



Stochastic interpretable machine learning based multiscale modeling in thermal conductivity of Polymeric graphene-enhanced composites

Bokai Liu ^{a,b}, Weizhuo Lu ^b, Thomas Olofsson ^b, Xiaoying Zhuang ^c, Timon Rabczuk ^{a,*}

^a Institute of Structural Mechanics, Bauhaus-Universität Weimar, Marienstr. 15, D-99423 Weimar, Germany

^b Intelligent Human-Buildings Interactions Lab, Department of Applied Physics and Electronics, Umeå University, 901 87 Umeå, Sweden

^c Institute of Photonics, Gottfried Wilhelm Leibniz Universität Hannover, 30167 Hannover, Germany



ARTICLE INFO

Keywords:

Polymeric graphene-enhanced composites (PGECs)

Interpretable Integrated Learning

Stochastic multi-scale modeling

Thermal properties

Data-driven technique

ABSTRACT

We introduce an interpretable stochastic integrated machine learning based multiscale approach for the prediction of the macroscopic thermal conductivity in Polymeric graphene-enhanced composites (PGECs). This method encompasses the propagation of uncertain input parameters from the meso to macro scale, implemented through a foundational bottom-up multi-scale framework. In this context, Representative Volume Elements in Finite Element Modeling (RVE-FEM) are employed to derive the homogenized thermal conductivity. Besides, we employ two sets of techniques: Regression-tree-based methods (Random Forest and Gradient Boosting Machine) and Neural networks-based approaches (Artificial Neural Networks and Deep Neural Networks). To ascertain the relative influence of factors on output estimations, the SHapley Additive exPlanations (SHAP) algorithm is integrated. This interpretable machine learning methodology demonstrates strong alignment with published experimental data. It holds promise as an efficient and versatile tool for designing new composite materials tailored to applications involving thermal management.

1. Introduction

Polymeric graphene-enhanced composites (PGECs) are a class of advanced materials that have attracted significant attention in recent years due to their outstanding and designed materials properties, particularly in thermal conductivity since graphene embedded. Graphene, a two-dimensional material composed of carbon atoms arranged in a honeycomb lattice, is an excellent heat conductor due to its high intrinsic thermal conductivity [1]. Polymeric materials, on the other hand, have relatively low thermal conductivity. However, by incorporating graphene into polymers, the resulting PGECs can exhibit significantly enhanced thermal properties [2].

Thermal conductivity is the measure of how well a material conducts heat and is a critical property for many industrial applications, including electronic packaging, thermal management, Aerospace engineering, Automotive, and energy storage [3]. For example, some of the potential applications of PGECs can be used as thermal management materials in electronic devices to dissipate heat generated by the components [4]. This can improve the reliability and lifespan of electronic devices. Or it can be used as energy storage components in battery electrodes or supercapacitors to improve heat dissipation, which can enhance the overall energy storage capacity and lifespan of these devices [5].

The enhancement in thermal conductivity of PGECs is due to the excellent thermal conductivity of graphene, which can efficiently transfer heat throughout the material. Additionally, the high surface area and flexibility of graphene enable efficient heat transfer between the polymer matrix and graphene fillers. The thermal conductivity of PGECs can be further improved by controlling the size, shape, and concentration of graphene fillers [6]. Moreover, the choice of polymer matrix can also impact the thermal conductivity of PGECs. For example, polyethylene-based PGECs have shown high thermal conductivity due to their low thermal resistance at the interface between the graphene fillers and polymer matrix.

In order to figure out the inter mechanism, we need to apply a computational approach named multi-scale modeling that combines multiple length scales to study material behavior and properties. Multi-scale analysis can bridge the gap between the different length scales in materials, from fine scale to the coarse scale [7]. It allows for the investigation of phenomena that occur at different scales, such as the influence of atomic arrangements on the mechanical properties of materials, the effect of microstructure on material behavior, or the impact of macroscopic loading on the microstructure. In addition, uncertainties should also be considered in reality due to variations in material properties, manufacturing processes, and environmental conditions. These

* Corresponding author.

E-mail address: timon.rabczuk@uni-weimar.de (T. Rabczuk).

uncertainties can lead to variations in the behavior and performance of materials, which can impact the reliability and safety of engineering structures. Therefore, stochastic multiscale modeling is applied in this study. It involves incorporating the inherent uncertainties that exist at various length scales in materials and predicting their effects on material properties and behavior. The goal of this approach is to provide a more accurate and reliable prediction of material performance under various conditions by taking into account the uncertainties in the material properties and the interaction between different scales.

Many studies have been conducted experimentally. Joanna Wilk et al. study the thermal properties of graphene oxide/rubber composites experimentally, with varying weight concentrations of graphene oxide as a filler [8]. C. Selvam et al. investigate the thermal conductivity of ethylene glycol and water with graphene nanoplatelets using sodium deoxycholate as a surfactant [9]. Na Song et al. present a novel approach for fabricating highly thermally conductive polypropylene/graphene composites with a three-dimensional graphene framework, achieved through a matrix functionalization method [10]. However, due to the limitations of pure experimental methods, which can be time-consuming and cannot fully capture complex interface behavior, computational approaches have been increasingly utilized to investigate polymeric graphene-enhanced composites at different length scales [11]. Van-Thien Tran et al. propose a BCMO-ANN algorithm for vibration and buckling optimization of functionally graded porous (FGP) microplates [12]. Bao-Loi Dang et al. propose a systematic and time-efficient approach with data-driven techniques in combination with numerical and experimental data to calibrate 2D VARANS-VOF models for simulation of wave interaction with porous plate in numerical wave tank [13]. Shengchun Wang et al. introduce a deep learning-based algorithm for recognizing rail profiles and employ a template-matching driven tracking algorithm to rapidly track railhead laser stripes, resolving profile measurement for passing trains at crossings and achieving precise positioning and swift tracking of diverse laser stripes [14].

Numerous studies have indicated that the macroscopic thermal conductivity of composites is significantly influenced by those various factors. Mahmood M. Shokrieh et al. developed a stochastic multiscale model to analyze the mechanical properties of carbon nanotube (CNT) composites [15]. The model takes into account several uncertain input parameters, such as the length, orientation, agglomeration, curvature, and dispersion of the CNTs. Similar contributions have also been made by Vu-Bac et al. [16]. In our previous work, we propose an uncertainty analysis method for stochastic modeling of polymeric nanocomposites (PNCs) [17–20].

Due to the high computational cost of stochastic multiscale models, surrogate approaches have been developed to propagate uncertain parameters across scales [21,22]. Machine learning has become a popular tool for this purpose, with the emergence of high-performance computing and artificial intelligence [23]. It provides a powerful set of tools for analyzing and modeling such complex systems. By using statistical and computational techniques, machine learning algorithms can automatically identify patterns in the data and make predictions about future outcomes [24]. Machine learning has been used in materials design and multiscale analysis. For example Hongwei Guo et al. propose a stochastic deep collocation method (DCM) based on neural architecture search (NAS) and transfer learning for heterogeneous porous media [25]. Other examples include the combined technique of long short-term memory and hidden Markov models to predict crack patterns presented by Nguyen-Le et al. [26], an effective method for crack identification suggested by S. Khatir et al. [27], and a new modified damage indicator using transmissibility techniques combined with ANN to improve Local Frequency Response Ratio (LFCR) proposed by Roumaissa Zenzen et al. [28]. Huang et al. adopt a machine learning approach to predict the mechanical properties of carbon nanotube (CNT)-reinforced cementitious composites [29]. Tran-Ngoc et al. put forward a novel machine-learning approach that uses global search

techniques and vectorized data for damage detection in structures [30]. However, these works focus more on the performance of algorithms and models, and do not discuss in-depth the internal mechanism and interpretability of the models. One of the criticisms of machine learning (ML) methods in materials design is that they are often considered black-box models that lack transparency and do not provide insights into the underlying physical mechanisms governing the problem at hand.

To address this issue, researchers have been working on developing more transparent ML models that can provide insights into the input-output relationships and sensitivities of the model. One approach to achieving this is through the use of explainable AI (XAI) techniques, which aim to make the decision-making process of ML models more interpretable and transparent [31]. For a better interpretation of ML models, a unified approach, SHAP (SHapley Additive exPlanations) is recently developed and commonly used [32]. It provides a local explanation for a particular prediction by assigning each feature a value that indicates its contribution to the prediction relative to the average prediction across the entire dataset. These contributions are then used to generate a summary plot that ranks the features by their importance and shows how they contribute to the prediction for a given instance [33].

The objective of this study is to develop a stochastic integrated machine learning (ML) model that can accurately predict the thermal conductivity of Polymeric graphene-enhanced composites (PGECs). Additionally, the study aims to establish an interpretable quantitative relationship between the model predictions and input variables using the SHAP (Shapley Additive Explanations) method. This article is structured as follows. Section 2 presents the general methodology, followed by an introduction of the materials in Section 3. Machine learning (ML) models and interpretability are discussed in Section 4, and the numerical results are presented in Section 5. Finally, the manuscript concludes with Section 6, which outlines the conclusions.

2. Methodology of research

We propose a machine learning-based multi-scale stochastic model, which comprises two main components: stochastic multi-scale modeling and machine learning methods. Our modeling strategy employs a hierarchical approach, bridging two different length scales from the meso to macro scales, as illustrated in Fig. 1. Our approach begins with a bottom-up approach that transfers information through the length scales while accounting for uncertainties. In this hierarchical framework, the output of the finer scale serves as the input for the next coarser scale. Finally, we employ machine learning to analyze the output from the stochastic multi-scale model. The entire approach involves three steps:

- (1) Bottom-up modeling;
- (2) Stochastic modeling;
- (3) Data-driven methods.

3. Stochastic multi-scale modeling

3.1. Multi-scale modeling

We employ a bottom-up approach, i.e. a hierarchical multiscale method where information is transferred only from the fine scale to the next coarser scale. Fig. 2 shows the associated flowchart. The models at different length scales will be described subsequently.

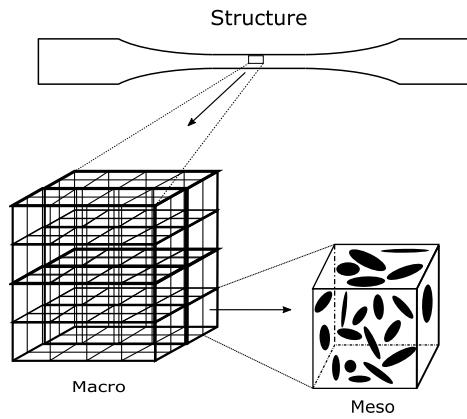


Fig. 1. Multi-scale modeling scheme.

3.1.1. Meso-scale modeling

Our multi-scale modeling approach involves the use of continuum models at the meso-scale, with Representative Volume Elements (RVEs) consisting of a limited number of inclusions that can accurately represent the material properties. The cubic RVE, shown in Fig. 3, is used as a common assumption, and the graphene fillers within the RVE are simplified as disks. We use the commercial package Abaqus with a user-designed Python script, which automatically generates the RVE structures based on a 3D non-collision algorithm written in C++ [6]. The fillers are placed based on given probability density functions (PDFs) of the input parameters. The flowchart for our meso-scale modeling approach is summarized in Fig. 2 on the upper side, while the discretization with quadratic tetrahedra elements is illustrated in Fig. 3. Regarding the element type, we apply DC3D4 in the finite element model for both matrix and inside fillers.

At the meso-scale, we also consider the presence of agglomerations and dispersions of graphene sheets, which can occur when there is a high aspect ratio and high volume fraction of graphene. To analyze and quantify the degree of agglomeration, we use a two-parameter method. This approach involves generating different spheres that can be regarded as gathered zones, as shown in Fig. 4. The spherical area within these zones is considered as the ‘inclusions’. The entire space is divided into two components - $V_{\text{Graphene}}^{\text{inclusion}}$ and $V_{\text{Graphene}}^{\text{matrix}}$:

$$V_{\text{Graphene}} = V_{\text{Graphene}}^{\text{inclusion}} + V_{\text{Graphene}}^{\text{matrix}} \quad (1)$$

where the $V_{\text{Graphene}}^{\text{inclusion}}$ and $V_{\text{Graphene}}^{\text{matrix}}$ denote the graphene sheets placed in the inclusions and matrix, respectively. The agglomeration index ξ and dispersion index ζ are defined as

$$\xi = \frac{V_{\text{inclusion}}}{V}, \quad \zeta = \frac{V_{\text{Graphene}}^{\text{inclusion}}}{V_{\text{Graphene}}} \quad (2)$$

The agglomeration index ξ is the volume fraction of inclusions relative to the entire volume of the RVE. Additionally, we use the dispersion index ζ , which represents the inner volume fraction of graphene sheets in the inclusion relative to the total volume of the disks. When $\xi = \zeta$, the disks are uniformly distributed in the RVE, indicating no agglomeration. However, when $\xi > \zeta$, the disks are unevenly spaced in the RVE, indicating agglomeration.

Determining a suitable RVE size is the first step, and one approach is to use the sample enlargement method. This method involves increasing the RVE size incrementally until the homogenized thermal conductivity converges to a specific value. The convergence criterion is based on averaging the values of the thermal conductivity over a large number of samples:

$$\langle R \rangle = \frac{1}{M} \sum_{K=1}^M R^{(K)} \quad (3)$$

where $R^{(k)}$ is the current value in the k th RVE, and M is the total RVE number. After the ensemble is averaged, a convergence criterion must be satisfied to define a suitable RVE size:

$$\left| \frac{\langle R^{(K+1)} \rangle - \langle R^{(K)} \rangle}{\langle R^{(K)} \rangle} \right| < Tol = 1\% \quad (4)$$

where $R^{(k)}$ is the current value in the k th RVE, and $R^{(k+1)}$ is the $k+1-i$ th RVE. The heat transfer problem is governed by

$$C_f \frac{\partial \theta}{\partial t} + \nabla \cdot q - Q = 0 \quad (5)$$

where θ represents the absolute temperature and Q is the heat source. The heat capacity is denoted by C_f , and the heat flux vector is given by q . For quasi-steady problems, the time-dependent term $C_f \frac{\partial \theta}{\partial t}$ is commonly neglected. Substituting Fourier’s law into the governing equation yields

$$\operatorname{div}(\kappa \nabla \theta) + Q = 0 \quad \text{in } \Omega \quad (6)$$

with natural boundary conditions

$$q_n = -\mathbf{q} \cdot \mathbf{n} = \bar{q} \quad \text{on } \Gamma_q \quad (7)$$

where \mathbf{n} is the normal vector and \bar{q} is the flux at boundary Γ_q . The weak form of the heat equation is given by: Find $\theta \in v$ such that:

$$\int_{\Omega} \kappa \nabla \theta \cdot \nabla \delta \theta d\Omega = - \int_{\Gamma_q} \delta \theta \bar{q} d\Gamma + \int_{\Omega} \delta \theta Q d\Omega \quad \forall \delta \theta \in v_0 \quad (8)$$

with $\theta \in v$ and $\delta \theta \in v_0$, in which θ denotes the trial function and $\delta \theta$ the test function. As Fig. 5 suggests, we apply two different heat fluxes from one direction through the cubic RVE results in a temperature gradient. We then apply Fourier’s law to compute the homogenized thermal conductivity:

$$q = -\kappa \nabla T \quad (9)$$

$$\text{with } \kappa = \begin{Bmatrix} \kappa_{xx} & 0 & 0 \\ 0 & \kappa_{yy} & 0 \\ 0 & 0 & \kappa_{zz} \end{Bmatrix} \quad (10)$$

where $\kappa_{xx} = \kappa_{yy} = \kappa_{zz}$ is defined by applying boundary conditions at different RVE edges and κ is the conductivity of the composite. The macroscopic thermal conductivity of the composite is the output at the meso-scale.

3.1.2. Macro-scale modeling

The Mori-Tanaka method is a widely used model for estimating the effective thermal conductivity of composite materials [6]. It provides a means to predict the overall thermal conductivity of a composite based on the properties of its constituents and their spatial arrangement. At the macroscopic level, a larger structure is considered to account for uncertainties, and it is homogenized by randomly distributing cubes with different thermal properties extracted from simulations at the mesoscale, see Fig. 6. In the context of thermal conductivity, the Mori-Tanaka method considers the thermal conductivities of the matrix material and the reinforcement material, as well as their volume fractions in the composite. It utilizes an averaging technique to estimate the effective thermal conductivity by assuming the thermal conductivity of the composite is governed by two main mechanisms: the thermal conductivity of the matrix and the enhanced thermal conduction pathways provided by the reinforcement.

The equation for estimating the effective thermal conductivity of a polymeric graphene-enhanced composite using the Mori-Tanaka method is as follows:

$$k_{\text{eff}} = k_m + \frac{4k_g V_g}{(1 - V_g) + \frac{k_m}{k_g}} \quad (11)$$

where k_{eff} is the effective thermal conductivity of the composite; k_m is the thermal conductivity of the polymer matrix; k_g is the thermal

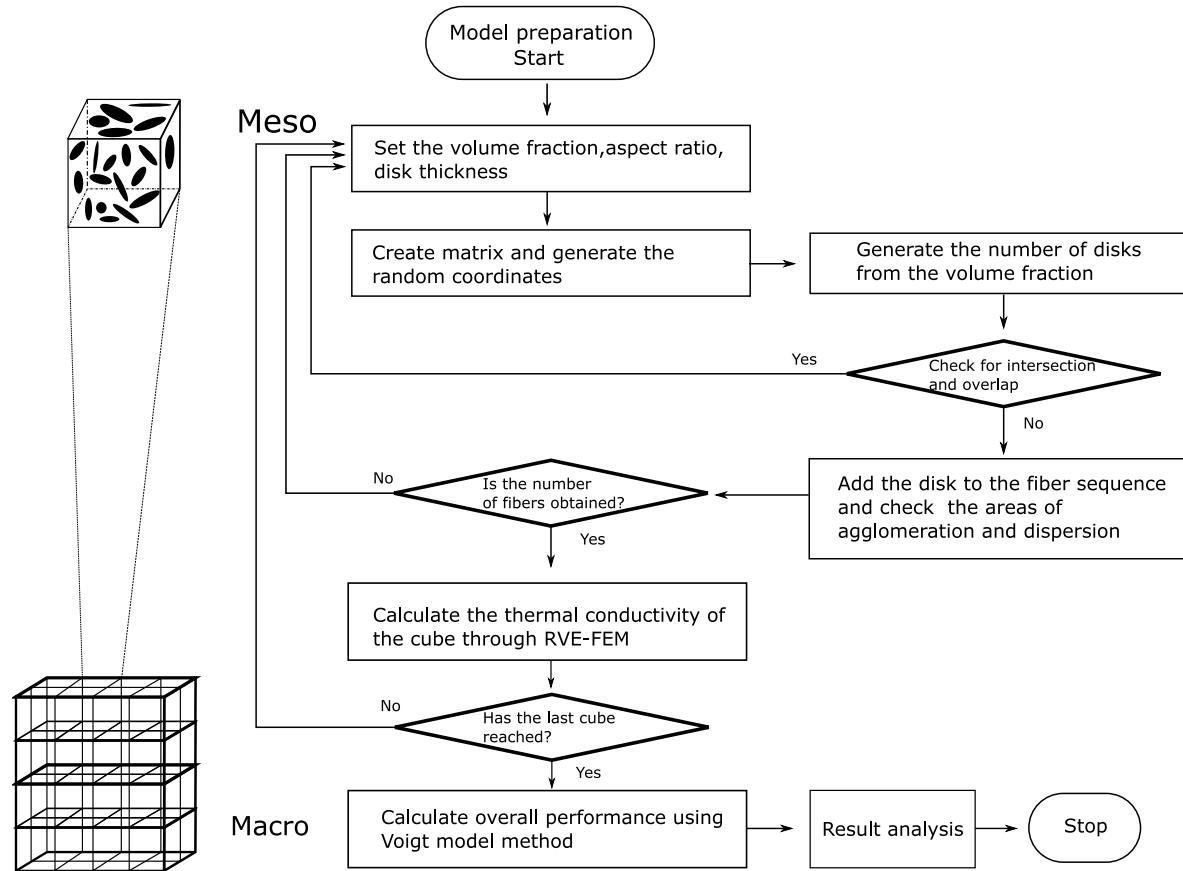


Fig. 2. Multi-scale modeling scheme.

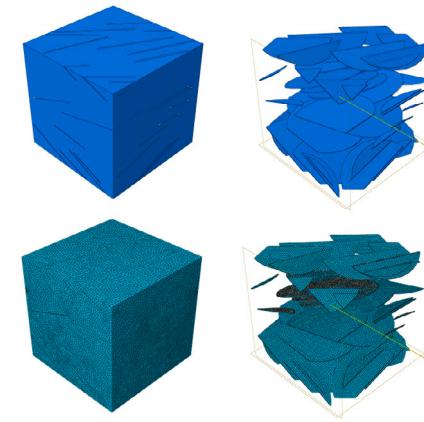


Fig. 3. The RVE cube and Meshing.

conductivity of graphene, and V_g is the volume fraction of graphene in the composite. This equation assumes that the graphene flakes are thin and have a high aspect ratio, allowing heat to be conducted mainly along the graphene plane.

3.2. Stochastic modeling

In this study, uncertainties are considered at different scales, encompassing material properties, micro-structure, and manufacturing processes (see Table 1). While the boundary conditions are assumed to be deterministic, a stochastic analysis is performed by defining a probability density function (PDF) for each uncertain input parameter, which includes its mean and variance. To efficiently model the

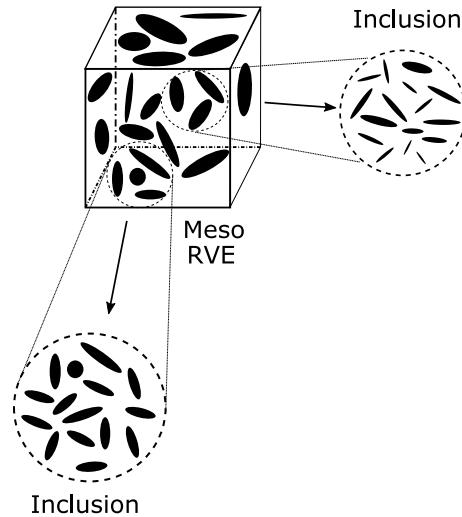


Fig. 4. The agglomeration and dispersion in the Cubic RVE.

stochastic behavior of the system, the Latin Hypercube Sampling (LHS) method is utilized [34]. This approach generates a design matrix of size $N \times m$, where m is the number of input parameters, and N is the number of intervals where the cumulative probability curve is equally divided. By using the PDF of the input parameters, the mean, standard deviation, and variance of the output parameter can be determined. The design matrix is then mapped to the physical model to calculate the actual target output value [35]. LHS optimizes the sampling cost by randomly generating multiple input parameters, which reduces the

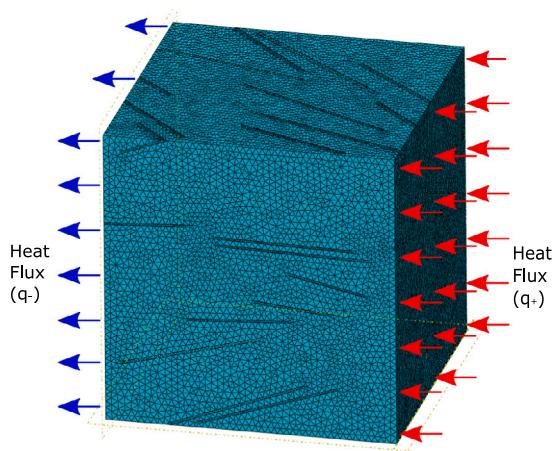


Fig. 5. Applying heat flux in both sides.

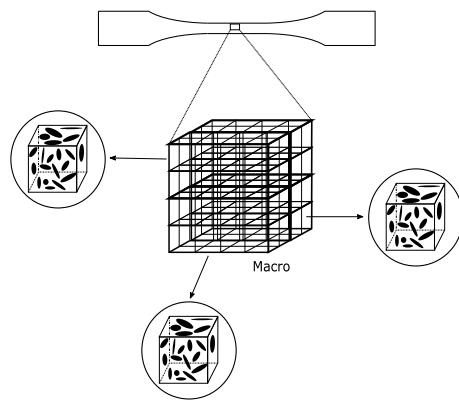


Fig. 6. The material region in macro-scale modeling.

calculation cost and sampling time compared to Monte Carlo Sampling (MCS) [36]. Finally, surrogate models are used to dramatically reduce the computational cost for uncertainty analysis.

4. Data-driven method

4.1. Dataset preparation

We obtain the 'raw data' directly from physical models at different scopes, i.e., FE simulations by generating RVE at the mesoscale and numerical analysis with MATLAB at the macroscale. This initial data is divided into two groups: the training set, which accounts for 80% of the entire database, and the remaining 20% called test set.

Subsequently, all the data should be normalized to ensure that the data of attributes are compressed to the same scale. It can reduce the computational cost and increase the robust. Predictability and accuracy are measured by coefficient of determination (R^2), root mean square error (RMSE), and mean absolute error (MAE). All those metrics are functions of model residuals, where RMSE, R^2 and MAE values can be calculated with

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (Y_{ri} - Y_{pi})^2} \quad (12)$$

$$R^2 = 1 - \frac{\sum_{i=1}^N (Y_{ri} - Y_{pi})^2}{\sum_{i=1}^N (Y_{ri} - Y_{mean})^2} \quad (13)$$

$$MAE = \frac{\sum_{i=1}^N |Y_{ri} - Y_{pi}|}{n} \quad (14)$$

Lastly, the Interpretability should be discussed based on previous steps. SHAP is then used to explain the model predictions, i.e., feature importance and Shapley values.

4.2. Cross validation and hyper-parameters tuning

Cross-validation (CV) is effectively used in machine learning methods for estimating hyper-parameters and building integral models. CV avoids over-fitting to the training data by putting divided data into different training and test or validation sets. The most commonly applied CV method is K-Fold, that is, K verification sets are separated. During training process, the test set does not participate in but verify the performance of models and determine the insufficiency in fitting. If the model performance on the test set is much lower than that on the training set, it indicates the model is over-fitting, otherwise it is under-fitting.

Specifically, the basic idea behind k-fold cross-validation is to split the available data into k subsets, or 'folds', and then use each fold as a testing set while training the model on the remaining k-1 folds. This process is repeated k times, with each fold used once as the testing set. The performance of the model is then calculated by averaging the performance metric (such as accuracy or mean squared error) obtained over the k folds. This provides a more reliable estimate of the model's performance on unseen data compared to using a single train-test split. K-fold cross-validation is particularly useful when the size of the dataset is small, as it allows us to use all of the data for both training and testing. It can also help to reduce the impact of bias and variance in the model by testing it on multiple subsets of the data.

Hyper-parameter tuning is the process of selecting the optimal values for the hyper-parameters of a machine learning model. Hyper-parameters are parameters that cannot be learned during the training process and must be set prior to training. It is an important step in building a successful machine learning model. Choosing the right hyper-parameters can significantly improve the accuracy and performance of the model. However, hyper-parameter tuning can be a challenging and time-consuming task, as the optimal values for the hyper-parameters are often not known beforehand and must be determined through trial and error. Regardless of the approach used, hyper-parameter tuning is an iterative process that involves training and evaluating the model with different sets of hyper-parameters until the optimal set is found. It is important to carefully consider the trade-offs between model complexity and performance, as overly complex models can lead to overfitting, while overly simple models may underfit the data. In this case we use Particle Swarm Optimization to tune the hyper-parameters. The PSO algorithm, which has been proven successful also in our previous studies [18,19]. The SSE is selected as fitness function from the 10-fold CV; it is minimized continuously during the PSO process. It is given below:

$$SSE = \frac{1}{N} \sum_{i=1}^N (Y_{ri} - Y_{pi})^2 \quad (15)$$

where Y_{ri} and Y_{pi} are the required and predicted i th output parameters, respectively; N is the number of output parameters. A swarm size of 450 is chosen, ω , c_1 and c_2 are 1 and (2.0,2.0), respectively [19].

4.3. Regression-tree-based approaches

The tree-based model employs a divide-and-conquer strategy to establish a connection between inputs and outputs. As illustrated in the accompanying Fig. 7, the model divides the data multiple times, with each division corresponding to an if-then judgment. The model's three key elements are the depth and complexity of the tree, the segmentation points, and the prediction equations at the final node. However, a regression model with a single regression tree may not fit the data sufficiently. To address this issue, we utilize two integrated tree-based models: Random Forest (RF) and Gradient Boost Machine (GBM), both of which are based on a tree architecture. The only difference between the two methods is their contribution ratio to the total result obtained by different branches.

Table 1
Model uncertainties.

Scale	Inputs	Mean	Standard deviation	Type of distribution	Sources
Meso	Thermal conductivity of fillers	3978.85	580.79		Suchismita, et al. [37]
	Thermal conductivity of Matrix	0.252806	0.098237		A. Moisala et al. [38]
	Interface resistance	20.2356	5.9294		M. Freitag et al. [39]
	Aspect ratio	59.1911	123.2185		Khoa Bui et al. [40]
	Agglomeration index	0.55105	0.26127		Nam Vu-Bac et al. [16]
	Dispersion index	0.55010	0.26154		Nam Vu-Bac et al. [16]
Macro	Volume fraction	0.054164	0.025811		M. Shokrieh, et al. [15]

4.3.1. Random forest

Random Forest is a powerful machine learning algorithm proposed by Breiman [41] that is commonly used for classification, regression, and other predictive modeling tasks. It is an ensemble learning method that combines multiple decision trees, which are individually weak classifiers, into a strong classifier. In a Random Forest model, multiple decision trees are trained on random subsets of the original data set, with each tree making its own independent prediction. The final prediction of the Random Forest model is obtained by aggregating the predictions of all individual trees, usually by taking the majority vote. The architecture of this tree-based model is illustrated in Fig. 8. One of the main advantages of Random Forest is that it is highly accurate and robust to overfitting, which can be a major issue for decision trees. By combining multiple decision trees, the Random Forest model is able to reduce the variance of the model and improve its generalization performance. Random Forest is also able to handle missing values and outliers in the data, making it a useful tool in many real-world applications. Additionally, it can be used to extract feature importance, which can help in understanding the underlying patterns in the data and improve the interpretability of the model [42]. Alg 1 presents the random forest algorithm

4.3.2. Gradient boosting machine

Gradient Boosting Machine (GBM) is a powerful machine learning technique used for both regression and classification problems. GBM is an ensemble method that combines multiple weak or base models into a stronger predictive model [43]. The idea behind GBM is to iteratively train weak models, where each subsequent model is trained to improve the predictions made by the previous models. The term “gradient” in GBM refers to the use of gradient descent optimization to minimize the loss function. The algorithm uses a differentiable loss function, such as mean squared error or binary cross-entropy, to measure the difference between the predicted values and actual values. The gradient of this loss function is then computed, and the algorithm tries to minimize the loss function by updating the parameters of the weak models in the direction of the negative gradient. The weak models in GBM are

Algorithm 1 Random Forest Algorithm.

```

Require: The number of predictors  $k$ ;  

         The number of trees  $t$ ;  

         The number of split points  $s$ ;  

Ensure: Predictive value  $Y(x)$ ;  

1: Choose the number of models  $m$   

2: For  $t=1,2,\dots,m$  do  

3:   Generate a bootstrap sample from the original data  

4:   Train a tree model on this sample  

5:   For  $s=1,2,\dots,m$  do  

6:     Randomly extract  $k < d$  attributes as predictors  

7:     Choose the optimal variable among  $k$  attributes  

8:   end for  

9:   Tree model rule termination conditions take effect  

10: end for  

11: return  $Y(x)$ ;
```

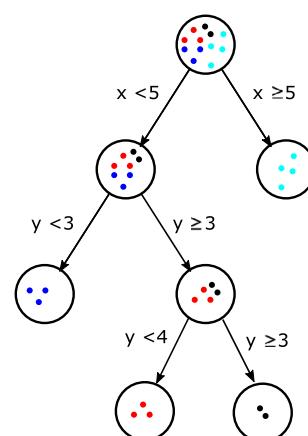


Fig. 7. The illustration of regression tree.

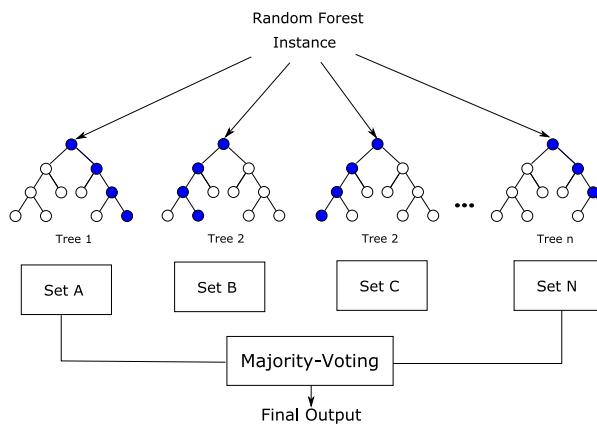


Fig. 8. The framework of Random Forest.

typically decision trees, where each tree makes a prediction based on a set of rules or conditions on the input features. The trees in GBM are trained in a sequential manner, where each tree is trained to correct the errors made by the previous trees. The final prediction is made by combining the predictions of all the trees. Alg. 2 shows a pseudo code of the Gradient Boosting Machine. GBM has several advantages over other machine learning algorithms. It can handle a large number of input features, can capture non-linear relationships between the features and the target variable, and can handle missing data. The Gradient Boosting Machine (GBM) exhibits certain similarities to the Random Forest algorithm in integration. The Gradient Boosting Machine (GBM) shares similarities with the Random Forest algorithm, as both employ a similar integrated idea to improve overall performance. However, there is a fundamental difference between them. In the Random Forest, the individual components are trained simultaneously and contribute equally to the complete model. In contrast, the GBM relies on chain relationships between neighboring components. This means that each component is trained sequentially to improve upon the errors made by the previous one, resulting in a more refined and accurate final model.

Algorithm 2 Gradient Boosting Machine.

Require: The Depth of trees D ;
The number of iterate K ;

Ensure: Predictive value $Y(x)$

- 1: Calculate the mean of the response variable as the initial predicted value of each sample
- 2: **For** $k=1,2,\dots,K$ **do**
- 3: Calculate the residual between observed value and predictive value;
- 4: Use residuals as response variables to fit a tree with depth D ;
- 5: Use the previous steps to get the regression tree to predict each sample;
- 6: Update each sample and add the obtained in the previous step;
- 7: **end for**
- 8: **return** $Y(x)$;

4.4. Neural-network-based approach

A neural network-based approach is a type of machine learning algorithm that is inspired by the structure and function of the human brain. It involves the use of artificial neural networks, which are composed of interconnected nodes or neurons, to process and analyze data [44]. Neural networks learn from examples, and they are trained using large sets of labeled data. During training, the network adjusts the strengths of the connections between its neurons to minimize the difference between its output and the desired output. Once the network has been trained, it can be used to make predictions on new, unseen

data. Neural networks are widely used in a variety of applications, including image and speech recognition, natural language processing, and autonomous driving. They have the ability to learn complex patterns in data, and can often outperform traditional machine learning algorithms in tasks that involve large amounts of data or complex relationships between variables.

In our network, we consider l hidden layers, where the 0-th layer represents the input layer and the $(l+1)$ th layer represents the output layer. To obtain the output for the l th layer, we use the following equation:

$$Y_i^l = F_{l-1} \left(\sum_{i=1}^{m_{l-1}} (X_t^{l-1} w_{ij}^l) + b_i^l \right) \quad (16)$$

where Y_i^l represents the weighted input into the i th neuron on the l th layer; w_{ij}^l is the weight and b_i^l is the bias. The activation function in the l th layer is denoted by F_{l-1} , and m_{l-1} is the number of neurons in the $(l-1)$ th layer. For the input layer and hidden layer 1, we use the Rectified Linear Unit (ReLU) activation function, which is also employed for hidden layer 1 and hidden layer 2. The equations for these activation functions are:

$$F_{ReLU}(x) = \max(0, x) \quad (17)$$

To self-adjust the initial weights and biases after minimizing the computational error, we utilize the Back Propagation method (Alg 3). The Stochastic Gradient Descent (SGD) method is used during the training process to minimize the loss function. Starting from the input layer, the BP algorithm transmits data layer by layer until it reaches the output layer. Once the error between the target value and the output generated by the network is calculated, the error is backpropagated to the hidden layers for comparison, resulting in the adjustment of weights and biases. Each iteration of this process is called an epoch, and the epochs are repeated multiple times until a stopping condition is satisfied. To perform these computations, we use the TensorFlow r2.0 platform with Python/Rstudio, utilizing the RMSProp optimizer, an extension of the Stochastic Gradient Descent (SGD). We define an improved L_2 -loss Root Mean Square Error (RMSE) as our loss function.

Algorithm 3 Back Propagation method.

Require: The training set $D = (x_k, y_k)_{k=1}^m$;
The learning rate η ;

Ensure: Minimize the cumulative error on D

- 1: Randomly initialize weights and deviations in the range of $(0, 1)$
- 2: **For all** $(x_k, y_k) \in D$ **do**
- 3: Calculate the output of the current sample \hat{y}_k ;
- 4: Calculate the gradient of the neurons in the output layer g_j ;
- 5: Calculate the gradient of hidden layer neurons e_h ;
- 6: Update connection weight w_{hj} , v_{ih} and bias θ_j , γ_h
- 7: **end for**
- 8: **return** Neural network Y with determined weights and biases

4.5. Interpretable machine-learning methods

Interpretable machine learning (IML) refers to the ability to understand and interpret the decisions made by machine learning models [45]. This is particularly important in applications where the decision-making process of the model can have significant consequences. Interpretable machine learning methods are designed to address this issue and make machine learning models more transparent and explainable. These methods include Feature importance, Model visualization, Rule-based models, Surrogate models, Counterfactual explanations. By improving the transparency and interpretability of machine learning models, interpretable machine learning methods can increase trust in machine learning models and enable their use in critical applications.

4.5.1. Interpretable machine learning

In traditional machine learning, models are often black boxes, where it is difficult to understand how the model made its predictions or decisions. This lack of transparency can be a significant issue, particularly in applications where the decision-making process of the model can have important consequences. Interpretable machine learning (ML) methods have gained popularity recently because they address the long-standing issue of black-box models. It aims to increase the degree to which humans can understand the reason for a model's output given specific inputs. This not only involves the theoretical mapping of inputs to outputs but also considers factors such as feature importance and how each feature influences the model's output for individual samples and the entire dataset [46].

There are two main categories of interpretable methods: model-specific and model-agnostic [46]. Model-specific methods are limited to certain types of models, such as linear regression or decision trees, which have explicit mathematical rules and a reduced model complexity. However, many advanced ML methods are not inherently interpretable, such as random forests and gradient boosting machine, which combine multiple decision trees.

Model-agnostic interpretable methods are not specific to any particular ML model, and can be applied to a variety of models. These methods analyze the relationship between inputs and outputs, such as feature importance or partial dependence plots. Among the model-agnostic methods, SHAP has gained popularity due to its ability to provide both global and local interpretability [47]. Global interpretability refers to the ability of a model to explain its overall behavior by showing how the output is influenced by different features across the entire dataset. It provides a summary of the model's behavior on a global scale. On the other hand, local interpretability focuses on understanding how the model arrives at its output for a specific input or sample. It provides a detailed understanding of how each feature affects the model's output for a single data point. Both global and local interpretability are important for understanding the behavior of a model and can be achieved using model-agnostic methods such as SHAP. It quantifies the importance of each feature and how it affects the model's prediction on both a whole database and a single sample level, making it a valuable tool for interpretable ML.

4.5.2. Shapley additive explanations

Shapley Additive Explanations (SHAP) is a method for interpreting the output of machine learning models. The method is based on Shapley values, which is a concept from cooperative game theory used to assign a value to each player in a game based on their contribution to the overall outcome [48]. In the context of machine learning, Shapley values are used to determine the contribution of each feature in the input data to the final prediction of the model. It provides a way to explain the predictions of any machine learning model by assigning a contribution value to each feature in the input data. This contribution value indicates how much each feature contributes to the final prediction of the model. The contribution value can be positive or negative, depending on whether the feature increases or decreases the prediction. The framework of this method is illustrated in Fig. 9.

The SHAP method is model-agnostic, which means it can be used to explain the output of any machine learning model, regardless of the algorithm used to train the model. To compute SHAP values, the method uses a sampling-based approach to estimate the contribution of each feature to the prediction. The method generates many possible combinations of input features and calculates the prediction for each combination. It then uses these predictions to estimate the contribution of each feature using the Shapley value concept. The contribution of each feature is represented by a Shapley value [49]. Consequently, the explanation model, $g(x_0)$, can be defined as follows:

$$g(x') = \varphi_0 + \sum_{i=1}^M \varphi_i x'_i \quad (18)$$

where x' represents the vector of simplified input variables that are derived from the original input variables x in the dataset. M represents the number of features in the dataset, while φ_0 is a constant when all inputs are 0. The attribution values for each feature i are represented by φ_i . The SHAP method uses game theory to assign the contribution of each feature to the output of the model, with the Shapley value representing the contribution of each feature.

The explanation model must meet certain requirements, including the following:

- Local accuracy: The output generated by the explanation model should be consistent with the output produced by the original machine learning model for the specific input being explained:

$$g(x') = \varphi_0 + \sum_{i=1}^M \varphi_i x'_i = \varphi(x) \quad (19)$$

In this context, $\varphi(x)$ refers to the machine learning model, for instance, decision tree-based models.

- Missingness: If a feature is absent in a sample, the corresponding feature attribution value will be zero.

$$x'_i = 0 \Rightarrow \varphi_i = 0 \quad (20)$$

- Consistency: the feature attribution values, such as the Shapley value, should have consistent changes in their contributions. For instance, if we have two machine learning models ϕ and ϕ' , we expect the feature attribution values to vary consistently, as expressed by the following equation:

$$\phi_{x'}(z') - \phi_{x'}(z' \setminus i) \geq \phi_x(z') - \phi_x(z' \setminus i) \Rightarrow \varphi_i(\phi, x) \geq \varphi_i(\phi', x) \quad (21)$$

In the given equation, z' represents a subset of the input x' , and $z' \setminus i$ refers to the subset $z'_i = 0$ with the i th feature removed (i.e., its value is set to zero).

The three properties mentioned earlier need to be constrained to obtain a unique solution for the explanation model $g(x)$. It has been theoretically established that there is only one value of φ_i that satisfies these properties and leads to a unique solution.

$$\varphi_i(\phi, x) = \sum_{z' \subseteq x'} \frac{|z'|! (M - |z'| - 1)!}{M!} [\phi_x(z') - \phi_x(z' \setminus i)] \quad (22)$$

where $z' \subseteq x'$ represents the set of all possible subsets of x' , and $|z'|$ refers to the number of non-zero entries in the subset z' .

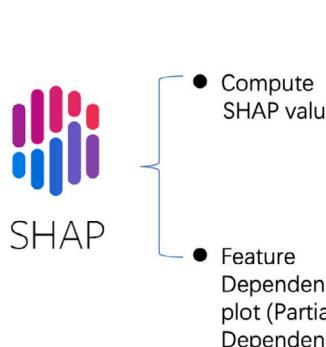
It is evident that solving Eq. (22) directly can be computationally expensive, considering the large number of possible subsets of features. Therefore, various approximation methods have been proposed to calculate the Shapley value. In this case, TreeSHAP is adopted as it is a natural match for tree-based models like random forest and gradient boosting machine, and provides efficient computation of Shapley values for all features. Once the Shapley values are computed, they can be used to explain the output of the model. The individual interpretation plot including Shapley values can be generated for different samples. The Shapley value can be positive (in red) or negative (in blue), indicating whether the feature has a positive or negative impact on the model output, respectively.

When we get the Shapley Value, we also need to consider the impact of different features on the final model prediction results. A feature dependence plot is a visualization tool used to explore the relationship between a feature and the target variable in a machine learning model. It is a type of partial dependence plot (PDP) that shows the marginal effect of a feature on the predicted outcome, while holding all other features constant at their average values or a specific value. In a feature dependence plot, the x -axis represents the values of the feature of interest, and the y -axis shows the corresponding predicted values of the target variable. The plot may also include shaded areas or error bars to indicate the degree of uncertainty in the predictions.

Table 2
The distribution of properties in material region.

Numbers	V_f (%)	Thermal conductivity (W/mk)	Numbers	V_f (%)	Thermal conductivity (W/mk)
1	0.096	3.1925	9	0.019	1.3347
2	0.028	0.2741	10	0.045	1.1527
3	0.047	0.2116	11	0.091	0.9655
4	0.085	0.3651	12	0.04	2.8799
5	0.053	2.8005	13	0.081	0.8083
6	0.037	1.3141	14	0.016	3.9115
7	0.048	1.5166	15	0.048	2.0096
8	0.072	0.4339	16	0.093	1.9335

Thermal conductivity in Voigt model: 1.49147853 (W/mk) (Volume fraction: 8.99%)



- Compute SHAP value
- 1. Global Interpretation
- 2. Individual Interpretation

- Feature Dependence plot (Partial Dependence)

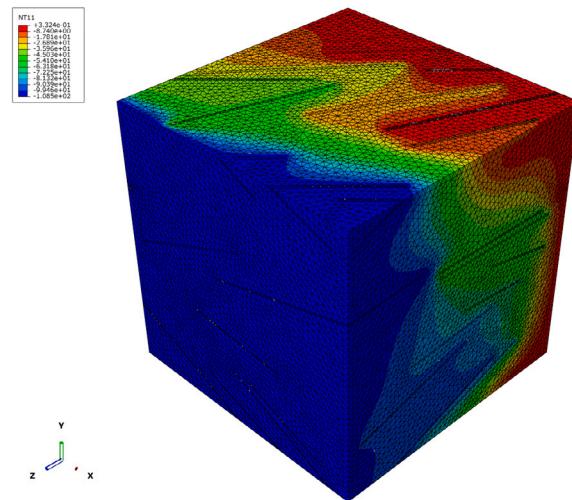


Fig. 9. The framework of Shap Additive Explanations.

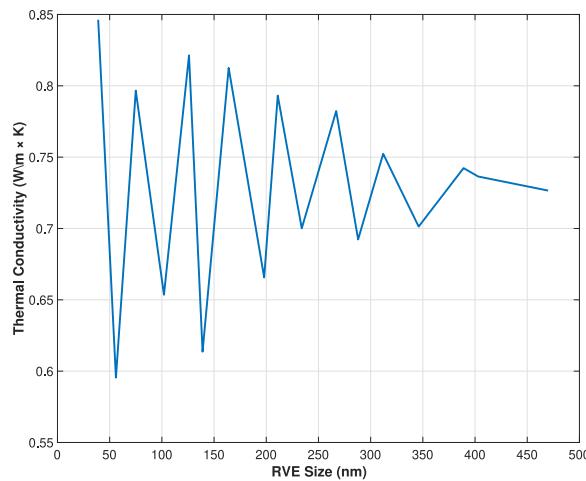


Fig. 10. Thermal conductivity versus RVE size.

5. Numerical results and discussion

5.1. Multi-scale modeling results

Regarding the convergence study, Fig. 10 shows the predicted macroscopic thermal conductivity with the RVE size in the trend converges to a stationary size at specific point. The temperature distribution inside the FEM-RVE model is presented in both Figs. 11 and 12.

Table 2 shows the cubes composition in specific material region with different volume fraction extracted from finer scale. It yields a final thermal conductivity of 1.49147853 (W/mk) with overall volume fraction 8.99%. We also make a comparison between the values in the literature and in the Table 3, where the models' outputs generated by stochastic modeling in multi-scale approach.

Fig. 11. Temperature distribution of composites within RVEs.

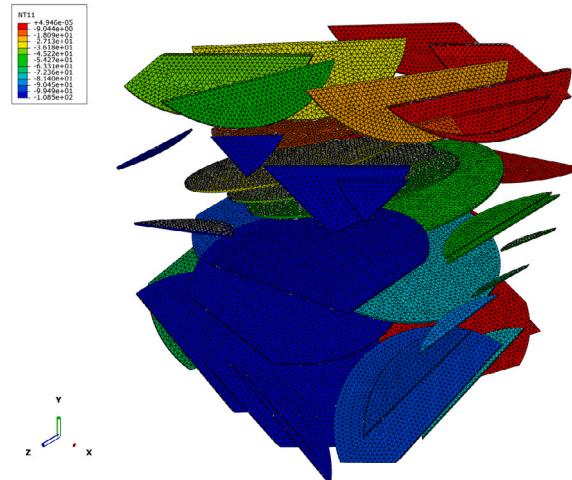


Fig. 12. Temperature distribution of inner plates within RVEs.

5.2. Machine learning modeling results

The first is the result of PSO hyperparameter tuning describing the optimal parameters of the random forest, as shown in the Figs. 13 14 15, where the complexity, the minimum sample of the split is 0, 5, 10. The Fig. 16 illustrates the GBM's hyperparameter tuning, where the learning rate, maximum number of regression tree, and interaction depth are 0.01, 10000, 9 respectively. Besides the optimal values of neural networks are presented in Fig. 17. The loss and mean absolute error over various epochs is shown in Fig. 18. All the selected hyper-parameters is summarized in Table 4.

Table 3

The comparison between experimental value and FEM-RVE predictive value.

Literature	Materials properties	κ_m	V_f	Thermal conductivity (W/mk)		Percent error
				Experiment value	RVE-FEM value	
Sung Ho Song [50]	f-GFs/epoxy (0.19 W/mk)		10%	1.53	1.6671	8.22%
Fuzhong Wang [51]	GnP-C750/epoxy 0.16 W/mk)		3%	0.37	0.4849	23.69%
			5%	0.45	0.5718	21.3%
Yan-Jun Wan [52]	DGEBA-f-GO/epoxy (0.14 W/mk)		4.64%	0.72	0.5698	20.86%
Jingjing Chen [53]	PA6/graphene-GO (0.2575 W/mk)		10%	2.14	2.2681	5.64%
			7.5%	1.78	1.7179	3.4%
			5%	1.39	1.2571	9.50%
			2.5%	0.82	0.9139	10.27%
			1%	0.57	0.4644	18.52%
Zhenghai Tang [54]	BE/graphene 0.05 W/mk)		2.5%	0.542	0.3214	8.17%

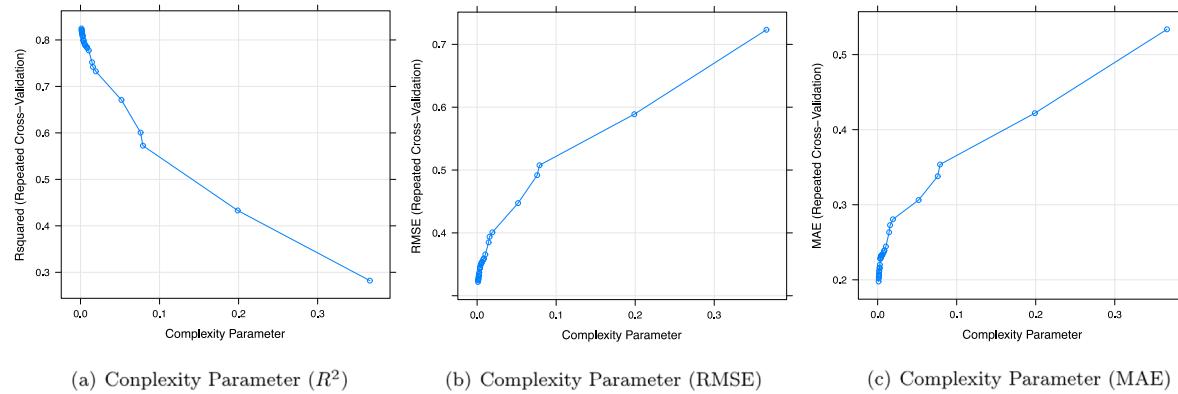


Fig. 13. Hyper-parameter tuning in Regression Tree (Complexity Parameter).

Table 5 presents all the performance of each model. All of them are reliable and supportive to predict the required properties, however, Deep neural network with two hidden layers perform the best among all models.

Let us illustrate the scatter plots of the specific model prediction. The scatter plot of the random forest is shown in Fig. 19, where the most predictions are distributed on the Y=T line. The 20 presents the prediction performance of GBM, and the data are mostly concentrated near the diagonal. Figs. 21 and 22 show the performance of the single hidden layer neural network (ANN) and the deep neural network (DNN). It can be seen from all model performances that DNN is not only better than ANN but also better than other regression tree-based models. In addition, the scattered points are more concentrated in the data interval [0-2.5], indicating the prediction is more accurate in this domain.

When comparing different algorithms, the computational cost also needs to be considered, i.e., the computational complexity. The computational complexity of all models has summarized in Table 6. From here we can see that the regression tree is the most lightweight algorithm, requiring only a small computational cost. The advantage of regression trees is faster prediction and the strongest interpretability. Random forest is an integration of regression trees, and it also has great advantages in terms of computational cost. Meanwhile, the training time increases significantly as the number of samples increases. Fig. 23 shows how computational time increases with sample size. Regression trees are apparently the least sensitive while DNN have the most significant time increase due to the highest complexity.

5.3. Model interpretations

In terms of interpretable machine learning, it is divided into Global Interpretations, Individual Interpretations, and Feature Dependency. The SHAP value can be applied to observe the internal mechanism of 'black box'.

Let us begin with global interpretability. We take the absolute mean of each feature's SHAP value as the feature's importance, resulting in a standard bar chart shown in Fig. 24. This indicates the general influences of features on different predictions. It can be observed from that, the thermal conductivity of the matrix has a significant effect on the macroscopic thermal conductivity of the composite material, while the dispersion index are the least important. Volume fraction also affects composite thermal conductivity, but with less importance. The effects from agglomeration and dispersion index on the result are relatively small. The SHAP summary plot, i.e., distribution of SHAP values for each feature is demonstrated in Fig. 25, where the corresponding influence trend is also indicated. It can better understand the overall models and allow the detection of predicted outliers. The color marks the feature value (red - high, blue - low). According to the color and SHAP value in Fig. 25, it can be seen that the thermal conductivity of graphene and Kapitza resistance have an effect on improving the predicted performance of the ML model. These plots give a global interpretations about how input feature variables affect predictive outputs.

In addition to the global interpretations, individual interpretations for each single sample also be provided by SHAP values. Figs. 26 and 27 are two selected typical samples which can be better illustrated. The basic idea in these figures is pushing the model's predictions from the base value to the final model output. The red bars indicates the features that contribute to increases in model output from the base value, whereas blue bars represent features that lead to decreases. The length of the bar presents the corresponding increases and decreases. Regarding sample 34, the aspect ratio and Kapitza resistance have positive effects on predicting the most critical features, whereas the agglomeration index has a negative effect on the final prediction. In terms of sample 23, the volume fraction and Kapitza resistance positively affect the final output of the data model while thermal matrix has a negative effect on the final prediction to some extent.

To understand how a single feature's value affects the final model's output, we can compare the SHAP value of that feature with the all

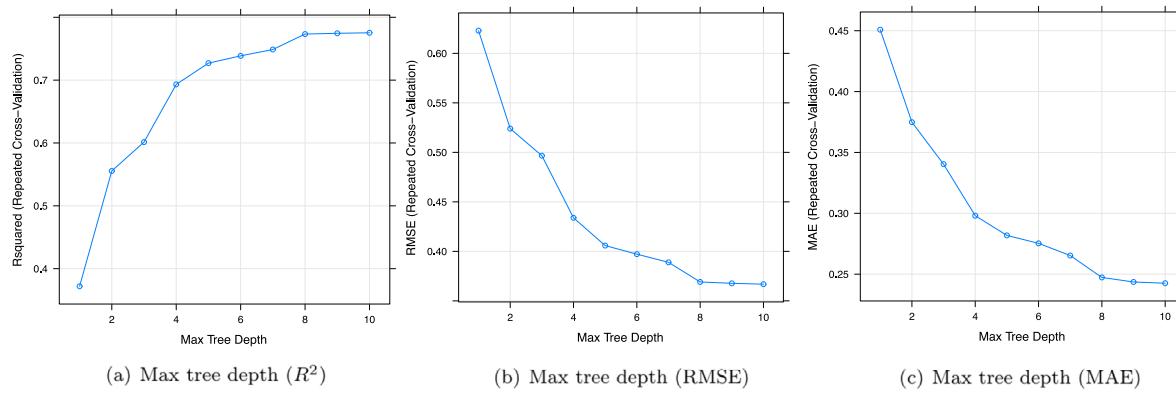


Fig. 14. Hyper-parameter tuning in Regression Tree (Max Tree Depth).

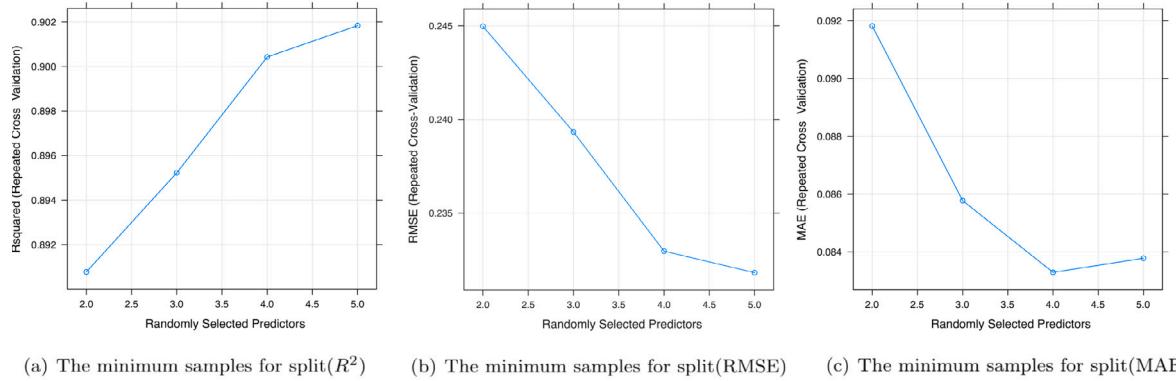


Fig. 15. Hyper-parameter tuning in Random Forest.

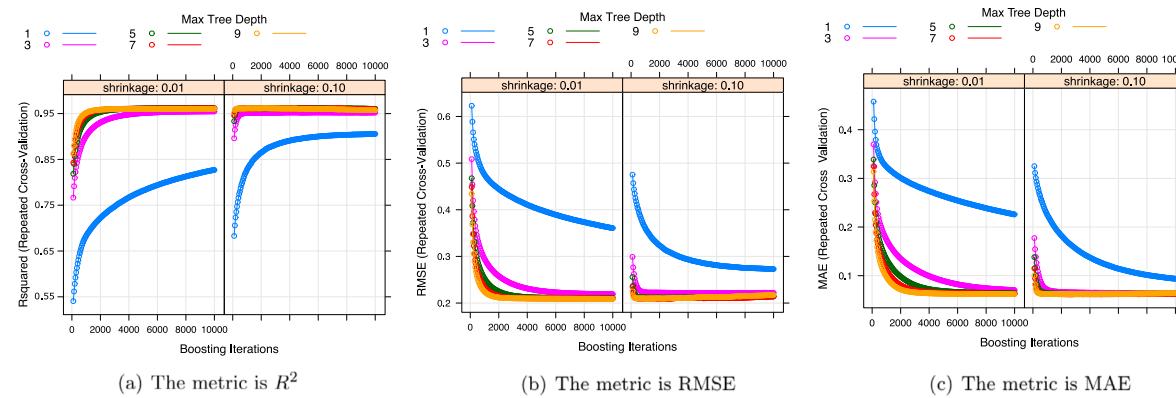


Fig. 16. Hyper-parameter tuning in GBM method with different metrics.

feature values of whole samples in the dataset. Fig. 28 shows the interaction between single feature and others. Among all of them, the data in Fig. 28(a) is the most concentrated, and its trend is relatively obvious. For relatively large aspect ratios, the increase of thermal conductivity of matrix has a negative interaction with its SHAP value, i.e., it reduces the influence on the final model outputs. High Kapitza resistance in Fig. 28(b) also has a negative SHAP value interaction with increasing volume ratio compared to other features. However, at higher values of matrix thermal conductivity - Fig. 28(d), the increase of Kapitza has a positive effect on its SHAP value. As same as volume ratio's interaction - Fig. 28(e), the increase of thermal conductivity of graphene has a positive effect on the prediction of the final model.

Since SHAP gives detailed impact rules of different features on composites' thermal conductivity, it can be applied to explore new structures of components for materials. These results/rules can also be used to devise a new form that will efficiently probe the sought-after behavior in an optimal manner.

6. Conclusions

We employ a data-driven multi-scale methodology to predict the thermal conductivity of Polymeric Graphene-Enhanced Composites. Our approach involves a hierarchical stochastic analysis that encompasses both meso- and macro-scales. This comprehensive data-driven modeling process comprises three distinct steps: bottom-up modeling,

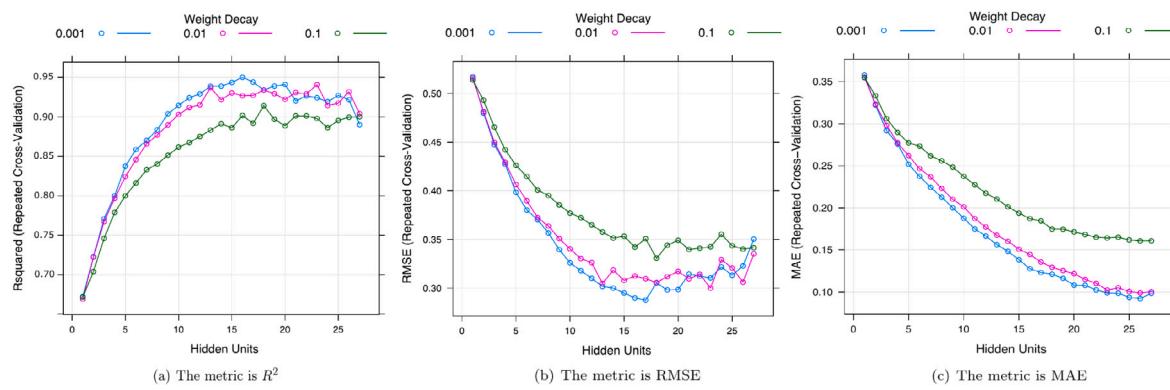


Fig. 17. Hyper-parameter tuning in ANN method with different metrics.

Table 4
Hyper-parameters tuning.

ML method	Hyper-parameters	Definition	Interval	Optimum value
Random Forest	C_p	Complexity parameter	[0,1]	0
	Min-sample-split	The minimum samples for split	[2,5]	5
	M	The maximum depth of tree	[1,10]	10
	Max-DT	The maximum numbers of regression tree	[100,10000]	10 000
Gradient Boosting Machine	λ	The learning rate	[0.001,0.99]	0.01
	N	The maximum number of regression tree	[100,10000]	10 000
Artificial Neural network (1 hidden layer)	D_{ia}	Interaction depth	[1,10]	9
	N_{hn}	The number of neurons in hidden layer	[1–100]	17
	λ	The learning rate	[0.001–1]	0.001
Deep Neural network (2 hidden layers)	N_{hn}	The number of neurons in hidden layer	[1–100]	[64,64]
	λ	The learning rate	[0.001–1]	0.001
	$Epoch_{max}$	The maximum epoch	[1–10000]	3000

Table 5
Predictive performance in ML models.

ML method	Training			Test		
	R^2	RMSE	MAE	R^2	RMSE	MAE
Random Forest	0.9018358	0.2318173	0.08378478	0.9277542	0.1550168	0.0469515
Gradient Boosting Machine	0.9104604	0.2091706	0.06196632	0.94626539	0.13263531	0.04831758
Artificial neural network	0.9439432	0.2875832	0.12312090	0.8332558	0.2371077	0.1157105
Deep neural network	0.9998	0.1420634	0.02745	0.95744352	0.16056436	0.05418494

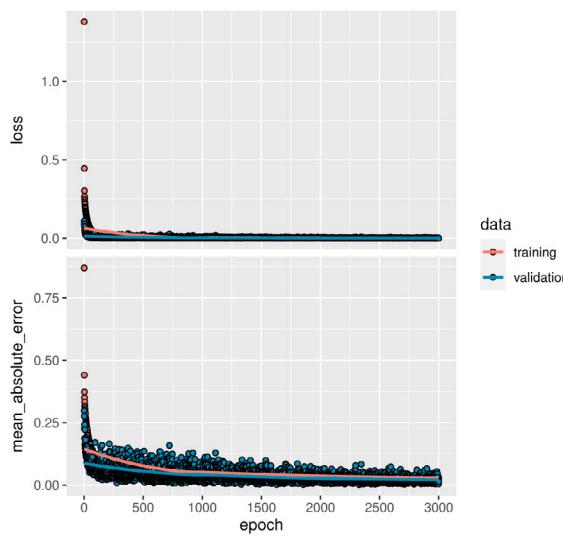


Fig. 18. The loss function w.r.t epoch in training with Tensorflow.

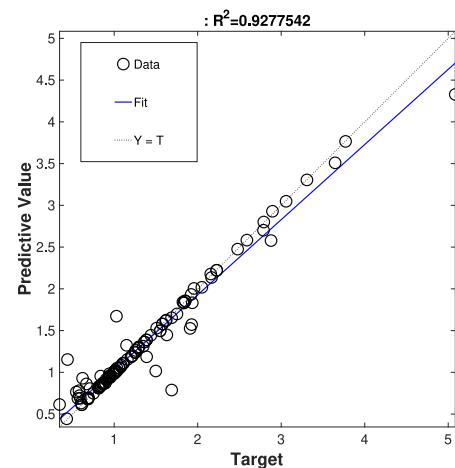


Fig. 19. The predictive performance of RF in test set.

stochastic modeling, and interpretable machine learning. From mesoscopic Representative Volume Elements (RVE) and macroscopic Finite Element Modeling (FEM), we carefully select seven critical effective parameters or features. These include the thermal conductivity of fillers, the thermal conductivity of the matrix, interface resistance, aspect

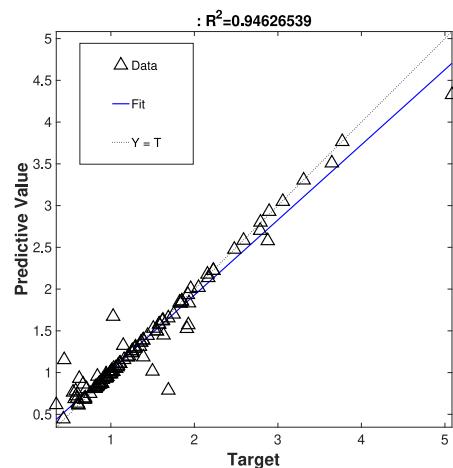


Fig. 20. The predictive performance of GBM in test set.

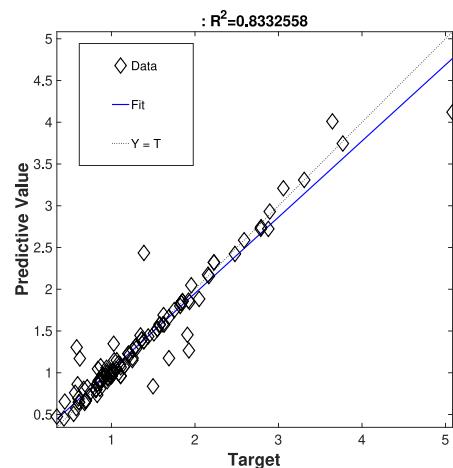


Fig. 21. The predictive performance of ANN in test set.

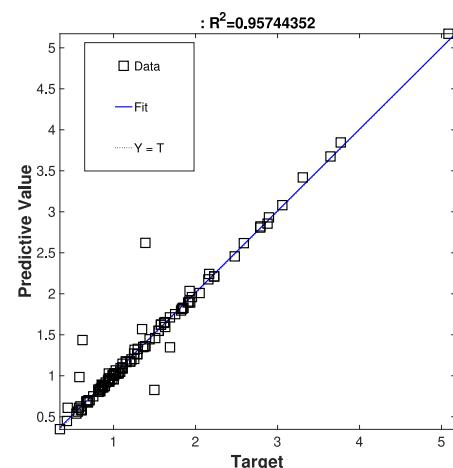


Fig. 22. The predictive performance of DNN in test set.

Table 6

Time complexity in ML models(n is the samples number).

ML method	Time complexity	Computation time
Regression Tree	$O(\log_2 n)$	18.321 s
Random Forest	$O(m(\log n))$	270.964 s
Gradient Boosting Machine	$O(mn\log n)$	3042.084 s
Artificial Neural network	$O(mn^2)$	12167.895 s
Deep Neural network	$O(n^3)$	2571 s (42 m 51 s)

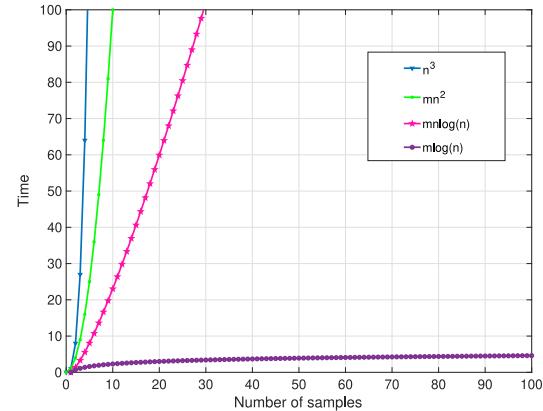


Fig. 23. The time complexity among ML models.

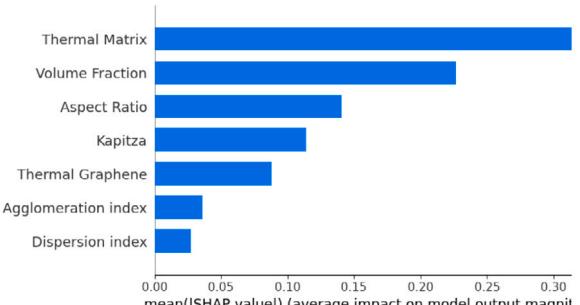


Fig. 24. Global interpretations by SHAP values (SHAP feature importance).

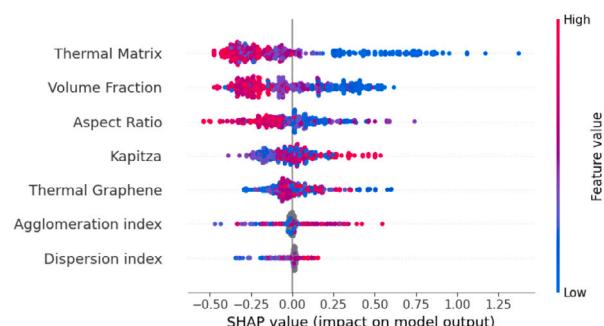


Fig. 25. Global interpretations by SHAP values (SHAP summary plot).

ratio, agglomeration index, dispersion index, and volume fraction. We rigorously quantify the uncertainties associated with these parameters based on their probability density functions (PDFs). To model and predict the thermal conductivity, we harness the power of integrated machine learning techniques, both tree-based (Random Forests and



Fig. 26. Individual interpretations for Sample 34.

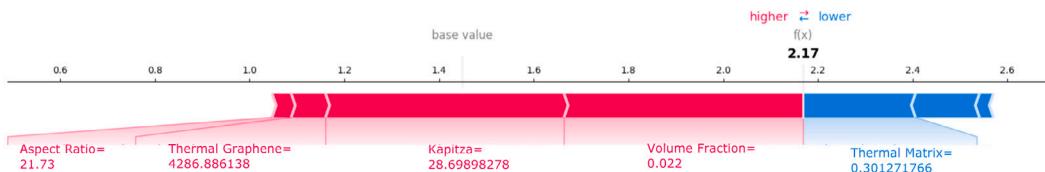


Fig. 27. Individual interpretations for Sample 23.

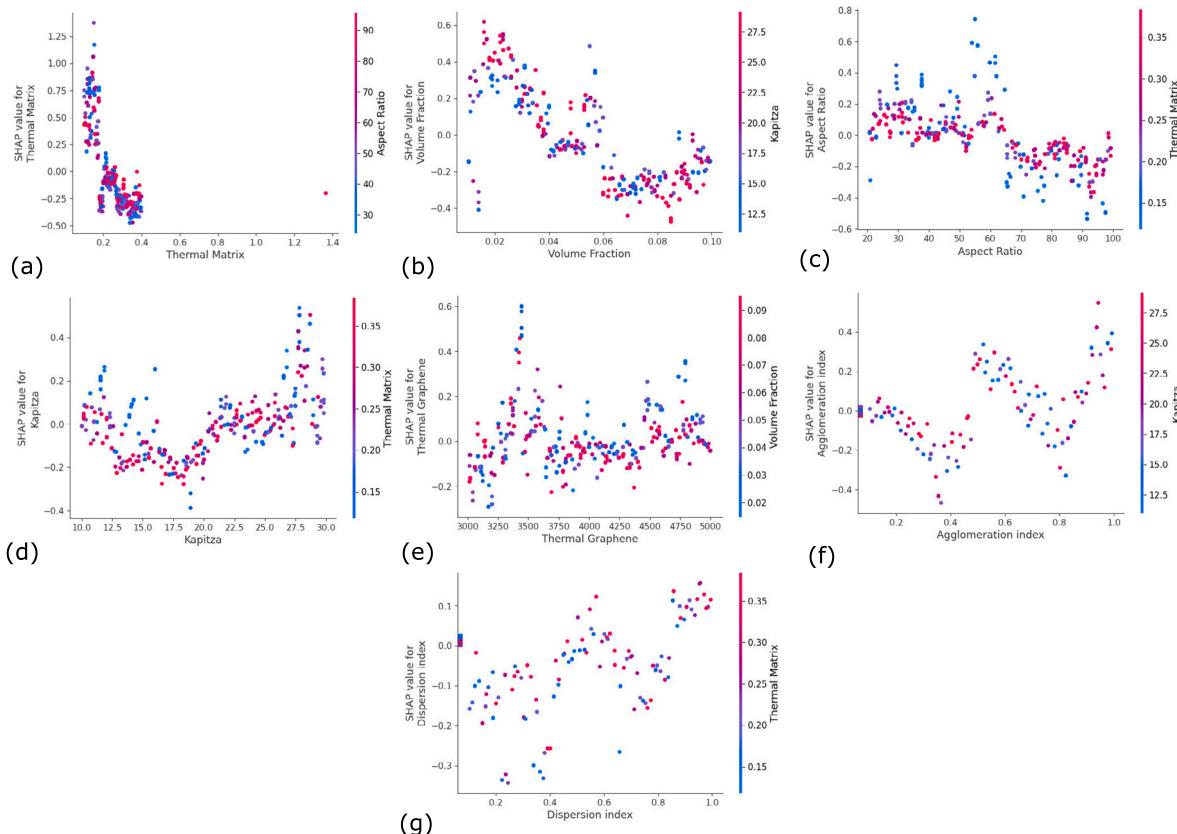


Fig. 28. Feature dependence plots.

Gradient Boost Machine) and neural network-based (Artificial Neural Networks and Deep Neural Networks). All of these machine learning approaches yield credible and reasonable output models.

Furthermore, we employ the SHAP model to provide insight into the mechanisms underlying our models and enhance interpretability. This includes offering global interpretations, individual interpretations, and depicting feature dependencies separately. By utilizing SHAP values, we elucidate the inner workings of the so-called 'black box' in a visually accessible and comprehensible manner.

This data-driven method can significantly reduce the requirements for analytical modeling and simulation in materials structural design, and it can also considerably reduce computational cost than previous multi-scale stochastic modeling. The following conclusion can be summarized:

1. PSO and 10-fold CV can significantly find the global optimal on hyper-parameter tuning.

2. Each ML model has good performance on predictive outputs. The Deep neural networks (DNN) is the most accurate one but also computationally most expensive. RF has relative low accuracy but lower time complexity in terms of computation time.

3. SHAP is demonstrated to provide both global and local interpretations for the outputs predictions. At the global level, it not only produce feature importance but also define the distribution of Shapley values.

4. In terms of global interpretations by SHAP values, the thermal conductivity of the matrix has a significant effect on the macroscopic thermal conductivity, then comes to volume fraction, while the dispersion index are the least important.

5. At the local level, the final prediction of a specific sample is decomposed into base value and contributions by each feature, and thus the effects of those individual features could be quantified in a visible manner.

6. The variation of effects for each feature on the macroscopic thermal conductivity against its value is also quantified by SHAP. These results provide detailed and intuitive insights into how they work. This information from the outputs can be applied for searching optimal value ranges for the single features and their different combinations for developing the composite structural design.

CRediT authorship contribution statement

Bokai Liu: Formal analysis, Validation, Visualization, Methodology, Writing – original draft, Writing – review & editing. **Weizhuo Lu:** Methodology, Supervision, Project administration. **Thomas Olofsson:** Methodology, Supervision. **Xiaoying Zhuang:** Methodology, Supervision. **Timon Rabczuk:** Methodology, Conceptualization, 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.

Data availability

Data will be made available on request.

Acknowledgments

We gratefully acknowledge the support of the EU project H2020-AURORAL Grant agreement ID: 101016854 (Architecture for Unified Regional and Open digital ecosystems for Smart Communities and Rural Areas Large scale application) and the Kempe Foundation Sweden (Kempestiftelsen - Stiftelsen J.C. Kemps och Seth M. Kemps minne). This work is also funded and supported by J. Gust. Richert stiftelse, SWECO, Sweden (Grant agreement ID: 2023-00884).

The computations handling were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) and Academic Infrastructure for Supercomputing in Sweden (NAIIS) at High-Performance Computing Center North (HPC2N) partially funded by the Swedish Research Council through grant agreement no. 2018-05973 and no. 2022-06725.

References

- [1] Stankovich S, Dikin DA, Domke GH, Kohlhaas KM, Zimney EJ, Stach EA, et al. Graphene-based composite materials. *Nature* 2006;442(7100):282–6.
- [2] Huang X, Qi X, Boey F, Zhang H. Graphene-based composites. *Chem Soc Rev* 2012;41(2):666–86.
- [3] Balandin AA, Ghosh S, Bao W, Calizo I, Teweldebrhan D, Miao F, et al. Superior thermal conductivity of single-layer graphene. *Nano Lett* 2008;8(3):902–7.
- [4] Ghosh D, Calizo I, Teweldebrhan D, Pokatilov EP, Nika DL, Balandin AA, et al. Extremely high thermal conductivity of graphene: Prospects for thermal management applications in nanoelectronic circuits. *Appl Phys Lett* 2008;92(15):151911.
- [5] Mortazavi B, Madjet ME, Shahrokh M, Ahzi S, Zhuang X, Rabczuk T. Nanoporous graphene: A 2D semiconductor with anisotropic mechanical, optical and thermal conduction properties. *Carbon* 2019;147:377–84.
- [6] Mortazavi B, Baniassadi M, Bardon J, Ahzi S. Modeling of two-phase random composite materials by finite element, Mori-Tanaka and strong contrast methods. *Composites B* 2013;45(1):1117–25.
- [7] Fish J, Wagner GJ, Keten S. Mesoscopic and multiscale modelling in materials. *Nat Mater* 2021;20(6):774–86.
- [8] Wilk J, Smusz R, Filip R, Chmiel G, Bednarczyk T. Experimental investigations on graphene oxide/rubber composite thermal conductivity. *Sci Rep* 2020;10(1):15533.
- [9] Selvam C, Lal DM, Harish S. Thermal conductivity enhancement of ethylene glycol and water with graphene nanoplatelets. *Thermochim Acta* 2016;642:32–8.
- [10] Song N, Cao D, Luo X, Wang Q, Ding P, Shi L. Highly thermally conductive polypropylene/graphene composites for thermal management. *Composites A* 2020;135:105912.
- [11] Rafiee R, Eskandariyan A. Estimating Young's modulus of graphene/polymer composites using stochastic multi-scale modeling. *Composites B* 2019;173:106842.
- [12] Tran V-T, Nguyen T-K, Nguyen-Xuan H, Abdel Wahab M. Vibration and buckling optimization of functionally graded porous microplates using BCMO-ANN algorithm. *Thin-Walled Struct* 2023;182:110267. <http://dx.doi.org/10.1016/j.tws.2022.110267>, URL <https://www.sciencedirect.com/science/article/pii/S0263823122008199>.
- [13] Dang B-L, Nguyen-Xuan H, Abdel Wahab M. An effective approach for VARANS-VOF modelling interactions of wave and perforated breakwater using gradient boosting decision tree algorithm. *Ocean Eng* 2023;268:113398. <http://dx.doi.org/10.1016/j.oceaneng.2022.113398>, URL <https://www.sciencedirect.com/science/article/pii/S0029801822026816>.
- [14] Wang S, Wang H, Zhou Y, Liu J, Dai P, Du X, et al. Automatic laser profile recognition and fast tracking for structured light measurement using deep learning and template matching. *Measurement* 2021;169:108362. <http://dx.doi.org/10.1016/j.measurement.2020.108362>, URL <https://www.sciencedirect.com/science/article/pii/S0263224120308988>.
- [15] Shokrieh MM, Rafiee R. Stochastic multi-scale modeling of CNT/polymer composites. *Comput Mater Sci* 2010;50(2):437–46.
- [16] Vu-Bac N, Lahmer T, Zhang Y, Zhuang X, Rabczuk T. Stochastic predictions of interfacial characteristic of polymeric nanocomposites (PNCs). *Composites B* 2014;59:80–95.
- [17] Liu B, Vu-Bac N, Zhuang X, Rabczuk T. Stochastic multiscale modeling of heat conductivity of Polymeric clay nanocomposites. *Mech Mater* 2020;142:103280. <http://dx.doi.org/10.1016/j.mechmat.2019.103280>, URL <https://www.sciencedirect.com/science/article/pii/S0167663619308579>.
- [18] Liu B, Vu-Bac N, Zhuang X, Fu X, Rabczuk T. Stochastic integrated machine learning based multiscale approach for the prediction of the thermal conductivity in carbon nanotube reinforced polymeric composites. *Compos Sci Technol* 2022;224:109425.
- [19] Liu B, Vu-Bac N, Rabczuk T. A stochastic multiscale method for the prediction of the thermal conductivity of Polymer nanocomposites through hybrid machine learning algorithms. *Compos Struct* 2021;273:114269.
- [20] Liu B, Vu-Bac N, Zhuang X, Fu X, Rabczuk T. Stochastic full-range multiscale modeling of thermal conductivity of polymeric carbon nanotubes composites: A machine learning approach. *Compos Struct* 2022;289:115393.
- [21] Liu B, Vu-Bac N, Zhuang X, Lu W, Fu X, Rabczuk T. Al-DeMat: A web-based expert system platform for computationally expensive models in materials design. *Adv Eng Softw* 2023;176:103398.
- [22] Liu B, Li W. Surrogate models in machine learning for computational stochastic multi-scale modelling in composite materials design. *Int J Hydromech* 2022;5(4):336–65.
- [23] Thrampoloudis E, Mavromatis G, Lucchi A, Oreounig K. A machine learning-based surrogate model to approximate optimal building retrofit solutions. *Appl Energy* 2021;281:116024.
- [24] Xia Y, Zhang C, Wang C, Liu H, Sang X, Liu R, et al. Prediction of bending strength of glass fiber reinforced methacrylate-based pipeline UV-CIPP rehabilitation materials based on machine learning. *Tunn Undergr Space Technol* 2023;140:105319. <http://dx.doi.org/10.1016/j.tust.2023.105319>, URL <https://www.sciencedirect.com/science/article/pii/S0886779823003395>.
- [25] Guo H, Zhuang X, Chen P, Alajlan N, Rabczuk T. Stochastic deep collocation method based on neural architecture search and transfer learning for heterogeneous porous media. *Eng Comput* 2022;1–26.
- [26] Nguyen-Le DH, Tao Q, Nguyen V-H, Abdel-Wahab M, Nguyen-Xuan H. A data-driven approach based on long short-term memory and hidden Markov model for crack propagation prediction. *Eng Fract Mech* 2020;235:107085.
- [27] Khatir S, Bouchicha D, Le Thanh C, Tran-Ngoc H, Nguyen T, Abdel-Wahab M. Improved ANN technique combined with Jaya algorithm for crack identification in plates using XIGA and experimental analysis. *Theor Appl Fract Mech* 2020;107:102554.
- [28] Zenzen R, Khatir S, Belaidi I, Le Thanh C, Wahab MA. A modified transmissibility indicator and artificial neural network for damage identification and quantification in laminated composite structures. *Compos Struct* 2020;248:112497.
- [29] Huang J, Liew J, Liew K. Data-driven machine learning approach for exploring and assessing mechanical properties of carbon nanotube-reinforced cement composites. *Compos Struct* 2021;267:113917.
- [30] Tran-Ngoc H, Khatir S, Le-Xuan T, De Roeck G, Bui-Tien T, Wahab MA. A novel machine-learning based on the global search techniques using vectorized data for damage detection in structures. *Internat J Engrg Sci* 2020;157:103376.
- [31] Arrieta AB, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, et al. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf Fusion* 2020;58:82–115.

- [32] Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. *Adv Neural Inf Process Syst* 2017;30.
- [33] Covert IC, Lundberg S, Lee S-I. Explaining by removing: A unified framework for model explanation. *J Mach Learn Res* 2021;22(1):9477–566.
- [34] Iman RL, Conover W. Small sample sensitivity analysis techniques for computer models. with an application to risk assessment. *Comm Statist Theory Methods* 1980;9(17):1749–842.
- [35] Novák D, Teplý B, Keršner Z. The role of Latin Hypercube Sampling method in reliability engineering. In: Proc. of ICOSSAR, vol. 97. 1998, p. 403–9.
- [36] Keitel H. Bewertungsmethoden für die Prognosequalität von Kriechmodellen des Betons [Ph.D. thesis]. Bauhaus-Universität Weimar, 2012, p. 390. <http://dx.doi.org/10.25643/bauhaus-universitaet.1556>.
- [37] Ghosh S, Bao W, Nika DL, Subrina S, Pokatilov EP, Lau CN, et al. Dimensional crossover of thermal transport in few-layer graphene. *Nat Mater* 2010;9(7):555.
- [38] Moisala A, Li Q, Kinloch I, Windle A. Thermal and electrical conductivity of single-and multi-walled carbon nanotube-epoxy composites. *Compos Sci Technol* 2006;66(10):1285–8.
- [39] Freitag M, Steiner M, Martin Y, Perebeinos V, Chen Z, Tsang JC, et al. Energy dissipation in graphene field-effect transistors. *Nano Lett* 2009;9(5):1883–8.
- [40] Bui K, Grady BP, Papavassiliou DV. Heat transfer in high volume fraction CNT nanocomposites: Effects of inter-nanotube thermal resistance. *Chem Phys Lett* 2011;508(4–6):248–51.
- [41] Breiman L. Random forests. *Mach Learn* 2001;45(1):5–32.
- [42] Kuhn M, Johnson K, et al. Applied predictive modeling, vol. 26. Springer; 2013.
- [43] Friedman JH. Greedy function approximation: a gradient boosting machine. *Ann. Statist.* 2001;1189–232.
- [44] Jain AK, Mao J, Mohiuddin KM. Artificial neural networks: A tutorial. *Computer* 1996;29(3):31–44.
- [45] Molnar C. Interpretable machine learning. Lulu. com; 2020.
- [46] Murdoch WJ, Singh C, Kumbier K, Abbasi-Asl R, Yu B. Definitions, methods, and applications in interpretable machine learning. *Proc Natl Acad Sci* 2019;116(44):22071–80.
- [47] Slack D, Hilgard S, Jia E, Singh S, Lakkaraju H. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In: Proceedings of the AAAI/ACM conference on AI, ethics, and society. 2020, p. 180–6.
- [48] Lundberg SM, Erion GG, Lee S-I. Consistent individualized feature attribution for tree ensembles. 2018, arXiv preprint [arXiv:1802.03888](https://arxiv.org/abs/1802.03888).
- [49] Štrumbelj E, Kononenko I. Explaining prediction models and individual predictions with feature contributions. *Knowl Inf Syst* 2014;41:647–65.
- [50] Song SH, Park KH, Kim BH, Choi YW, Jun GH, Lee DJ, et al. Enhanced thermal conductivity of epoxy-graphene composites by using non-oxidized graphene flakes with non-covalent functionalization. *Adv Mater* 2013;25(5):732–7.
- [51] Wang F, Drzal LT, Qin Y, Huang Z. Mechanical properties and thermal conductivity of graphene nanoplatelet/epoxy composites. *J Mater Sci* 2015;50:1082–93.
- [52] Wan Y-J, Tang L-C, Gong L-X, Yan D, Li Y-B, Wu L-B, et al. Grafting of epoxy chains onto graphene oxide for epoxy composites with improved mechanical and thermal properties. *Carbon* 2014;69:467–80.
- [53] Chen J, Chen X, Meng F, Li D, Tian X, Wang Z, et al. Super-high thermal conductivity of polyamide-6/graphene-graphene oxide composites through in situ polymerization. *High Perform Polym* 2017;29(5):585–94.
- [54] Kapadia RS, Louie BM, Bandaru PR. The influence of carbon nanotube aspect ratio on thermal conductivity enhancement in nanotube–polymer composites. *J Heat Transfer* 2014;136(1).