

OPEN ACCESS

RECEIVED
15 July 2024

REVISED
16 September 2024

ACCEPTED FOR PUBLICATION
9 October 2024

PUBLISHED
5 November 2024

Abstract

In thermal engineering, predicting nanofluid thermophysical properties is essential for efficient cooling systems and improved heat transfer. Traditional methods often fall short in handling complex datasets. This study leverages machine learning (ML) and metaheuristic algorithms to predict key nanofluid properties, such as specific heat capacity (SHC), thermal conductivity (TC), and viscosity. By utilizing Artificial Neural Networks (ANN), Support Vector Regression (SVR), Gradient Boosting (GB), and Linear Regression (LR), alongside metaheuristic models like Differential Evolution (DE) and Particle Swarm Optimization (PSO), we achieve superior prediction accuracy compared to traditional models. The integration of these computational techniques with empirical data demonstrates their effectiveness in capturing the complex dynamics of thermofluids. Our results validate the precision of ML and metaheuristic models in predicting nanofluid properties and underscore their potential as robust tools for researchers and practitioners in thermal engineering. This work paves the way for future exploration of ML algorithms in thermal management, marking a significant advancement in optimizing nanofluid applications in industry and research.

1. Introduction

The advancement of nanotechnology and the increasing complexity of technological devices have contributed to the development of highly effective cooling fluids. The thermophysical characteristics of traditional coolants such as oil, water, and ethylene glycol are inadequate for modern heat transfer applications [1]. In keeping with current technological requirements, there is a strong interest in finding efficient ways to enhance the heat transfer performance of conventional fluids [2, 3].

Nanofluids are colloidal suspensions of nanoparticles in a base fluid, whose complex behavior is influenced by the presence of nanoparticles. They have found various industrial applications such as microelectronics [4, 5], biomedical [6, 7], heat exchangers and solar energy [8], among others. The application of nanofluids in solar technology has garnered significant research interest in multiple domains, which include solar collectors, photovoltaic/thermal systems, solar water heaters, solar-geothermal combined cooling heating and power systems, solar thermoelectric devices, and so on [9–13]. The utilization of nanofluids in these applications is made possible by their advanced heat transfer properties. This feature warrants an in-depth study of their thermophysical properties, with the aim to facilitate the progress and development of nanofluid-supported technologies.

The thermophysical properties of nanofluids are crucial for various applications, particularly in heat transfer and thermal systems. These include thermal conductivity, viscosity, density, specific heat capacity, and thermal diffusivity [14, 15]. Thermal conductivity is a measure of a material's ability to conduct heat in the presence of

temperature gradients (transfer of thermal energy by the kinetic energy of the constituent particles like molecules, atoms, ions). Materials with high thermal conductivity, such as metals, are effective in transferring heat, whereas materials with low thermal conductivity, like rubber, are used as insulators [16]. Viscosity, on the other hand, represents the resistance of a fluid to flow, where fluids with high viscosity, such as vegetable oils and molasses, flow more slowly than those with low viscosity, such as water, and this property is crucial in applications ranging from lubrication to the food industry [17]. Specific heat capacity is the amount of thermal energy per unit mass required to raise the temperature of a substance by unit degree, a critical factor in understanding the thermal properties of substances and in designing energy systems and materials for thermal management [18].

The introduction of nanoparticles into traditional fluids results in improved thermal and transport properties of these dispersed systems, as a result of the combined effects of nanoparticles and base fluids [19]. Published research literature amply demonstrates that the inclusion of nano-sized solid particles in conventional carrier fluids can lead to notable enhancements in heat transfer rates, which primarily arises from an increase in the effective thermal conductivity [3, 20, 21]. When solid particles are added to the base fluid, various thermophysical properties of the system undergo changes, which include alterations in density, thermal conductivity, and specific heat capacity [2, 22]. The variations in these properties depend on various factors such as the concentration of solid particles, temperature, and characteristics of the base fluid [23, 24]. Typically, both viscosity and thermal conductivity of nanofluids increase with the rise in volume fraction of nanoparticles [25, 26]. In most cases, there exists an optimal concentration (of solid nanoparticles) due to the undesirable effect of increased dynamic viscosity, in spite of the benefits of higher thermal conductivity for efficient heat transfer [27, 28].

The essential properties of nanofluids for various applications include Thermal Conductivity (TC), diffusivity, viscosity, Specific Heat Capacity (SHC), heat transfer coefficient, and density [29]. These properties play a crucial role in determining the heat transfer capabilities and flow characteristics of nanofluids. Consequently, considerable research has been conducted in recent years to investigate the thermophysical properties of nanofluids. As experimental determination of the thermophysical properties is costly, several theoretical and correlation based models have been developed to predict thermophysical properties of nanofluids. However, these models are subjected to uncertainties and fail to accurate model the experimental results, which limit their applications in practical scenarios [30–32]. In recent years, ML and AI techniques have been explored as alternative, cost-effective and more accurate ways to model nanofluid thermophysical properties [29–31, 33].

Rostamian *et al* [34] presented an approach of using ANN to predict thermal conductivity of CuO-SWCNTs hybrid nanofluid. The ANN model showed superior performance when compared to the correlation based model proposed by the same authors.

Alade *et al* [30] demonstrated the versatility of SVR in predicting the TC enhancement of metal and metal oxide nanofluids. The SVR models, in this study, significantly outperformed theoretical models.

Sharma *et al* [35] presented an application of various ML algorithms like GBR, SVR, Decision Tree Regression to model the thermal conductivity of TiO_2/water nanofluids.

Ahmadi *et al* [36] presented a detailed study on the use of different ML algorithms such as multivariate polynomial regression, multivariate adaptive regression splines, ANN, Group Method of Data Handling and M5-tree, to predict the dynamic viscosity of a CuO/water nanofluid. Their study confirmed ANN to be the most reliable approach among the different algorithms used.

Kanti *et al* [37] aimed to model the viscosity of fly ash nanofluids and fly-ash-Cu hybrid nanofluids using novel algorithms like the Multi Gene Genetic Programming.

Dai *et al* [38] presented an application of different ML algorithms like the Gaussian Process Regression to model torque and dynamic viscosity of SiO_2/EG nanofluid.

Alade *et al* [39] applied SVR and ANN to model the SHC of CuO/water nanofluid, with nanoparticle volume fractions in range 0.4% to 2%, and fluid temperature in range 293-338 K. They also achieved superior performance compared to theoretical models.

Zhang and Xu [40] explored the use of Gaussian Process Regression to predict the SHC of nanofluids with Al_2O_3 and CuO nanoparticles in EG and water basefluid. The model, applied to different range of temperatures and compositions, offered low-cost and quick estimations of nanofluid SHC.

Said *et al* [41] demonstrated the superior performance of ML models over traditional experimental and theoretical methods for prediction of SHC of metal oxide multiwall carbon nanotubes-water nanofluids. The authors used three ML models in their study: the XGBoost, the GPR, and the SVM. Among these, the XGBoost showed the best performance.

The various reviewed literature have endeavored to demonstrate the efficacy of ML methods over traditional theoretical and correlation-based models for specific thermophysical properties of some nanofluids. Building upon this foundation, our study aims to comprehensively explore a broad range of intelligence algorithms,

encompassing ML (ANN, SVM, LR, GB) as well as metaheuristic approaches (PSO, DE), applied to different nanofluids - both hybrid and non-hybrid. Our investigation focuses diverse thermophysical properties, including thermal conductivity, specific heat capacity, and viscosity, seeking to address the dual attributes of thermofluidic analysis in engineering applications - namely, heat transfer rate estimation and pumping power evaluation. Although prediction of heat transfer rates is the fundamental question in heat transfer applications, an estimation of pumping power or exergy payment required (to keep the stream flowing) is the key question in fluid mechanics as it applies to heat transfer science. This is because the pumping power required is directly related to the pressure drop, which is proportional to the drag force. The drag force, in turn, is determined by the effective viscosity of the working medium.

The result of these ML and metaheuristic models have been compared with the results from theoretical models, which in turn outline their superiority compared to the traditional theoretical models in the prediction of nanofluid thermophysical properties. These models demonstrate their potential as cost-effective alternatives for calculating the thermophysical properties of nanofluids in industrial and research applications.

2. Methodology

Machine Learning techniques have gained significant prominence due to their remarkable ability to solve both linear and non-linear tasks with high accuracy. These techniques are primarily data-driven. Over the years, numerous experiments have generated substantial amounts of experimental data, which can now be effectively utilized with ML techniques for classification and estimation.

In this experiment, SVR, GB, ANN, LR, DE and PSO methods have been employed to predict the SHC, TC and viscosity of nanofluids. These ML models were carefully chosen based on their unique strengths, each specifically adept at capturing intricate relationships within the data. SVR was selected for its robustness in handling non-linear data and its effectiveness in high-dimensional spaces, making it a suitable choice for modeling complex patterns in nanofluid properties. ANNs, with their layered structure, were deemed ideal for capturing intricate and non-linear interactions between nanofluid components. ANN also offers flexibility in terms of architecture and parameters, allowing for fine-tuning and achieving high levels of accuracy. GB, with its powerful ensemble-learning capability, was employed to construct a predictive model in a stage-wise fashion. This approach effectively reduces bias and variance, thereby enhancing the precision of predictions. LR was included as a baseline model to provide a straightforward understanding of the relationships within the data. It serves as a benchmark for evaluating the complexity of the other models. DE was chosen for its simplicity and its robustness in handling complex problems without needing the problem's gradient information. Similarly, PSO is chosen for its relative ease of understanding, high efficiency and its ability to escape from the local minima, and converge towards the global optimum solution of a problem.

The datasets underwent a normalization process before the models were trained for the ANN, GB, and SVR models, in order to achieve consistency and uniformity in the data representation. Z-Score normalisation was used as it is more robust to outliers compared to other methods such as min-max normalization. Equation (1) represents the equation of z-score normalization. Here x' is the normalized value, x represents the data point, $\bar{x}$ represents the mean and σ represents the standard deviation of the dataset.

$$x' = \frac{x - \bar{x}}{\sigma} \tag{1}$$

To optimize the hyperparameters of the SVR, GB, and ANN models, a Bayesian optimization algorithm was utilized during training, employing a ten-fold cross-validation approach and utilizing the MSE as the error metric. The hyperparameters can be found in tables 1–9. RMSE, Coefficient of Determination (R^2) and Lin's Concordance Correlation Coefficient (LCCC) were the chosen evaluation metric for all the models.

An overview of the fundamental characteristics of these methods is outlined below.

2.1. Support vector machines

The SVM is a powerful ML algorithm, which has gained significant popularity due to its ability to handle both linear and nonlinear classification and regression tasks effectively. It is particularly useful when dealing with complex datasets and finding optimal decision boundaries.

Rooted in statistical learning theory, the SVM employs the principle of structural risk minimization. This principal aims to minimize an upper bound of the generalization errors, in contrast to neural networks that primarily focus on minimizing prediction errors on the training data. The learning process of SVMs bears resemblance to solving linearly constrained quadratic programming problems. This feature ensures that the

solution obtained is unique, optimal, and global, and also less susceptible to over-fitting issues. Consequently, SVM is an appealing technique for applications that prioritize repeatability and consistency.

Initially, the SVM was developed for classification tasks; later it has been expanded to address regression problems by introducing an ε-insensitive loss function. The SVM works by attempting to find the best hyperplane that separates the different classes of data with the largest margin (in case of classification) or fits the regression data with the least margin (in case of regression) [42–44].

The SVM algorithm employs the use of kernel functions to transform the data into a higher-dimensional space without explicitly calculating the coordinates of the data points. This allows SVM to efficiently handle nonlinear relationships between the input features and the target variable [45].

When applied for solving regression problems, the above method is called the SVR. In this technique, our objective is to identify a function $f(x)$ that calculates the correlation between the input variables $X = x_1, x_2, x_3, \ldots, x_n; x_i \in R^n$ and the target values $Y = y_1, y_2, y_3, \ldots, y_n; y_i \in R$.To attain superior generalization performance in SVR, careful selection of its parameters (which includes C, ε, and the kernel function parameter) is crucial. Manual searching of the hyperparameter space is one approach to accomplish this task. However, this method is laborious and enhances possibilities of over-prediction errors.

Various extensions and variations of SVM have been developed over the years to address specific challenges and improve its performance. These include the use of different kernel functions such as linear, polynomial, radial basis function (RBF), and sigmoid kernels [46]. Additionally, techniques like SVR and Support Vector Clustering (SVC) have been derived from the original SVM formulation [47].

For the present study, a soft-margin SVR model with a linear kernel, selected through hyperparameter optimization using Bayesian Optimization, has been employed.

2.2. Gradient boosting machine

The GBM is another powerful ML technique. GBM belongs to the ensemble learning methods and is based on the concept of boosting, where weak learners are combined to create a strong learner. The fundamental idea is to iteratively train a sequence of weak learners (here, decision trees) in a stage-wise manner, with each subsequent learner correcting the mistakes made by the previous ones. Thus, by combining multiple weak learners, one can create a powerful ensemble model that can generalize well and provide robust predictions [48]. This iterative process minimizes a loss function by optimizing the residuals of the previous models [49]. The core principle of GBM lies in the gradient descent optimization algorithm, where the negative gradient of the loss function is used as a direction for updating the model parameters in each iteration [50]. This ensures that subsequent models focus on the areas where the previous models performed poorly, leading to improved overall performance.

GBM can be applied to solve various predictive modeling problems, including heat transfer and engineering applications of nanofluids. When applied in thermal applications of nanofluids, GBM initially consists of gathering relevant data, which may include experimental measurements or simulations of nanofluid properties (e.g. thermal conductivity, specific heats, heat transfer coefficients, and viscosity under different conditions; e.g., nanoparticle concentration, temperature, and fluid type). Next, preparation and preprocessing of data are done, where the data is cleaned, missing values are handled, and features are normalized (or scaled).The relevant features (related to nanoparticle size, shape, concentration, or surface properties) are then created that capture the physics of nanofluids and significantly impact nanofluid behavior. Next, the dataset is categorized into training, validation, and testing sets, which allows for training and tuning of hyperparameters, and evaluation of GBM model on separate datasets. A suitable GBM algorithm (e.g., Gradient Boosting Trees, XGBoost [48], LightGBM [51], or CatBoost [52]) is then selected (after experimentation with different algorithms), which is suitable for prediction of the best nanofluid grade suitable for a particular application. The hyperparameters are tuned to optimize the model's performance; key hyperparameters may include the number of trees (or boosting rounds), learning rate, maximum tree depth, and regularization parameters. With the hyperparameters finely tuned, the GBM model is trained on the training data. The model will iteratively fit decision trees to minimize the error (e.g., mean squared error) between the predicted and the actual nanofluid properties. Depending on a specific nanofluid application, one may need to interpret the model's predictions, where techniques like shapely additive explanations (SHAP) values or Partial Dependence Plots can help understand the role of individual features in nanofluid behavior. The model's performance is subsequently evaluated on the validation set through utilization of appropriate metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), or others relevant to specific nanofluid property prediction.

An assessment of the model's generalization performance can be done on the testing set to ensure it can make accurate predictions for new, unseen data. Finally, if the model performs well and meets desired requirements, it can be deployed for real-time predictions or decision support in nanofluid-related processes or applications.

2.3. Artificial neural networks

ANNs are a class of ML models inspired by the structure and function of the human brain. ANNs consist of interconnected artificial neurons that work together to process and learn from input data to make predictions or decisions. One of the key advantages (of ANNs) is their ability to learn and model complex patterns and non-linear relationships in data. ANNs are a versatile and powerful tool for solving various predictive modeling problems in nanofluid applications, and can be employed to predict and model various nanofluid properties and behaviors. However, this warrants domain knowledge in nanofluid physics and careful preprocessing of data to achieve accurate results.

The basic building block of an ANN is the artificial neuron, also known as a node or perceptron. Each neuron receives input signals, applies a mathematical transformation to them, and produces an output signal. The strength of the connections between neurons, represented by weights, determines the importance of each input in the overall computation. By incorporating multiple layers of interconnected neurons, ANNs can capture non-linear dependencies and make predictions on various types of tasks, including regression, classification, and pattern recognition.

ANNs can be applied in nanofluid-related investigations in the following way. At first, the data relevant to an application is gathered either through experimentation and/or simulation. This data often includes experimental measurements of nanofluid properties such as thermal conductivity, viscosity, heat transfer coefficients, and particle size, for different types of base fluid, (thermal) boundary conditions, and varying nanoparticle concentration. Then, the task of data preprocessing is done, which involves cleaning the data, handling missing values, and invoke normalization or scaling features as needed. Preparation and preprocessing is particularly important when working with datasets of hybrid nanofluids. As the next step, relevant features are created that capture the underlying physics of nanofluids, which could include features related to nanoparticle characteristics (e.g., size, shape, material properties), properties of base fluid (e.g., temperature, viscosity), and experimental conditions (e.g., pressure, shear rate). The (preprocessed) dataset is then divided into training, validation, and testing sets, which allows for the training of the ANN model, tuning the hyperparameters, and evaluation of its performance on separate datasets. Next, the architecture of a suitable neural network is designed. Literature reviews indicate use of feedforward neural network in nanofluid applications [53–56]. The number of layers, neurons in each layer, and the choice of activation functions should be determined through experimentation and optimization. An appropriate loss function, based on specific nanofluid property prediction tasks, is then defined; for example, the Mean Squared Error (MSE) is commonly used for regression problems. The neural network is then trained, based on training data. Back propagation and gradient descent algorithms are typically used to adjust the network's weights and biases to minimize the loss function. During training, the network adjusts the weights of its connections based on the provided input-output pairs, aiming to minimize the difference between predicted and target outputs. This optimization process is typically guided by a specific algorithm (e.g. Back propagation algorithm [57]), which calculates the gradients and updates the weights iteratively. Hyperparameters like the learning rate, batch size, and the number of training epochs are tuned to optimize the ANN's performance. Techniques like grid search or random search can aid in finding the best hyperparameter configuration. Thereafter, the performance of the ANN model is evaluated on the validation set. Use of appropriate metrics, such as Mean Absolute Error (MAE) or Mean Squared Error (MSE), can be made in order to assess how well the ANN generalizes to unseen data. In other words, assessment of the model's performance on the testing set ensure that it can make accurate predictions for new, unseen nanofluid data. Depending on the specific thermal application, one may need to interpret the ANN's predictions. Techniques like feature importance analysis or visualization of neural network activations can help understand the factors influencing nanofluid behavior. If the ANN model performs well and meets necessary requirements, it can be deployed for real-time predictions or decision support in nanofluid-assisted thermal transport processes or applications.

The present study employs the ANN for solving regression problems, and uses the MSE as the loss function to be minimized.

2.4. Linear regression

LR is a statistical technique, which finds use in nanofluid research to model and analyze the relationship between different variables, particularly when studying the properties and behavior of nanofluids [58–60]. It serves as a foundation for more advanced regression techniques and ML algorithms. The LR tool can be applied in nanofluid research in the following contexts. For example, LR can be applied when investigating the thermal conductivity of nanofluids [61]. Empirical models can be developed with thermal conductivity data, which relate thermal conductivity to nanoparticle concentration, size, and other relevant parameters. In nanofluid experiments, LR can be used to analyze and interpret data; for instance, it can help determine whether there is a

linear correlation between two variables, which can provide insights into the underlying physical mechanisms governing nanofluid behavior. In addition, this analytical tool can be employed to assess the quality and consistency of nanofluid preparations. When developing theoretical models for nanofluid behavior, LR can be used to validate these models against experimental data [59]. For the purpose of comparative analysis, LR can be used for comparing the effects of different types of nanoparticles or base fluids on nanofluid properties [37].

Especially for the prediction of SHC of nanofluids, LR can be used to develop models that are based on various factors such as nanoparticle concentration, temperature, and fluid properties [62, 63]. Empirical relations between the SHC and these factors can be established by applying LR on experimental data. The resulting models can then be used to predict the SHC of nanofluids under different conditions [37, 62]. LR can also be used to quantify errors and uncertainties in experimental measurements, which is important for assessing the reliability and precision of experimental results in nanofluid research.

LR assumes a linear relationship between variables. It aims to find a linear equation that best describes the association between the variables and can be used for prediction and inference. It seeks to estimate the coefficients (intercept and slopes) of the linear equation that minimize the sum of squared differences between the observed and predicted values. However, in some cases, nanofluid behavior may exhibit nonlinearities, and more advanced regression techniques or ML methods may be necessary to capture these complex relationships.

2.5. Differential evolution

Differential Evolution (DE) has proved to be one of the most robust population-based optimization algorithms for most diverse, complex, multidimensional problems. Differential Evolution is actually a modern optimization algorithm proposed in 1995 by Storn and Price [64], which is very often applied in many scientific and engineering backgrounds.

DE operates like a fully evolutionary algorithm, but the key ideas of natural selection and genetics stem from Darwin's Evolution Theory. It starts with a population of candidate solutions, which changes over time due to operations like mutation, crossover, and selection. The key to distinguishing DE lies in its mutation policy: it takes the difference of two solutions randomly selected from the population, adds it to another solution to perturb it, and thereby creates a new candidate solution [65].

DE is better for very many causes attributed to the fact that it does have a balanced searching mechanism, which exploits the search space properly. It is effectively demonstrated over various optimization problems which are nonlinear, non-differentiable, and multimodal [66]. The algorithm has further been proven to be versatile in constrained structural optimization problems, when applied appropriately, were observed to perform remarkably [67].

In general, several variants of basic DE have been developed for the purpose of ensuring that it performs and adapts perfectly. Basically, the variants are different from one another in the mutation and crossover strategies along with the control parameters, which includes the mutation factor and crossover rate, setting and adapting. Over the years, in the process of evolution, a special stride has resulted in the self-adaptive DE (SaDE) algorithm, which brings dynamic changes to a strategy and parameter according to changes in search dynamics so that it can evolve to better optimization performance [68].

2.5.1. Particle swarm optimization

Another such technique is Particle Swarm Optimization (PSO), which is an optimization technique based on the notion of social behavior in animals, such as birds in a flock or fish in a school. PSO has been applied in large to wide scale fields in the solution of complex optimization problem.

PSO is based on a population of candidate solutions, called particles, moving around the search space according to simple mathematical formulae over the particle's position and velocity. Each particle is conducted in flight toward the best-known local positions for its locale, and also toward best-known positions in search space. The latter is updated as better positions are found by other particles.

The original concept of PSO was introduced by Kennedy and Eberhart, which laid the foundation for the algorithm and opened avenues for its application in optimization problems [69]. This was further studied by Shi and Eberhart, who conducted empirical studies highlighting the algorithm's rapid convergence capabilities, albeit noting a tendency to slow down as it approaches optimal solutions, indicating both the strength and a potential weakness in its mechanism [70].

The adaptability of PSO to different types of problems has been significant, as demonstrated by Coello *et al* [71], who extended PSO for multi-objective problems using Pareto dominance. This adaptation allows the algorithm to handle multiple competing objectives by maintaining a set of possible solutions, thus significantly enhancing its utility in complex scenarios.

Furthermore, Eberhart and Shi [72] have well elaborated on several applications, in addition to developments of PSO. They have pointed towards several developments in its applications, including changes in

its parameters like constriction factors and inertia weights that contribute to enhancement in performance as well as adaptability to change in environments. This underlines the evolving nature of PSO and its wide applicability.

Another landmark innovation in this direction is the Comprehensive Learning Particle Swarm Optimizer (CLPSO) proposed by Liang *et al* [73]. In this variant, the learning is organized on the historical best position of all other particles to update a particle's velocity. This approach is designed to maintain diversity within the particle swarm and avoid premature convergence, a common issue in optimization algorithms, thereby enhancing the PSO's ability to explore the search space effectively.

Recent systematic reviews highlight PSO's vast application range, from healthcare to environmental solutions and smart city applications. This underlines the versatility and ongoing relevance of the PSO algorithm in addressing modern-day challenges [74].

2.6. Hyperparameter optimization

In this work, the Bayesian Hyperparameter Optimization technique is used to optimize the hyperparameters for the SVR, ANN and GBM models. The hyperparameters for each of the models are shown in tables 1–9.

2.6.1. SHC

Table 1. Optimized hyperparameters for the proposed SHC SVR model.

SVR Hyperparameters	Values
C	141.88
epsilon	4.29e−03
kernel	linear

Table 2. Optimized hyperparameters for the proposed SHC ANN model.

ANN Hyperparameters	Values
batch_size	42
dropout_prob	2.94e−04
epochs	280
learning_rate	4.25e−02
nodes	391
num_hidden_layers	1
optimizer	Adamax

Table 3. Optimized hyperparameters for the proposed SHC GBM model.

GBM Hyperparameters	Values
learning_rate	8.89e−02
max_depth	5.0
n_estimators	89

2.6.2. TC

Table 4. Optimized hyperparameters for the proposed TC SVR model.

SVR Hyperparameters	Values
C	141.99
epsilon	7.54e−01
kernel	polynomial

Table 5. Optimized hyperparameters for the proposed TC ANN model.

ANN Hyperparameters	Values
batch_size	256
dropout_prob	1.01e−03
epochs	340
learning_rate	1.26e−04
nodes	494
num_hidden_layers	4
optimizer	Nadam

Table 6. Optimized hyperparameters for the proposed TC GBM model.

GBM Hyperparameters	Values
learning_rate	1.65e−01
max_depth	4
n_estimators	90

2.6.3. Viscosity

Table 7. Optimized hyperparameters for the proposed Viscosity SVR model.

SVR Hyperparameters	Values
C	190.57
epsilon	1.13e−03
kernel	rbf

Table 8. Optimized hyperparameters for the proposed Viscosity ANN model.

ANN Hyperparameters	Values
batch_size	37
dropout_prob	1.75e−03
epochs	280
learning_rate	1.81e−03
nodes	479
num_hidden_layers	3
optimizer	Nadam

Table 9. Optimized hyperparameters for the proposed Viscosity GBM model.

GBM Hyperparameters	Values
learning_rate	2.12e−01
max_depth	2
n_estimators	60

3. Dataset

In the development of an accurate ML model, a crucial decision involves selecting appropriate inputs for the algorithm. These inputs should effectively capture the property to be predicted, while being easily obtainable and not requiring extensive calculations. In this study, we have utilized experimental data obtained from existing literature for the SHC, TC and viscosity of nanofluids [75–77].

The SHC dataset has incorporated experimental data, which includes fluid temperature, specific heat capacities of CuO nanoparticles, and volume fractions of these nanoparticles. These inputs are collectively used for accurately estimating the SHC of CuO/water nanofluids. Notably, the nanoparticle size, within the range of $\approx 23 - 37$ nm, was not included as a descriptor due to the lack of significant dependence between nanoparticle size and the SHC of nanofluids [78, 79]. The experimental data used in constructing the model has been obtained from published literature [75], with an overall uncertainty of $\approx 0.3\%$ for SHC measurements.

The TC dataset is an extensive experimental dataset compiled by Patel *et al* [76] containing information regarding nanoparticle material, base liquid, nanoparticle size, nanoparticle volume fraction and the suspension temperature as inputs, with TC of the nanofluid as the output. The experiments employed Transient Hot Wire equipment and Temperature Oscillation equipment to measure the thermal conductivity of the liquids. Different nanoparticles and base fluids were used to conduct the experiment and create the dataset, namely - Cu, Al, CuO, Al_2O_3 in Water, EG and Transformer Oil.

The viscosity dataset is also an experimental dataset, where the experiments were conducted by Gallego *et al* [77] using Transient Hot Wire and Temperature Oscillation equipment. The input data contains weight fraction of the nanoparticles and temperature, with the viscosity being the output variable. Two different CuO nanoparticles were used in their experiment, one with a declared diameter distribution of 23-37 nm and the other with a diameter distribution of 11 ± 3 nm.

4. Computational methodology

The ML models have been developed in Python using TensorFlow and Keras for ANN, Sklearn for the SVR, GBM and LR models, Scipy for the DE and PySwarms for the PSO. The TensorFlow platform assists in implementing optimal methodologies for data automation, tracking models, monitoring performance, and retraining models. While TensorFlow is an open-source, comprehensive platform and library designed for a variety of machine learning tasks, Keras is a high-level neural network library that operates on top of TensorFlow. Sklearn, also known as scikit-learn, is a Python library that enables the implementation of ML models and statistical modeling. With scikit-learn, users can leverage a wide range of ML models for regression, classification, clustering, as well as statistical tools for analyzing these models. SciPy is a Python library used for scientific and technical computing, offering modules for optimization, interpolation, and other tasks common in science and engineering. PySwarms is a Python library for PSO, facilitating the solving of continuous and discrete optimization problems using the swarm intelligence paradigm.

The SHC dataset used here comprises 84 experimental data points, while the TC and the Viscosity Datasets contain 294 and 64 datapoints respectively. All the datasets have been randomly divided into training and testing datasets in the ratio 8:2. The ratio of 8:2 was found to be the best performing with good generalization compared to other split ratios like 9:1 or 7:3. Training datasets were employed in training the models, and the trained model was evaluated using the testing dataset. The hyperparameters of SVR, GB, and ANN models were optimized by Bayesian optimization algorithm, using ten-fold cross-validation score as the metric for the objective function.

5. Results

Various ML models - SVR, ANN, GB, LR, and Metaheuristic algorithms such as DE and PSO have been used to model the thermal conductivity, viscosity and SHC of nanofluids. The results from these models have been compared with existing theoretical models. Mean Squared Error (MSE), Coefficient of Determination (R^2) and Lin's Concordance Correlation Coefficient (LCCC) have been used as metrics to evaluate the performance of the different models.

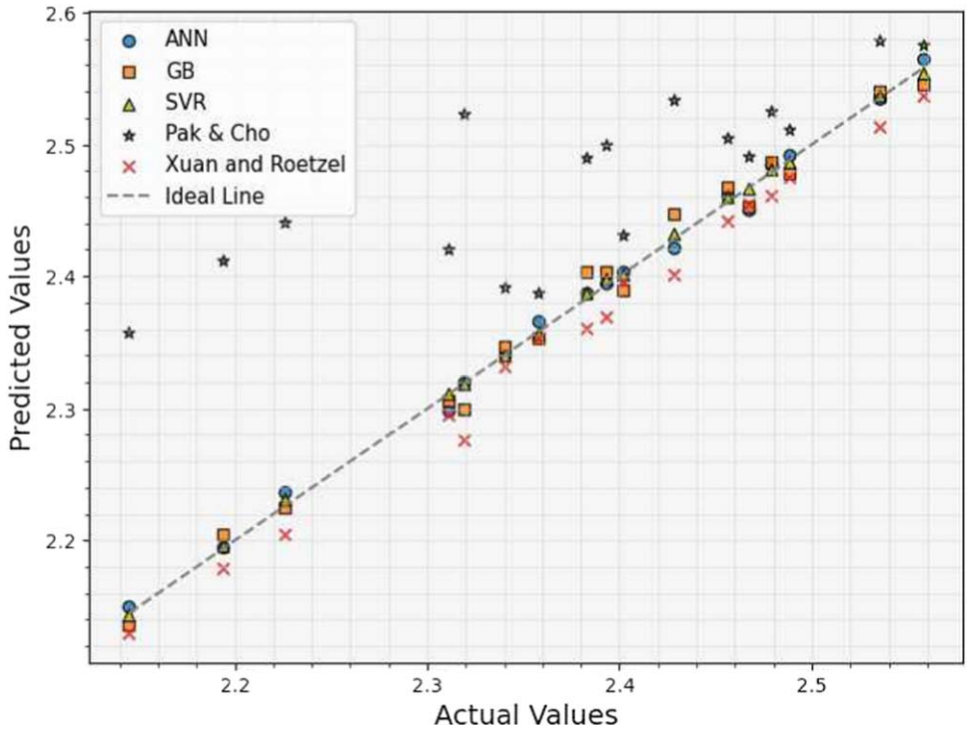

Figure 1. Actual versus Predicted values of ML and theoretical models on SHC test data.

Table 10. Performance of different models on the SHC testing dataset.

Model	MSE	R^2	LCCC
ANN	4.808e−05	0.996	0.9981
SVR	8.828e−06	0.999	0.9996
GB	1.382e−04	0.989	0.9946
LR	7.718e−06	0.999	0.9997
DE	9.627e−06	0.999	0.9996
PSO	1.583e−04	0.988	0.9933
Pak & Cho [80]	1.405e−02	−2.372	0.4542
Xuan & Roetzel [81]	3.912e−04	0.969	0.9849

5.1. SHC

The ML and metaheuristic models have been trained and evaluated on the SHC dataset. These results have been compared with the results from two theoretical models [80, 81].

The theoretical models show large deviations and are outperformed by all the ML and metaheuristic models. Table 10 shows the MSE, R^2 and LCCC values of the different models on the test SHC dataset.

Figures 1 & 2 show the accuracy of the different computational intelligence-based models when compared with the theoretical models. The X axis represents the actual value of SHC, while the Y axis represents the predicted value of SHC. The dotted line represents x = y. Points closer to the line indicate better prediction.

Figure 3 represents a bar chart comparing the different algorithms used for modelling the SHC dataset.

It is observed that LR performed best among the different models.

5.2. Thermal conductivity

Different ML and metaheuristic algorithms have been used on the TC dataset. These results have been compared with the results from the Hamilton Crosser model [82].

All the ML and metaheuristic models have outperformed the Hamilton Crosser model. Table 11 shows the MSE, R^2 and LCCC values of the different models on the test TC dataset.

Figures 4 & 5 show the accuracy of the different computational intelligence-based models when compared to the theoretical model (Hamilton Crosser). The X axis represents the actual value of TC enhancement, while the

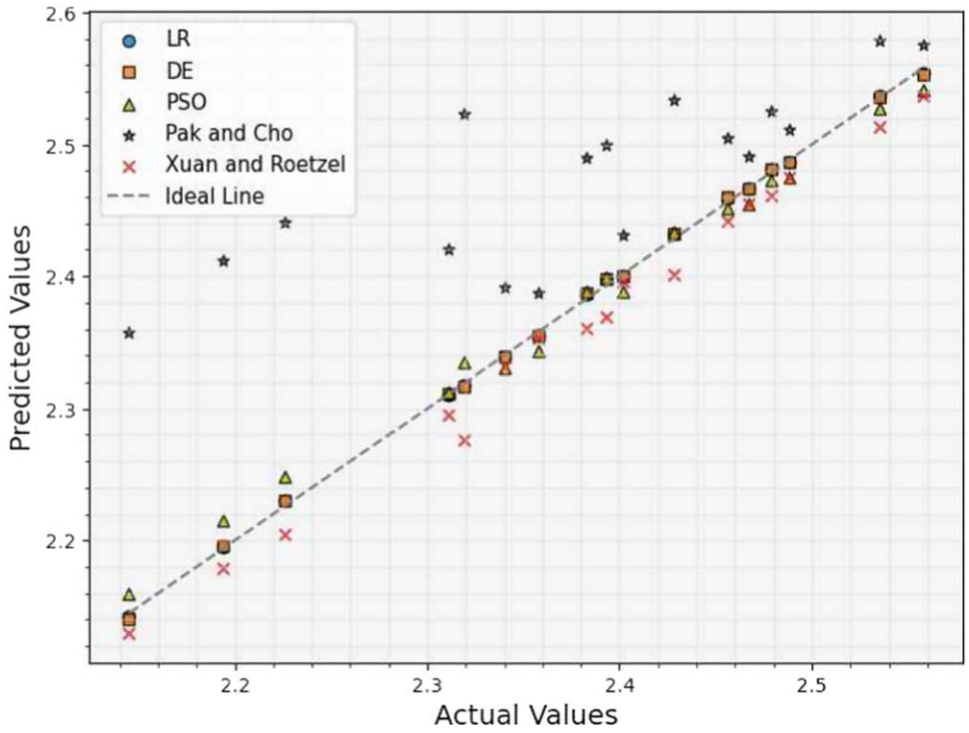

Figure 2. Actual versus Predicted values of LR, PSO, DE and theoretical models on SHC test data.

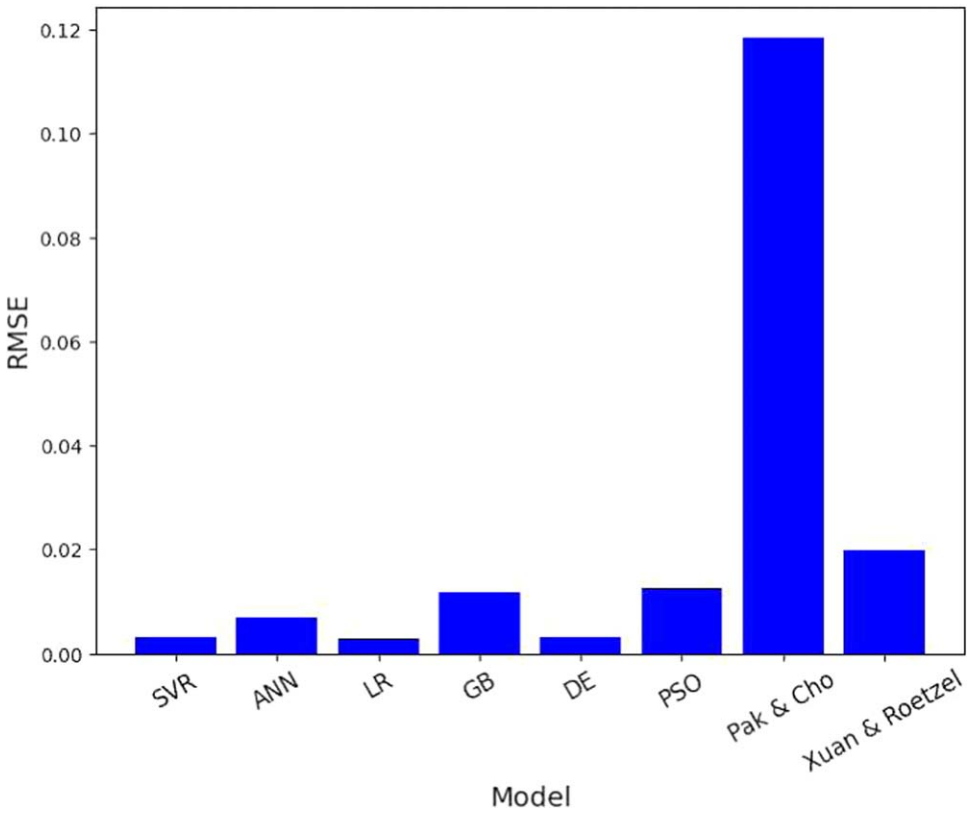

Figure 3. Comparison between RMSE values of the different models on the SHC dataset.

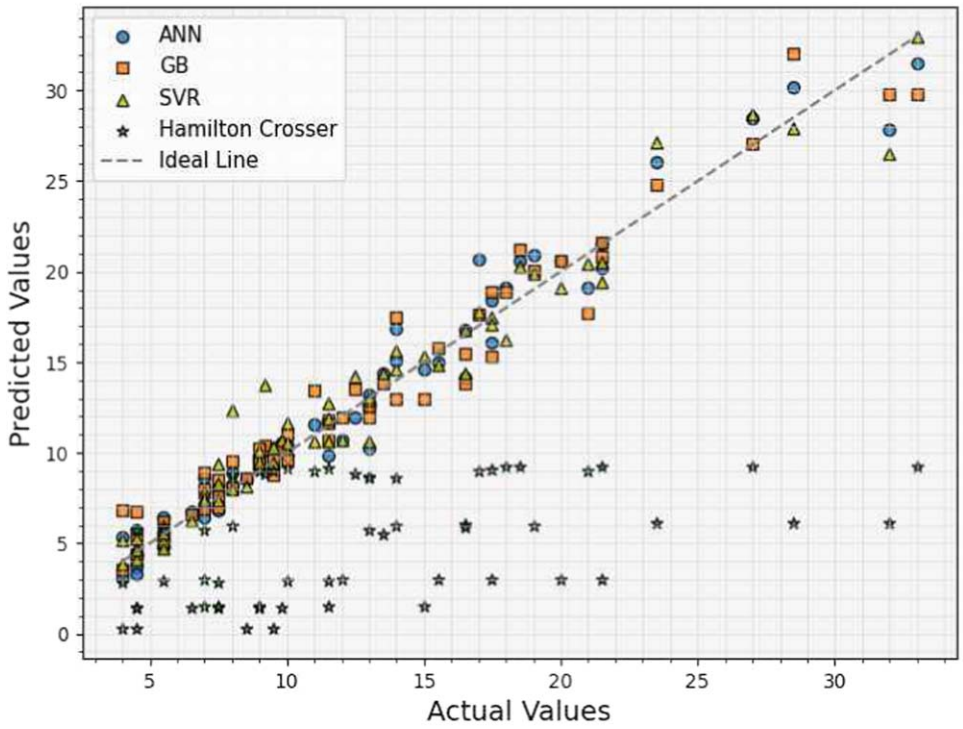

Figure 4. Actual versus Predicted values of ML and theoretical models on TC test data.

Table 11. Performance of different models on the TC testing dataset.

Model	MSE	R^2	LCCC
ANN	1.821e00	0.962	0.9809
SVR	2.329e00	0.951	0.9746
GB	2.04e00	0.957	0.9779
LR	1.34e01	0.717	0.8510
DE	1.34e01	0.717	0.8496
PSO	1.39e01	0.708	0.8336
Hamilton & Crosser [82]	9.59e01	−1.018	0.1673

Y axis represents the predicted value of TC enhancement. The dotted line represents x = y. Points closer to the line indicate better prediction.

Figure 6 represents a bar chart comparing the different algorithms used for modelling the SHC dataset.

It is observed that ANN has performed best among the different models used.

5.3. Viscosity

The different ML algorithms have been applied on the viscosity dataset and these results have been compared to the results from two theoretical models [83, 84]. Table 12 shows the MSE, R^2 and LCCC values of the different models on the test TC dataset.

From the above table, it is observed that the theoretical models are outperformed by all the ML models. However, the theoretical models performed slightly better than the metaheuristic models and the LR model.

Figures 7 & 8 show the accuracy of the different computational intelligence-based models when compared to the theoretical model (Hamilton Crosser). The X axis represents the actual value of TC enhancement, while the Y axis represents the predicted value of TC enhancement. The dotted line represents the locus of x = y. Points closer to the line indicate better prediction.

Figure 9 represents a bar chart comparing the different algorithms used for modelling the SHC dataset.

It is observed that GB performed best among the different models used.

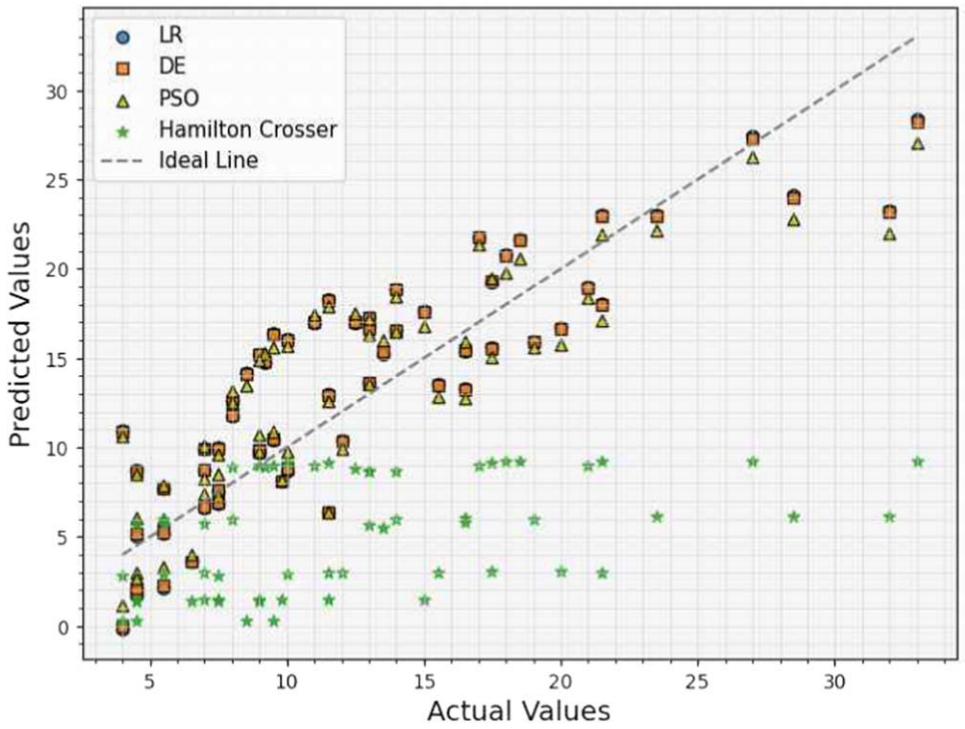

Figure 5. Actual versus Predicted values of LR, PSO, DE and theoretical models on TC test data.

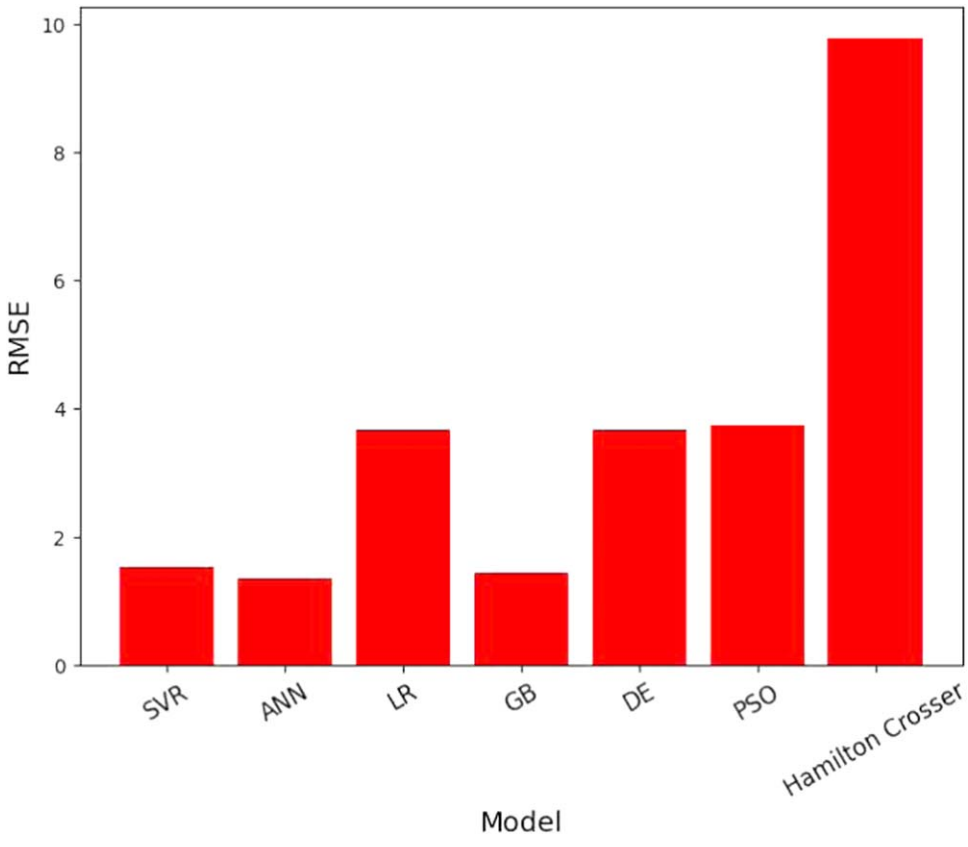

Figure 6. Comparison between RMSE values of the different models on the SHC dataset.

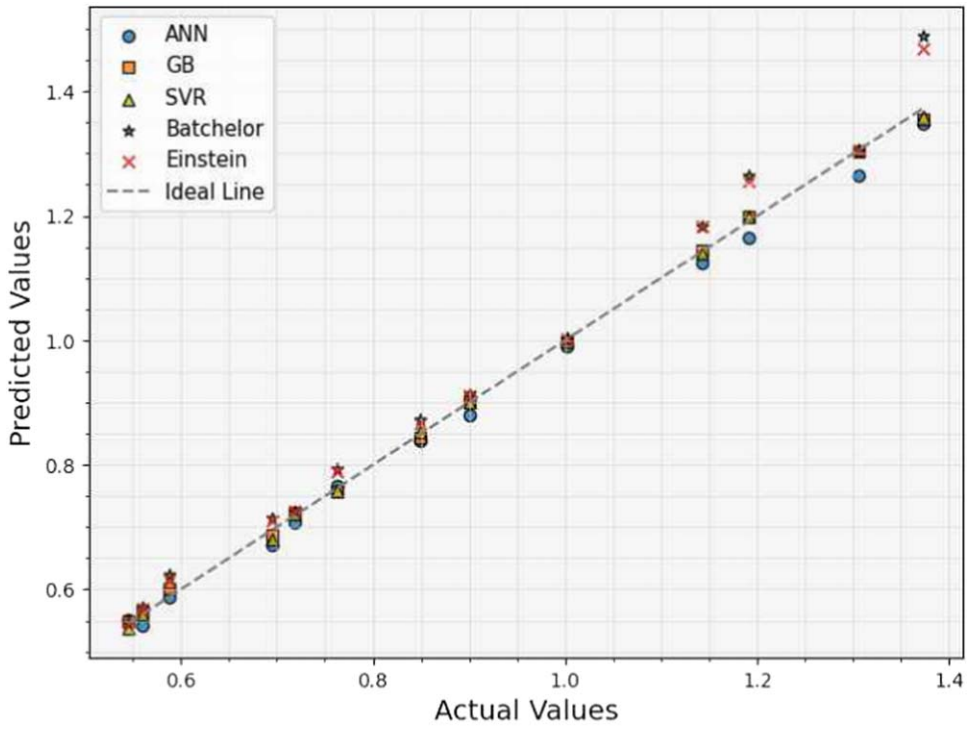

Figure 7. Actual versus Predicted values of ML and theoretical models on viscosity test data.

Table 12. Performance of different models on the viscosity testing dataset.

Model	MSE	R^2	LCCC
ANN	3.761e-04	0.995	0.9974
SVR	8.556e-05	0.999	0.9994
GB	5.478e-05	0.999	0.9996
LR	3.302e-03	0.956	0.9788
DE	3.273e-03	0.956	0.9789
PSO	3.302e-03	0.956	0.9789
Batchelor [84]	1.844e-03	0.975	0.9886
Einstein [83]	1.316e-03	0.982	0.9918

6. Discussion

The present work showed the applicability of various ML and meta-heuristic models for predicting thermophysical properties of nanofluids and compared them with traditional models. The importance of this lies in the significance of nanofluid thermophysical properties for engineering and industrial applications [32].

While the addition of nanoparticles is known to enhances thermal conductivity, accurate forecast of the thermophysical properties like thermal conductivity, specific heat capacity, viscosity is crucial. The relationship between these thermophysical properties with parameters like temperature, volume fraction etc are complex and non-linear [32, 85]. These properties directly affect heat transfer rates and pumping power, and thus their accurate prediction holds industrial and engineering significance.

However, acquiring experimental data of nanofluids is time-consuming and expensive and traditional theoretical and correlation based models fail to predict these thermophysical properties with sufficient accuracy [30, 31]. As such, AI methods like ML and meta-heuristic algorithms offer a cost-effective and efficient alternative for predicting thermophysical properties like viscosity, SHC, and thermal conductivity [31]. These AI-based tools can be developed into apps or integrated into industrial processes for rapid, accurate estimation of these properties, leading to enhanced operational efficiency.

It is worthy to note that methods like Lattice Boltzmann (LBM) [86–88] and Finite Difference (FDM) [89, 90] are useful for simulating fluid dynamics and heat transfer. But they are computationally intensive, especially in

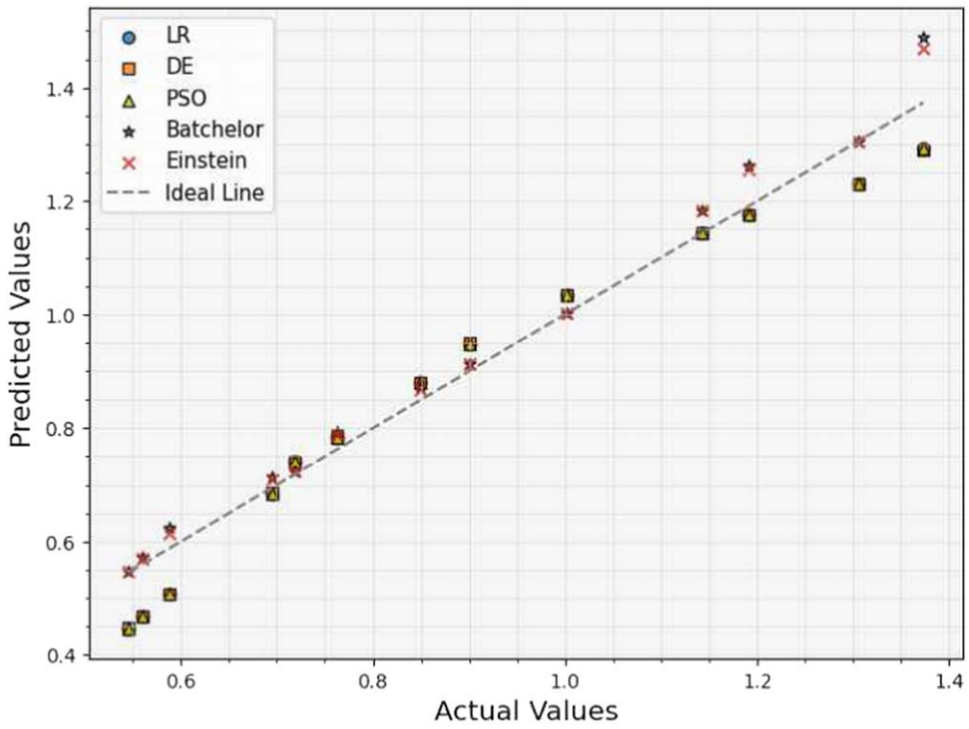

Figure 8. Actual versus Predicted values of LR, PSO, DE and theoretical models on viscosity test data.

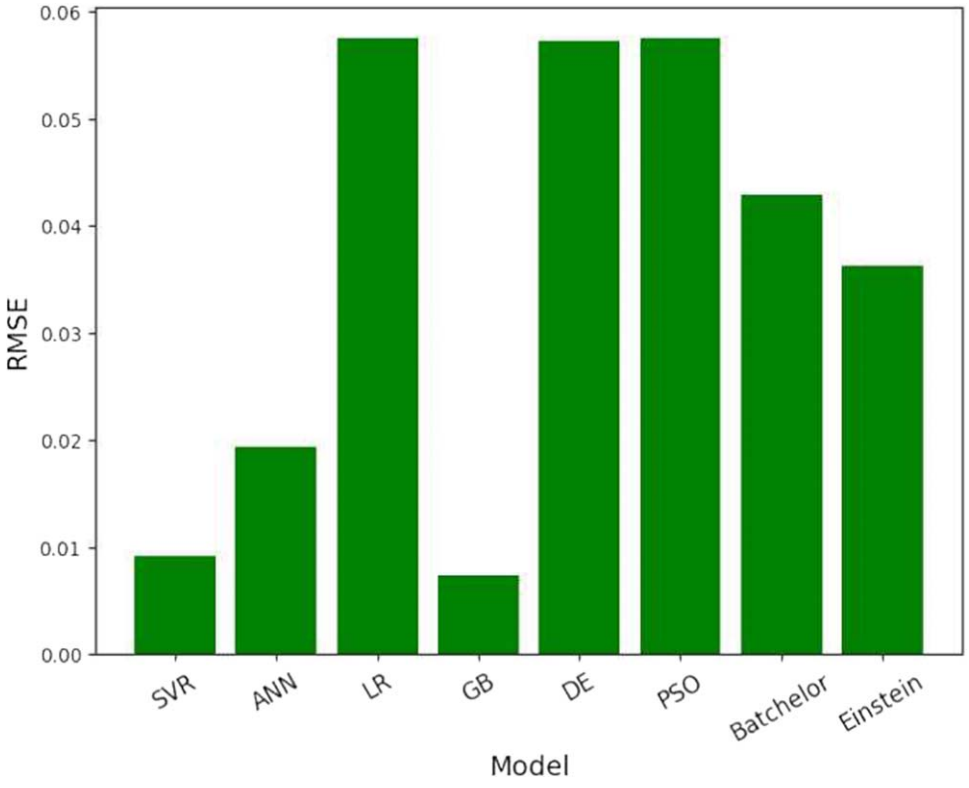

Figure 9. Comparison between RMSE values of the different models on the viscosity dataset.

nanoscale or complex simulations [91–93]. Although GPU computing can speed up the process, these methods still need substantial resources.

ML models, on the other hand, are data-based which allowing them to capture complex relationships without needing detailed knowledge of the underlying physics [94]—which, in the case of nanofluids, can be

complicated. Although ML models can be resource-intensive during training, once trained, they are efficient and computationally inexpensive for predictions. This makes them more practical for real-time or large-scale applications.

While all the ML and meta-heuristic models, in our study, performed well, even small differences in accuracy can have a significant impact on industrial applications. This importance is especially amplified by the fact that experimental determination of these properties are cost prohibitive. The goal is to identify the best-performing model for predicting thermophysical properties with the highest accuracy. Minor improvements in predictions can translate into substantial cost savings and efficiency gains in industries that rely on nanofluids.

7. Conclusion

This study presents a comprehensive examination of ML techniques for predicting the thermophysical properties of various nanofluids. It is evident that integrating advanced computational methods significantly enhances the understanding and prediction accuracy of these properties. Our findings highlight the relative limitation of traditional theoretical models in capturing the complex physics of nanofluids under various conditions. In contrast, ML algorithms such as ANN, SVR, GBM, and LR, along with metaheuristic models like DE and PSO, show superior performance in predicting specific heat capacity (SHC), thermal conductivity (TC), and viscosity with high precision.

The superior performance of the LR model for SHC, the ANN for TC, and the GBM for viscosity illustrates the transformative potential of these computational tools in nanofluid research. The use of ML offers a cost-effective and efficient alternative to exhaustive experimental methods, enabling rapid advancements in the development and application of nanofluids across diverse industrial and technological fields. Additionally, this study demonstrates the effectiveness of hyperparameter optimization in improving model performance.

In conclusion, this research contributes to advancement in the field of nanofluid thermophysical property prediction, demonstrating the superiority of ML models over traditional theoretical approaches. These findings advocate for the broader adoption of machine learning techniques in nanofluid research, paving the way for innovative solutions in thermal management systems. Future research should explore emerging ML algorithms and their applicability to other complex nanofluid properties, further expanding the scope of nanotechnology and its applications.

Acknowledgments

This research received no specific financial grant from any funding agency.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

Author Contribution

Aritra Saha: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing - original draft; Ankan Basu: Data curation, Methodology, Validation, Visualization, Writing - review & editing; Sumanta Banerjee: Conceptualization, Project Administration, Supervision, Writing - review & editing.

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.

ORCID iDs

Aritra Saha https://orcid.org/0009-0000-3157-9447
Ankan Basu https://orcid.org/0009-0001-1966-1277
Sumanta Banerjee https://orcid.org/0009-0005-7854-8281

References

[1] Devendiran D and Amirtham V 2016 *Renewable and Sustainable Energy Reviews* **60** 21–40
[2] Dey D, Kumar P and Samantaray S 2017 *Heat Transfer Research* **46** 1413–42
[3] Assad M and Nazari M 2021 Heat exchangers and nanofluids *Design and Performance Optimization of Renewable Energy Systems* (Academic) 33–42
[4] Al-Rashed M, Dzido G, Korpy M, Smoka J and Wjcik J 2016 *Microelectronics Reliability* **63** 159–65
[5] El-Khouly M, El Bouz M and Sultan G 2021 *Journal of Energy Storage* **39** 102630
[6] Sharifi I, Shokrollahi H and Amiri S 2012 *Journal of Magnetism and Magnetic Materials* **324** 903–15
[7] Hajalilou A, Ferreira L, Melo Jorge M, Reis C and Cruz M 2021 *Journal of Magnetism and Magnetic Materials* **537** 168242
[8] Elsheikh A, Sharshir S, Mostafa M, Essa F and Ahmed Ali M 2018 *Renewable and Sustainable Energy Reviews* **82** 3483–502
[9] Farhana K, Kadirgama K, Rahman M, Ramasamy D, Noor M, Najafi G, Samykano M and Mahamude A 2019 *Nano-Structures & Nano-Objects* **18** 100276
[10] Al-Waeli A, Chaichan M, Kazem H and Sopian K 2017 *Energy Conversion and Management* **148** 963–73
[11] Michael J, S I and R G 2015 R *Renewable and Sustainable Energy Reviews* **51** 62–88
[12] Boyaghchi F, Chavoshi M and Sabeti V 2015 *Energy* **91** 685–99
[13] Soltani S, Kasaeian A, Sarrafha H and Wen D 2017 *Solar Energy* **155** 1033–43
[14] Kanti P K, Sharma P, Maiya M P and Sharma K V 2023 *Solar Energy Materials and Solar Cells* **253** 112207
[15] Alfellag M A, Kamar H M, Sidik N A C, Muhsan A S, Kazi S, Alawi O A and Abidin U 2023 *Journal of Thermal Analysis and Calorimetry* **148** 6645–86
[16] Cengel Y A, Ghajar A J and Kanoglu M 2011 *Heat and mass transfer : fundamentals and applications 4th ed engel series in engineering thermal-fluid sciences* (McGraw-Hill) (https://cir.nii.ac.jp/crid/1130282272223991552)
[17] Munson B R, Young D F and Okiishi T H 1995 *Oceanographic Literature Review* **42** 831
[18] Cengel Y A, Boles M A and Kanoğlu M 2011 *Thermodynamics: an engineering approach* 5 (McGraw-hill)
[19] Pandey H and Gupta N K 2022 *Thermal Science and Engineering Progress* 101281
[20] Mohammadi M, Mohammadi M, Ghahremani A R S and Mohammadi N 2014 *Heat Transfer Engineering* **35** 25–33
[21] Devendiran D K and Amirtham V A 2016 V *Renewable and Sustainable Energy Reviews* **60** 21–40
[22] Saeedi A H, Akbari M and Toghraie D 2018 *Physica E: Low-dimensional Systems and Nanostructures* **99** 285–93
[23] Hamzah M H, Sidik N A C, Ken T L, Mamat R and Najaf G 2017 *International Journal of Heat and Mass Transfer* **115** 630–46
[24] Maleki A, Haghighi A and Mahariq I 2021 *Journal of Molecular Liquids* **322** 114843114853
[25] Esfe M H, Rostamian H, Esfandeh S and Afrand M 2018 *Physica A: Statistical Mechanics and its Applications* **510** 625–34
[26] Esfe M H, Esfandeh S, Afrand M, Rejvani M and Rostamian S H 2018 *Applied Thermal Engineering* **133** 452–63
[27] Nazari M A, Ghasempour R, Ahmadi M H, Heydarian G and Shafi M B 2018 *International Communications in Heat and Mass Transfer* **91** 90–4
[28] Bakthavatchalam B *et al* 2021 *Journal of Nanomaterials* **11** 320–35
[29] Hemmati-Sarapardeh A, Varamesh A, Husein M and Karan K 2018 *Renewable and Sustainable Energy Reviews* **81** 313–29
[30] Alade I, Oyehan T, Bagudu A, Popoola I and Olatunji S 2017 *Advanced Powder Technology* **29** 157–67
[31] Basu A, Saha A, Banerjee S, Roy P C and Kundu B 2024 *Energies* **17** (https://mdpi.com/1996-1073/17/6/1351)
[32] Ma T, Guo Z, Lin M and Wang Q 2021 *Renewable and Sustainable Energy Reviews* **138** 110494
[33] Hemmat Esfe M, Ahangar M, Toghraie D, Hajmohammad M, Rostamian H and Tourang H 2016 *Journal of Thermal Analysis and Calorimetry* 1–7
[34] Rostamian S, Biglari M, Saedodin S and Hemmat Esfe M 2017 *Journal of Molecular Liquids* **231** 364–9
[35] Sharma P, Ramesh K, Parameshwaran R and Deshmukh S S 2022 *Case Studies in Thermal Engineering* **30** 101658
[36] Ahmadi M H, Mohseni-Gharyehsafa B, Ghazvini M, Goodarzi M, Jilte R D and Kumar R 2020 *Journal of Thermal Analysis and Calorimetry* **139** 2585–99
[37] Kanti P, Sharma K, Yashawantha K and Dmk S 2021 *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects* 1–20
[38] Dai X, Andani H T, Alizadeh A, Abed A M, Smaisim G F, Hadrawi S K, Karimi M, Shamsborhan M and Toghraie D 2023 *Engineering Applications of Artificial Intelligence* **122** 106107
[39] Alade I O, Abd Rahman M A, Abbas Z, Yaakob Y and Saleh T A 2020 *Solar Energy* **197** 485–90
[40] Zhang Y and Xu X 2021 *AIChE Journal* **67** e17289
[41] Said Z, Sharma P, Elavarasan R M, Tiwari A K and Rathod M K 2022 *Journal of Energy Storage* **54** 105230
[42] Cortes C and Vapnik V 1995 *Machine Learning* **20** 273–97
[43] Vapnik V 1999 *IEEE Transactions on Neural Networks and Learning Systems* **10** 988–99
[44] Koji P and Omorjan R 2017 *Chemical Engineering Research and Design* **125** 398–407
[45] Shawe-Taylor J and Cristianini N 2004 *Kernel Methods for Pattern Analysis* (Cambridge University Press)
[46] Boser B, Guyon I and Vapnik V 1992 A training algorithm for optimal margin classifiers *Proceedings of the fifth annual workshop on Computational learning theory* (https://doi.org/10.1145/130385.130401)
[47] Smola A and Schölkopf B 2004 *Statistics and Computing* **14** 199–222
[48] Chen T and Guestrin C 2016 Xgboost: A scalable tree boosting system *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (https://doi.org/10.1145/2939672.2939785)
[49] Friedman J 2001 *The Annals of Statistics* **29** 1189–232
[50] Friedman J 2002 *Computational Statistics and Data Analysis* **38** 367–78
[51] Ke G *et al* 2017 *Lightgbm: A highly efficient gradient boosting decision tree Advances in Neural Information Processing Systems 30*
[52] Prokhorenkova L, Gusev G, Vorobev A, Dorogush A V and Gulin A 2018 *Catboost: unbiased boosting with categorical features Advances in Neural Information Processing Systems 31*
[53] Esfe M, Wongwises S, Naderi A, Asadi A, Safaei M, Rostamian H, Dahari M and Karimipour A 2015 *International Communications in Heat and Mass Transfer* **66** 100–4
[54] Heidari E, Sobati M and Movahedirad S 2016 *Chemometrics and Intelligent Laboratory Systems* **155** 73–85
[55] Sheikholeslami M, Gerdroodbary M, Moradi R, Shafee A and Li Z 2019 *Computer Methods in Applied Mechanics and Engineering* **344** 1–12
[56] Rostami S, Kalbasi R, Sina N and Goldanlou A 2021 *Journal of Thermal Analysis and Calorimetry* **145** 2095–104
[57] Rumelhart D *et al* 1986 *Nature* **323** 533–6
[58] Nield D and Kuznetsov A 2009 *International Journal of Heat and Mass Transfer* **52** 5792–5

[59] Khan W and Pop I 2010 *International Journal of Heat and Mass Transfer* **53** 2477–83
[60] Neethu T, Sabu A, Mathew A, Wakif A and Areekara S 2022 *International Communications in Heat and Mass Transfer* **135** 106115
[61] Esfahani J, Safaei M, Goharimanesh M, De Oliveira L, Goodarzi M, Shamshirband S and E B F 2017 *Powder Technology* **317** 458–70
[62] Kuznetsov A and Nield D 2014 *International Journal of Thermal Sciences* **77** 126–9
[63] Ahmad A, Asghar S and Afzal S 2016 *Journal of Magnetism and Magnetic Materials* **402** 44–8
[64] Storn R and Price K 1997 *Journal of global optimization* **11** 341–59
[65] Das S and Suganthan P N 2011 *IEEE Transactions on Evolutionary Computation* **15** 4–31
[66] Ahmad M F, Isa N A M, Lim W H and Ang K M 2022 *Alexandria Engineering Journal* **61** 3831–72
[67] Georgioudakis M and Plevris V 2020 *Frontiers in Built Environment* **6**
[68] Qin A K, Huang V L and Suganthan P N 2009 *IEEE Transactions on Evolutionary Computation* **13** 398–417
[69] Kennedy J and Eberhart R 1995 Particle swarm optimization *Proceedings of ICNN'95 - International Conference on Neural Networks* 4, 1942–8
[70] Shi Y and Eberhart R C 1999 Empirical study of particle swarm optimization *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)* 3, 1945–50
[71] Coello C, Pulido G and Lechuga M 2004 *IEEE Transactions on Evolutionary Computation* **8** 256–79
[72] Eberhart and Shi Y 2001 Particle swarm optimization: developments, applications and resources *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)* 81–6 1
[73] Liang J, Qin A, Suganthan P and Baskar S 2006 *IEEE Transactions on Evolutionary Computation* **10** 281–95
[74] Gad A G 2022 *Archives of computational methods in engineering* **29** 2531–61
[75] Barbs B, Pramo R, Blanco E and Casanova C 2014 *Journal of Thermal Analysis and Calorimetry* **115** 1883–91
[76] Patel H, Sundararajan T and Das S 2010 *Journal of Nanoparticle Research* **12** 1015–31
[77] Pastoriza Gallego M, Casanova C, Legido S J L and Pieiro M 2011 *Fluid Phase Equilibria* **300** 188–96
[78] Satti J, Das D and Ray D 2017 *International Journal of Heat and Mass Transfer* **107** 871–81
[79] Żyła G, Fal J, Bikić S and Wanic M 2018 *Physica E: Low-dimensional Systems and Nanostructures* **104** 82–90
[80] Pak B C and Cho Y I 1998 *Experimental Heat Transfer an International Journal* **11** 151–70
[81] Xuan Y and Roetzel W 2000 *International Journal of heat and Mass transfer* **43** 3701–7
[82] Hamilton R L and Crosser O K 1962 *Industrial & Engineering Chemistry Fundamentals* **1** 187–91
[83] Einstein A 1906 *Annln., Phys* **19** 289–306
[84] Batchelor G K 1977 *Journal of fluid mechanics* **83** 97–117
[85] Angayarkanni S and Philip J 2015 *Advances in Colloid and Interface Science* **225** 146–76
[86] Sheikholeslami M, Gorji-Bandpy M and Ganji D 2014 *Powder Technology* **254** 82–93
[87] Sheikholeslami M, Ashorynejad H and Rana P 2016 *Journal of Molecular Liquids* **214** 86–95
[88] Ma Y, Mohebbi R, Rashidi M M and Yang Z 2018 *Physics of Fluids* **30** 032001
[89] Hanif H 2021 *Physica Scripta* **96** 095215
[90] Abu-Hamdeh N H, Aljinaidi A A, Eltaher M A, Almitani K H, Alnefaie K A, Abusorrah A M and Safaei M R 2021 *Mathematics* **9** (https://mdpi.com/2227-7390/9/24/3153)
[91] Guo Z and Shu C 2013 *Lattice Boltzmann Method and Its Applications in Engineering* (World Scientific) (https://doi.org/10.1142/8806)
[92] LeVeque R J 2007 *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems* (SIAM)
[93] Blazek J 2015 *Computational fluid dynamics: principles and applications* (Butterworth-Heinemann)
[94] Mendez M A, Ianiro A, Noack B R and Brunton S L 2023 *Data-driven fluid mechanics: combining first principles and machine learning* (Cambridge University Press)