Alexandria Engineering Journal 86 (2024) 144–156

Contents lists available at ScienceDirect

Alexandria Engineering Journal

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

Predicting the critical superconducting temperature using the random forest, MLP neural network, M5 model tree and multivariate linear regression

Paulino José García Nieto [a], Esperanza García Gonzalo [a], Luis Alfonso Menéndez García [b], Laura Álvarez–de Prado [b,*], Antonio Bernardo Sánchez [b]

[a] Department of Mathematics, Faculty of Sciences, University of Oviedo, 33007 Oviedo, Spain
[b] Department of Mining Technology, Topography and Structures, Higher and Technical School of Mining Engineering, University of León, Campus de Vegazana s/n, 24071 León, Spain

ARTICLE INFO

Keywords:
Critical superconducting temperature
Random forest regression (RFR) technique
Artificial neural networks (ANNs)
M5 model tree
Multivariate linear regression (MLR)

ABSTRACT

Using a random forest regression (RFR) machine learning technique, the critical temperature (T_c) of a superconductor was predicted in the context of Industry 4.0 in this study using features derived from the material's physico-chemical properties, containing atomic mass, electron affinity, atomic radius, valence, and thermal conductivity. The same experimental data were also fitted with multilayer perceptron (MLP) artificial neural networks (ANN), M5 model tree and multivariate linear regression (MLR) model for comparison. The current investigation's findings show that the proposed RFR–relied model can successfully forecast the critical temperature of a superconductor. Additionally, the T_c estimate was reached with a correlation coefficient of 0.9565 and a coefficient of determination 0.9146, when the observed dataset was used to test this unique technique. Additionally, the outcomes from the MLP, M5, and MLR models are obviously worse than those from the RFR–relied model. When it comes to fully comprehending the superconductivity, this investigation is noteworthy. Regarding forecasting effectiveness and feature reduction rate, the RFR approach has obvious advantages and generalizability, and it also demonstrates suitability for high-temperature superconductor T_c forecasting. In fact, it offers a practical and affordable approach to data-driven superconductor investigation.

1. Introduction

Superconducting materials (these materials have zero resistance, so electricity can easily flow through them) can be used in many practical ways [1–4]. The most well-known use is in Magnetic Resonance Imaging (MRI) systems because MRI equipment allows medical professionals to view into patients' bodies in great detail. Some common uses for superconducting magnets include keeping the Large Hadron Collider's strong magnetic fields at CERN as well as the use of the sensitive magnetic sensors to measure things like the Earth's magnetic field (employing devices termed SQUIDS). Therefore, superconductors can help revolutionize the energy industry by making it possible to transport electricity without any loss of energy.

A superconductor can only conduct electricity without resistance, i.e., with zero resistance, at or below the critical superconducting temperature (T_c), as reported in [5–9]. Although the exact mechanism is still unknown, it is believed that the structures and some characteristics of the material like valency properties, bond lengths, and the Coulomb coupling between electronic bands determines the conductive properties. Data-driven methods allow learning from known superconductors and linking the characteristics of the material with its conductive properties and the critical temperature. Here, we adopt a wholly data-driven strategy to develop a statistical model that foretells T_c for a specific chemical formula of a material in the absence of any theory-based prediction models. Indeed, Machine learning (ML) approximation can be an alternative way to forecast the superconducting critical temperature, which builds data-driven predictive models to figure out how materials' composition and critical temperatures work together.

Machine learning (MLT) employ a lot of training data tow work well and has emerged as an important tool for predicting the critical temperatures (T_c) of superconductors, offering the possibility of design and

* Corresponding author.
E-mail address: laura.alvarez@unileon.es (L.Á. Prado).

https://doi.org/10.1016/j.aej.2023.11.034
Received 29 June 2023; Received in revised form 27 September 2023; Accepted 8 November 2023
Available online 28 November 2023

discovery of materials. Several existing works have explored this application, leveraging various machine learning techniques and datasets. Common machine learning methods include Linear Regression, LASSO Regression [10], Ridge Regression [11], Support Vector Regression (SVR) [12–14], Random Forest [15], Decision Tree [16], Elastic-net [17], XGBoost [18] and so on. These machine learning methods can be combined with intelligent optimization methods like Particle Swarm Optimization (PSO) [19,20]. In this subsection, we will review existing works focusing on building more effective feature extraction methods and using different machine learning methods to regress critical temperature. Zhang et al. [21] proposed an RS–PSO–SVR prediction model, combining Rough Set (RS) theory [22], PSO, and SVR methods. PSO is used to determine the critical parameters in SVR, including regularized constant C, and the kernel function parameter γ. RS preprocessing algorithm is used to calculate the weight of each feature. The vector of the distance between interacting layers ζ and the calculated spacing between interacting charges within layers l is the input of the RS-PSO-SVR prediction model. Back propagation neural network (BPNN) [23] is used as a baseline. Similarly, Liu et al. [24] proposed a PCA–PSO–SVR method, combining principal component analysis (PCA), PSO, and SVR methods. The feature vectors are established by the PCA method, which calculates the eigenvalues of the covariance matrix of the dataset, and selects the determined number of top eigenvalues of all the eigenvalues. Stanev et al. [25] built a classification firstly to separate materials into two distinct groups depending on whether T_c is above or below a threshold temperature T_{sep}. Random Forest and its variant methods are used to predict T_c. The Materials Agnostic Platform for Informatics and Exploration (Magpie) [26] was employed to calculate a set of attributes for each material like electronic structure attributes and elemental property statistics. Matsumoto et al. [27] calculated the mean value and deviation, and standard deviation for each composition in element groups to build 53 descriptors as input features. The machine learning method used is also Random Forest regression. Roter et al. [28] used Fine Tree, Exponential Gaussian Process Elimination, a Gaussian Support Vector Machine (SVM) and Boosted Tree for critical temperature regression. The Bagged Tree method best predicted the values of T_c. The element-vectors input is the chemical composition matrix to represent chemical content. The authors argued that predictors such as electronegativity, the number of valence electrons, covalent radius or electron affinity are not directly relevant to superconductivity. Gaikwad et al. [29] used chemical formula from the atomic table directly as input and applied Random Forest, Decision Tree, Bayes Model, Linear Regression, Decision Tree PCA, SVR, XGBoost, and SVMRBF methods for regression. García–Nieto et al. [30] used a hybrid regressive model combining the multivariate adaptive regression splines (MARS) approximation [31] with the whale optimization algorithm (WOA) [32] for prediction. The Lasso, Ridge, and Elastic-net regression models were used as baselines. Zhang et al. [33] developed the Gaussian process regression method, a nonparametric kernel-based probabilistic model, for doped Fe-based superconductor critical temperature prediction from structural and topological parameters, and they also applied the Gaussian process regression model to a wider variety of superconductor families [34]. Revathy et al. [35] utilized fie, radius, atomic mass, density, fusion heat, electron affinity, the valence electron, thermal conductivity and critical temperature. Random Forest Regressor, XGBoost Regressor, Artificial Neural Networks, Support Vector Regressor, Decision Tree Regressor, AdaBoost Regressor, Gradient Boosting Regressor, and Simple Linear Regressor are used for training and testing.

Estimating the critical temperature (T_c) of superconductors is a complex and important area of research, with several notable research gaps and challenges:

Complexity of High-Temperature Superconductors (HTS): Most superconductors of technological interest are HTS, and predicting their T_c accurately remains a challenge. These materials often have complex crystal structures, multiple elements, and unconventional pairing mechanisms, making it difficult to develop predictive models.

Doping and Defects: Doping and the presence of defects can significantly affect the T_c of superconductors. Understanding how different types and concentrations of dopants or defects impact Tc is an ongoing research area.

Influence of Multiband Effects: Some superconductors have multiple electron bands contributing to superconductivity. Understanding how these multiband effects impact T_c is a research gap.

Tailored Material Design: Researchers are interested in designing superconducting materials with specific T_c values for different applications. Developing methods to tailor T_c in a controlled manner is a research challenge.

Emerging Superconductors: Discovering and predicting T_c in new, unconventional superconducting materials is an ongoing area of research, with the potential for transformative applications.

While machine learning methods have been applied to predict T_c, there's a need for more robust and accurate models.

Advancements in understanding and estimating the critical temperature of superconductors will not only deepen our understanding of these materials but also enable breakthroughs in various technological applications.

The objective of this study is to obtain a predictive model for estimating the critical temperature (T_c) of a semiconductor material based on its intrinsic properties and external factors. The modeling goal is the minimization of the estimation error between the predicted T_c and the actual critical temperature. The modeling study involves preprocessing relevant data, selecting an appropriate model, training and validating the model, and iteratively refining it to improve predictive accuracy. Model performance can be assessed through cross-validation and testing against unseen data. Ultimately, the developed model should serve as a valuable tool for predicting critical temperatures in semiconductor materials, aiding in materials research and engineering applications.

In this study, the superconducting critical temperature T_c has been accurately predicted for various types of superconductors using a unique regressive model relied on the Random Forest Regression (RFR) approach. This method, the RFR approximation [36–40] in conjunction with the optimizer known as Grid Search (GS) [41–45], could be an attractive methodology to tackle this kind of high-nonlinear problems. Fundamentally, GS is an optimization technique that enables choosing the best parameters from a list of optional parameters for a problem optimization. Machine learning models typically have various parameters that impact their ability to learn and generalize from data, so optimizing these parameters lead to a more accurate and effective model. For comparative purposes, the MLP, M5 model tree and multivariate linear regression (MLR) models were also adjusted to the same experimental dataset both to calculate the T_c and contrast the outcomes found [46–54]. To cope with nonlinearities, including interactions between variables, the RFR approach is a statistical learning procedure that was developed conforming to statistics and mathematical analysis. It is a prolongation of linear models that mechanically models complex relationships between variables and nonlinearities. Comparing RFR approach to traditional and metaheuristic regression approaches, several advantages are apparent: (1) it is one of the most precise learning algorithms obtainable. Indeed, for a large enough dataset, it produces a very accurate regressor; (2) it can operate effectively on huge databases; (3) it is capable of handling hundreds of input variables without excluding any; (4) it provides estimates of the key variables in regression; (5) it permits to elude physical models of the superconductor; and (6) it makes possible to model nonlinear interactions between the physico-chemical input variables of the superconductor. Also, prior research has shown that RFR is a highly useful tool for a variety of practical applications, such as determining the temperature of the near-surface air in glacier zones [55], the mechanical properties of γ −TiAl alloys [56], erodibility of treated unsaturated lateritic soil [57], neighborhood environment's impact on peer-to-peer accommodation [58], etc. For the majority of superconductors, including high-temperature superconductors, it has only sometimes been utilized

to calculate the superconducting critical temperature T_c from the input physicochemical characteristics.

The organization of this article is as follows: the experimental design, all the variables used in this study, and the RFR, MLP, M5 model tree, and MLR techniques are all presented in Section 2; by compiling the RFR outcomes with the experimental values and the relevance order of the input parameters, Section 3 offers the insights gained with this intriguing technique, and Section 4 finishes this study by presenting a summary of the investigation's key findings.

2. Materials and methods

2.1. Experimental dataset

The world's largest and most complete database of superconductors is the SuperCon database [59]. Hamidieh [7] performed the processing on the SuperCon dataset so that this could be used for further research. Nowadays, the database is stored at the University of California-Irvine library's data storage place [60]. The dataset's pre-treatment eliminated materials with missing features. Preliminary processing also involved building new features on top of old ones. The first eight features were assumed to be the atomic mass, density, first ionization energy, atomic radius, density, electron affinity, fusion heat, thermal conductivity, and valence (see Table 1). In fact, the statistical parameters from the following characteristics— mean, geometric mean, weighted mean, standard deviation and weighted standard deviation, range, weighted range, as well as entropy and weighted entropy, —were used to derive the chemical formula for each substance (see Table 2). In this manner, 80 characteristics (8 × 10) are obtained. The superconductor's composition in terms of elements is another additional feature that is extracted (numeric variable). As a result, we have a dataset with 83 columns and 81 features: 1 column has information about the material, including its name and identification number, the last column contains the critical temperature (T_c) values for each material, and the first 81 columns correspond to the various attributes that have been extracted. The dataset includes details on each of the 21,263 superconductors. Each material has 82 numerically based properties. The model that predicts the critical temperature (T_c) (dependent variable) uses the 81 features that were retrieved from the data as input variables (independent predictors). This approach to figuring out how features form in materials is very general and can be used to study superconducting materials. This happens as a result of the critical temperature's ambiguous dependence.

2.2. Random forest (RF) approach

A method for lowering an estimated prediction function's variance is bootstrap aggregation or bagging [38–40]. In particular, trees and other high-variance, low-bias techniques seem to benefit from bagging. The bootstrap-sampled versions of the training data are used to fit the same regression tree repeatedly, and the outcomes are averaged. Bagging has been significantly modified by random forests [15,36–40], which aggregates a sizable group of de-correlated trees, and then averages them. A multitude of decision trees are constructed during the training step of the random forests (RF) ensemble learning method, which can be used for classification, regression, and other tasks. The average prediction of each individual tree is provided when focusing on the regression issue. Random forests perform better overall than decision trees because they compensate for decision trees' propensity to overfit their training dataset.

Trees are excellent candidates for bagging because, when developed deeply enough, they have relatively little bias and can catch complicated interaction structures in the data. Trees gain a lot from the averaging because they are known to be noisy. Moreover, since each tree formed in bagging is identically distributed (i.d.), the expectation of an average of B such trees is the same as the expectation of any one of them. Hence, the bias of bagged trees is identical to that of the individual trees, and the only way to improve is by reducing the variation. An average of B i.i.d. random variables, each with variance σ^2, has variance $\frac{1}{B}\sigma^2$. If the variables are simply i.d. (identically distributed, but not necessarily independent) with positive pairwise correlation ρ, the variance of the average is [15,36–40]:

$$\rho\,\sigma^2 + \frac{1-\rho}{B}\sigma^2 \tag{1}$$

The advantages of averaging are constrained by the size of the correlation between bagged trees pairs since when B rises, the second term vanishes but the first one stays. By lowering the correlation between the trees, random forests (see algorithm below) aim to improve variance reduction of bagging without substantially raising variance. This is accomplished during the tree-growing process by selecting the input variables at random. In particular, choose $m \le p$ input variables at random as candidates for splitting while constructing a tree on a bootstrapped dataset. In most cases, m values are $p/3$ or even 1. The random forest regression predictor is the following after B such trees $\{T(x;\Theta_b)\}_1^B$ have grown [36–40]:

Table 1
The physico-chemical characteristics of an element used to construct its features with the purpose of foretelling T_c.

Variable	Units	Description
Atomic Mass	Atomic mass units (AMU)	Total proton and neutron rest masses
First Ionization Energy	Kilo-Joules per mole (kJ/mol)	Energy required to remove a valence electron
Atomic Radius	Picometer (pm)	Calculated atomic radius
Density	Kilograms per meters cubed (kg/m^3)	Density at standard temperature and pressure
Electron Affinity	Kilo-Joules per mole (kJ/mol)	Energy required to add an electron to a neutral atom
Fusion Heat	Kilo-Joules per mole (kJ/mol)	Energy to change from solid to liquid without temperature change
Thermal Conductivity	Watts per meter-Kelvin (W/(m K))	Thermal conductivity coefficient κ
Valence	No units	Typical number of chemical bonds formed by the element

Table 2
Description of the steps involved in extracting features from a material's chemical composition. (In the last column, attributes for Re6Zr1 that rely on thermal conductivities are calculated and given to two decimal places as an illustration; the thermal conductivity coefficients for rhenium and zirconium are given by $t_1 = 48$ and $t_2 = 23$ W/(m K), each in order. Thus: $p_1 = \frac{6}{7}$; $p_2 = \frac{1}{7}$; $w_1 = \frac{48}{71}$; $w_2 = \frac{23}{71}$; $A = \frac{p_1 w_1}{p_1 w_1 + p_2 w_2} \approx 0.926$; $B = \frac{p_2 w_2}{p_1 w_1 + p_2 w_2} \approx 0.074$).

Feature and description	Formula	Sample value (Re_6Zr_1)
Mean	$\mu = (t_1 + t_2)/2$	35.5
Weighted mean	$\nu = (p_1 t_1) + (p_2 t_2)$	44.43
Geometric mean	$= \sqrt{t_1 t_2}$	33.23
Weighted geometric mean	$= (t_1)^{p_1}(t_2)^{p_2}$	43.21
Entropy	$= -w_1 \ln(w_1) - w_2 \ln(w_2)$	0.63
Weighted entropy	$= -A \ln(A) - B \ln(B)$	0.26
Range	$= t_1 - t_2 \ (t_1 > t_2)$	25
Weighted range	$= p_1 t_1 - p_2 t_2$	37.86
Standard deviation	$= [(1/2)((t_1-\mu)^2 + (t_2-\mu)^2)]^{\frac{1}{2}}$	12.5
Weighted standard deviation	$= [p_1(t_1-\nu)^2 + p_2(t_2-\nu)^2]^{\frac{1}{2}}$	8.75

$$\widehat{f}_{rf}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T(x;\Theta_b) \tag{2}$$

where Θ_b describes the split variables, cut points at each node, and terminal-node values of the bth random forest tree. It seems sense that lowering m would lower the correlation between any two trees in the ensemble and, consequently, lower the variance of the average using Eq. (1). In the case of regression, we will utilize Eq. (2) to obtain a foretelling at a new point x: $\widehat{f}_{rf}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$.

An illustration of this algorithm is shown in Fig. 1.

2.3. Neural network: Multilayer perceptron

The inability of the simple perceptron and ADALINE (adaptative linear element) to resolve nonlinear issues (such as XOR) was demonstrated by Minsky and Papert in 1969 [61]. The *Generalized Delta Rule* (GDL), which Rumelhart and other investigators first introduced in 1996 [62], allows weights to be adjusted by propagating errors backwards, or toward the lower hidden layers. Working with numerous layers and nonlinear activation functions is conceivable in this fashion. It is possible to demonstrate the universal approximator nature of this multilayer perceptron (MLP) [46–49,62]. An approximate nonlinear relationship between input and output data can be found using a *multilayer perceptron*.

The MLP is a type of multiple-layer artificial neural network (ANN) that can find solutions to problems that cannot be solved linearly [46–49]. The primary restriction on the simple perceptron is this issue. MLP, however, can be locally or fully networked. In order for a layer to be fully linked, every neuron in that layer must be connected to every neuron in the next layer. A locally connected MPL does not meet this requirement.

An MLP's layers can be divided into three categories (see Fig. 2) [46–49]:

- Input layer: there is no process here; only the independent variables' information arrives through this layer.
- Output layer: here, the link to the dependent variables is established.

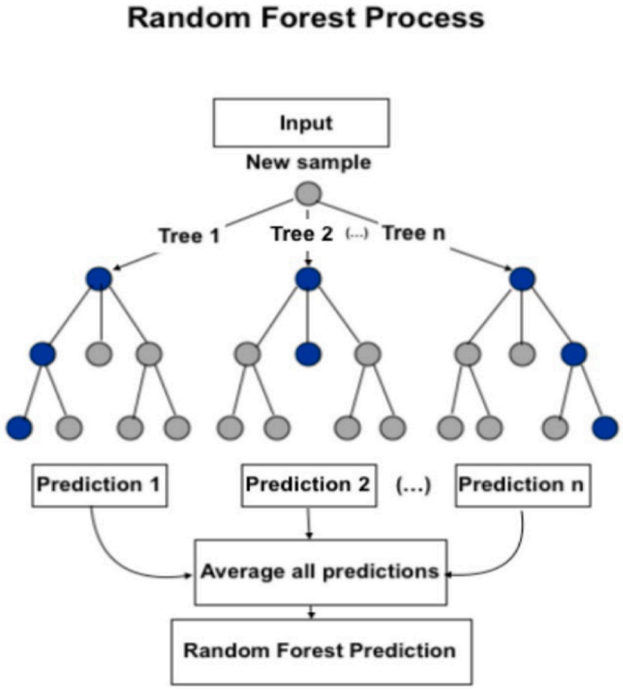

Fig. 1. An illustration of a Random Forest Regression process.

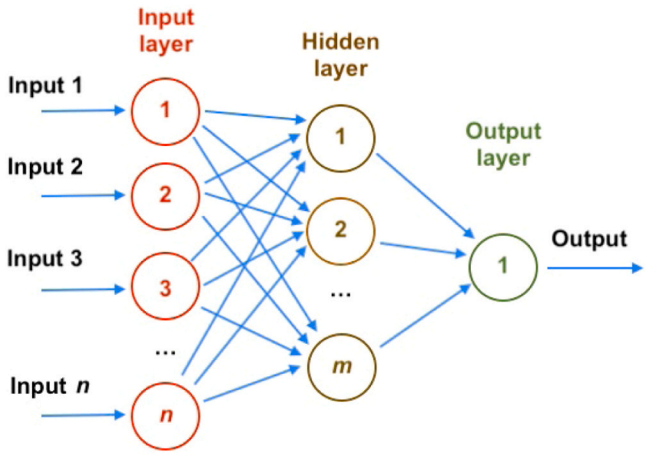

Fig. 2. An illustration of an MLP-inspired artificial neural network (in this case, picture shows n neurons forming part of the hidden layer, m neurons in the input layer and a single neuron in the output layer).

- Hidden layers: these are strata that transfer and process information from the input to the output layers and are positioned in between those layers.

The mathematical principle used to train these kinds of neural networks is *backpropagation*, commonly referred to as error backpropagation or the generalized delta rule [46–49]. In this context, a MLP is also referred to as a BP-ANN (Backpropagation Artificial Neural Network). The primary quality of these ANN is also the requirement of *derivable* transfer functions for the processing units (neurons).

The multilayer perceptron (MLP) uses this type of learning by adjusting the connection weights in light of the discrepancy between the expected and actual output values. For data point n the error at node j is $e_j(n) = d_j(n) - y_j(n)$, being d the observed value and y the value predicted by the multilayer perceptron. The total error to correct is [46–49]:

$$\varepsilon(n) = \frac{1}{2}\sum_{j} e_j^2(n) \tag{3}$$

Using the *gradient descent approach*, we discover that the following factors determine how the weights change [46–49]:

$$\Delta w_{ji}(n) = -\eta\frac{\partial\varepsilon(n)}{\partial v_j(n)}y_i(n) \tag{4}$$

where:

- η is the *learning rate*. It has to be carefully selected because a little value may cause very slow convergence and a large value may prevent the optimization from convergent. A range of acceptable values is from 0.1 to 0.8.
- y_i is the result of the neuron's work in the previous layer.
- v_j is the induced field that is localized. It is demonstrable that for a specific output node:

$$-\frac{\partial\varepsilon(n)}{\partial v_j(n)} = e_j(n)\cdot\phi'\left(v_j(n)\right) \tag{5}$$

being ϕ' the derivative of the activation function.

2.4. M5 model tree

The following inspired idea was used to create this approximation, which also relies on machine learning [50–52]. The parameter space can be divided into several subspaces and, in each of them, a linear

regression technique is built. Since the linear fits specialize in particular subsets of the input space, the resultant approximation would be regarded as a modular technique.

To force a model tree, the mathematical method known as algorithm M5 is used [50–52]. In fact, a set of T training data is taken into account here. Each instance is represented by the values of a set of related target output values and a set of not-variable input attribute values. The main purpose is to develop a method that links the input attribute values of the training data with an objective value of those data. If the model properly predicts the objective values of the unknown cases, its excellence can be assessed.

Divide-and-conquer is the approach used to construct tree-based machine learning models [63–66]. Many tests are chosen to partition the set T into smaller sets, or the set T is connected to a leaf. Recursive application of this splitting algorithm is used. The M5 model tree technique's division criterion uses the standard deviation of the class values that reach a node to measure the error at that node and then computes the projected decrease of this error to verify each attribute in that node. Definitely, the reduction of the standard deviation (SDR) can be ascertained by employing the following mathematical representation [50–52, 63–66]:

$$SDR = sd(T) - \sum \frac{|T_i|}{|T|} sd(T_i) \tag{6}$$

where T is the quantity of examples reaching the node, T_i denotes the subset of cases that have an impact on the ith possible collection outcome, and sd denotes the standard deviation [50–52,63–66].

The M5 model tree chooses the element that completely optimizes the anticipated error lowering after carefully examining all potential divisions [63–66]. When the class values of all examples reaching a node differ by just a very little tolerance (the stopping requirement), or else when only a small number of instances are left, the M5 model tree splitting mechanism comes to an end. An illustration of a simple M5 model tree can be seen in Fig. 3.

2.5. Multivariate linear regression (MLR)

A mathematical model known as *multivariate linear regression* (MLR) is used to roughly represent the relationship of dependence between a dependent variable Y, m independent variables $X_i$with $m \in Z^+$and a random termε (stochastic error) [53,54]. The hyperplane of the subsequent parameters β_i (termed the coefficients of the multiple regression model) can be used to define this MLR model [53,54,67–69]:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + . + \beta_m X_m + \varepsilon = \beta_0 + \sum_{j=1}^{m} \beta_j X_j + \varepsilon \tag{7}$$

where:

- Y is the dependent variable or response variable;
- $X_1, X_2, ., X_m$ are the m explanatory, independent, or regressor variables;
- $\beta_0, \beta_1, \beta_2, ., \beta_m$are the parameters of the MLR model and measure the influence that the explanatory variables have on the regressor. The term$\beta_0$is the *intercept* (constant term), the β_i $(i \geq 1)$ are the corresponding parameters for each independent variable, and m is the number of independent parameters to take into account in the regression.

The regression problem consists of choosing certain values for the unknown parameters β_j, so that the equation is completely specified. This requires a set of observations or a sample from this model. In any i-th observation (with $i = 1, 2, ., m$), the simultaneous behavior of the dependent variable and the explicit variables is recorded (random disturbances are assumed to be unobservable). Suppose that we have a sample of size n given by $\{(x_{ij}, y_i)\}$ with $j = 1, 2, ., m$ where x_{ij} denotes the i-th observed value in the regressor X_j and y_i denotes the i-th observation of Y, then the model takes the form [32,33,50–52]:

$$\hat{y}_i = \beta_0 + \sum_{j=1}^{m} \beta_j x_{ij} + \varepsilon_i \tag{8}$$

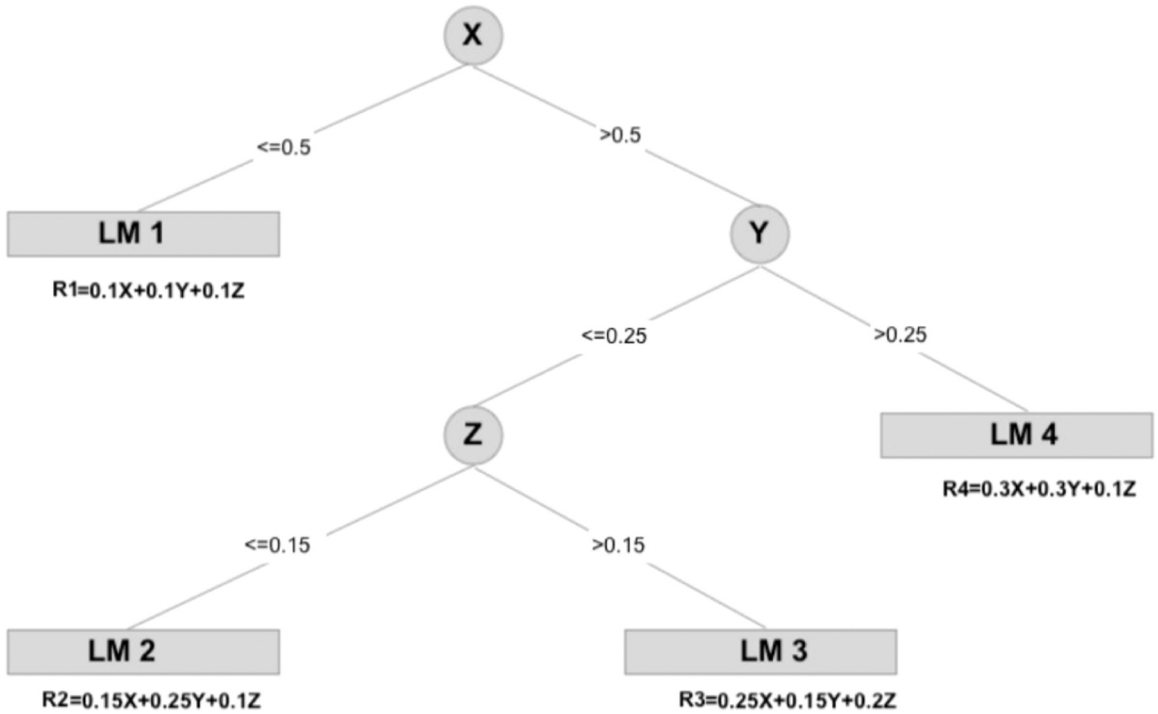

Fig. 3. An illustration of a M5 model tree model.

where $\widehat{y}_i$ is the value of Y predicted by the MLR model, $\varepsilon_i = y_i - \widehat{y}_i$ is the error associated with the i-th measurement of the value X_j and follows the usual assumptions so that $\varepsilon_i \sim N(0, \sigma^2)$ (zero mean, constant variance and equal to σ^2, and $Cov(\varepsilon_i, \varepsilon_j) = 0 \quad \text{if } i \neq j$). To assess the model parameters, the method of least squares can be used, in this case, the squared error function is given by [53,54,67–69]:

$$S(\beta_0, \beta_1, ., \beta_m) = \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{m} \beta_j x_{ij} \right)^2 \tag{9}$$

which we want to minimize. The least squares estimators denoted by β_0, $\beta_1, \beta_2, ., \beta_m$must satisfy [53,54,67–69]:

$$\left. \frac{\partial S}{\partial \beta_j} \right| = 0 \,, \quad \forall j = 0, 1, 2, ., m \tag{10}$$

This system with $m+1$ equations can be written in matrix form as [53,54,67–69]:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{11}$$

where$\mathbf{Y} \in \Re^{n\times 1}, \mathbf{X} \in \Re^{n\times(m+1)}, \boldsymbol{\beta} \in \Re^{(m+1)\times 1}$ and $\boldsymbol{\varepsilon} \in \Re^{n\times 1}$. In matrix form, the squared error function S can be written as [67–69]:

$$S(\boldsymbol{\beta}) = \sum_{i=1}^{n} \varepsilon_i^2 = \boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon} = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) \tag{12}$$

and Eq. (9) is reduced to the *normal equations* [67–69]:

$$\mathbf{X}^T \mathbf{X} \widehat{\boldsymbol{\beta}} = \mathbf{X}^T \mathbf{Y} \tag{13}$$

Then, the least squares estimator is given by [53,54,67–69]:

$$\widehat{\boldsymbol{\beta}} = \left(\mathbf{X}^T \mathbf{X} \right)^{-1} \mathbf{X}^T \mathbf{Y} \tag{14}$$

So the final fitted multivariate linear regression model is given by [53,54,67–69]:

$$\widehat{\mathbf{y}} = \mathbf{X}^T \widehat{\boldsymbol{\beta}} = \widehat{\beta}_0 + \sum_{j=1}^{m} \widehat{\beta}_j x_j \tag{15}$$

An illustration of the multivariate linear regression model is shown in Fig. 4.

2.6. Approach accuracy

Eighty of the input variables from Subsection 2.1 were used in this investigation to construct the unique GS/RFR–relied technique. The superconducting critical temperature T_c is the response variable that needs to be foretold, as is common knowledge. It is crucial to pick the model that best matches the experimental data in order to accurately forecast T_c from 80 factors. The coefficient of determinationR^2 [70–73]

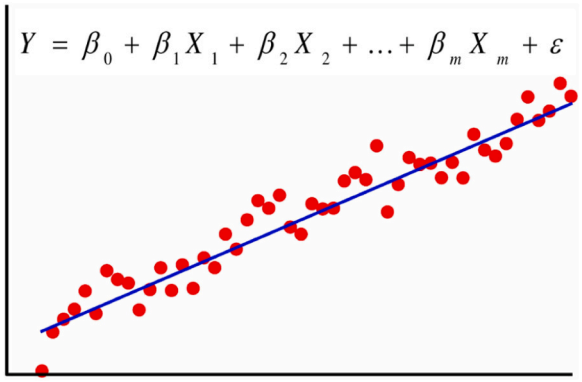

Fig. 4. An illustration of a Linear Regression model.

was the rule used in this study (even though there are many possible statistics that can be employed to determine the goodness-of-fit) because it is a statistic used in the context of a statistical model whose primary goal is to foretell future outcomes or to verify a supposition. The following sums of squares are defined by referring to the observed values as t_i and the values foretold by the model y_i as [70–73]:

$SS_{tot} = \sum_{i=1}^{n} (t_i - \bar{t})^2$: is the overall sum of squares, scaled to the variance of the sample.

$SS_{reg} = \sum_{i=1}^{n} (y_i - \bar{t})^2$: is also known as the explained sum of squares and is the regression sum of squares.

$SS_{err} = \sum_{i=1}^{n} (t_i - y_i)^2$: is the squared residual sum.

$\bar{t}$being the average of the n observed data:

$$\bar{t} = \frac{1}{n} \sum_{i=1}^{n} t_i \tag{16}$$

The following equation specifies the coefficient of determination relied on the earlier sums [70–73]:

$$R^2 \equiv 1 - \frac{SS_{err}}{SS_{tot}} \tag{17}$$

The mean absolute error (MAE) and root mean square error (RMSE) were suplementary criteria taken into account in this investigation [70–73]. The predictive power of a mathematical model is typically assessed using the RMSE statistic. The following equations provide the expression of the RMSE [70–73]:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (t_i - y_i)^2}{n}} \tag{18}$$

In the event that the root mean square error (RMSE) is zero, the expected and observed data are the same. The MAE, on the other hand, does not take into account the direction of the errors when calculating their average size in a collection of forecasts. The MAE is the average of the absolute values of the discrepancies between a forecast and the related observation over the verification sample. Its mathematical formulation is as follows [70–73]:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |t_i - y_i|}{n} \tag{19}$$

Moreover, the RFR approach largely depends on the following two hyperparameters: [36–40]:

Number of regression trees (ntree): amount of trees to be grown. Model construction will cost more to compute the larger the tree. The 500 trees setting is the default value.

Number of input variables per node (mtry): it deals with the number of variables we should choose during a node split. One-third of the full set of input variables, p, is taken as the default value. To prevent overfitting, we must always make an effort to avoid using small values of mtry.

It is crucial to keep in mind that the RFR method heavily depends on finding the two aforementioned optimal hyperparameters. Based on its effectiveness in resolving similar optimization issues, the optimizer known as Grid Search (GS) [41–45] has been used in this work to find these parameters.

Hence, the superconducting critical temperature T_c (output variable) has therefore been successfully predicted using a novel hybrid GS/RFR–based method by evaluating the influence of 80 variables (input variables) and successfully optimizing the computation through the examination of the coefficient of determination R^2. The flowchart for the RFR–relied model created in this study is shown in Fig. 5.

The most common method for calculating the actual coefficient of determination (R^2) is cross-validation [70–75]. In fact, a detailed 10–fold cross-validation method was employed to guarantee the

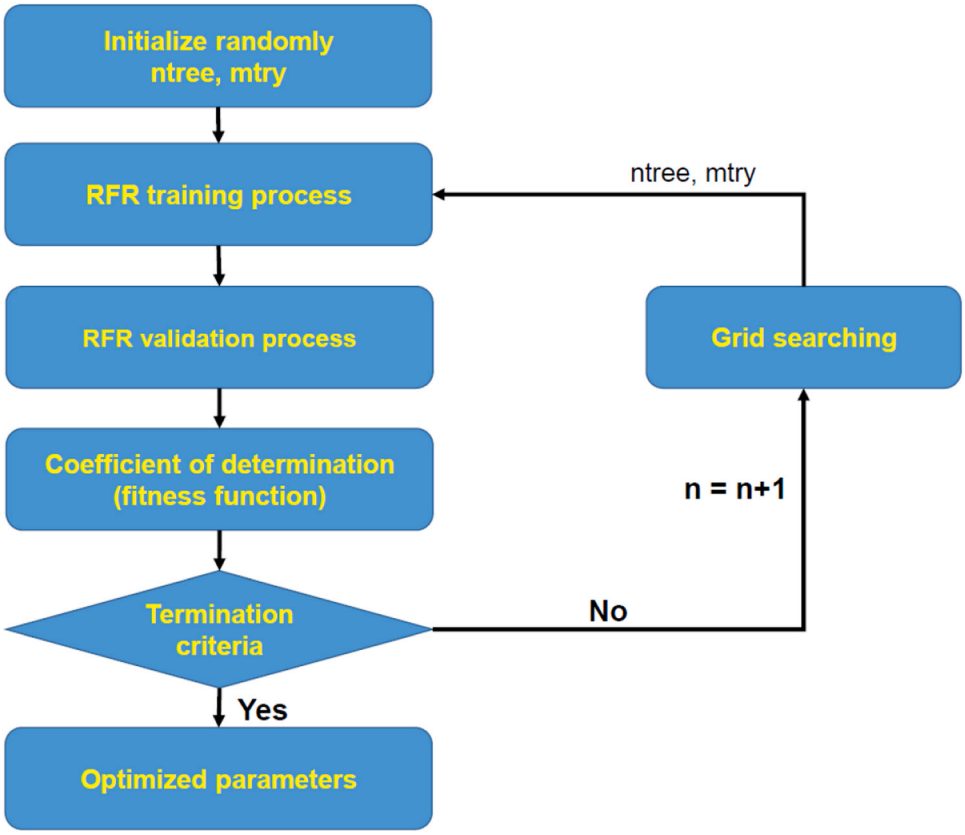

Fig. 5. Flowchart corresponding to GS/RFR model's parameter optimization.

RFR–relied model's predictive ability [74–77], which required dividing the sample into ten portions, utilizing nine for training and the final portion for testing. For testing and computing the average error, this process was carried out ten times utilizing each of the parties from the ten divisions. So, every possible variable within the RFR–relied model parameters has been assessed with the purpose of identifying the optimal point by first looking for those parameters that reduce the average error.

WEKA, an open-source machine learning program used in this study, was used to create the DE/RFR–relied model [78,79]. Additionally, the MLP, M5 model tree, and MLR models were implemented by also using the data-driven software WEKA [78,79].

In order to get the best ntree and mtry values for the RFR parameters, the cross-validation error for each iteration is compared using the GS optimizer. Using grid search (GS), the most precise values for ntree and mtry were 500 and 23, respectively (see Fig. 6).

3. Analysis of results and discussion

Tables 1 and 2 above list each of the 80 independent input variables

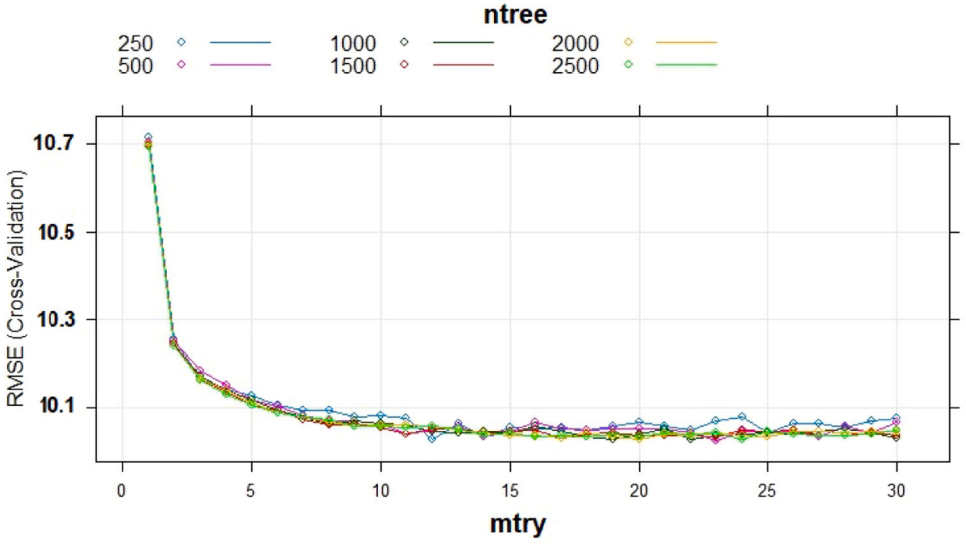

Fig. 6. Tuning of Random Forest parameters using the GS optimizer.

(80 physico-chemical variables). The present study used 21,263 samples in total, which means that data from 21,263 experimental samplings were constructed and processed. The whole dataset was roughly divided into two equal halves, the training set being one, and the testing set being the other. Given that the training set still had a substantial amount of samples, 1000 samples were taken out at random, and ten-fold cross-validation was used to tune the hyperparameters. The entire training dataset was used to build a model after the optimal parameters had been identified, and the testing dataset was used to validate the model.

3.1. Metrics evaluation

Using the test dataset and the subsequent computations, the RFR–relied approximation permitted the creation of a model with high perks to evaluate the critical temperature T_c. To anticipate the superconducting critical temperature of the superconductor state for various materials, the MLP, M5 model tree, and MLR approaches were also constructed for the T_c output factor. We employ a variety of metrics such as the R^2 score, Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) to assess the performance of machine learning models.

Table 3 shows the coefficient of determination and the coefficient of correlation (R^2 and r), as well as root mean square error (RMSE) and mean absolute error (MAE) over the test set for the RFR, MLP, M5 model tree, and MLR approaches for the response T_c variable.

3.2. Significance of variables

Assessing the importance of variables is a valuable step in reducing the number of variables in a regression model. This process also helps to improve model interpretability. Reducing the number of input variables, a process also known as feature selection or dimensionality reduction, can improve the performance of a regression model in several ways:

- Improved Model Interpretability: A simpler model with fewer variables is easier to understand and explain, making it more accessible for stakeholders, including non-technical audiences.
- Mitigating Overfitting: A model with too many variables is prone to overfitting, where it fits noise in the data rather than the underlying patterns. Reducing variables helps prevent overfitting and improves generalization to new data.
- Enhanced Model Robustness: Fewer variables can make your model more robust to outliers, noise, or small variations in the data, leading to more reliable predictions.
- Faster Model Training and Inference: With fewer variables, both model training and prediction become faster, which is crucial for real-time or resource-constrained applications.
- Avoiding Multicollinearity: Removing highly correlated variables reduces multicollinearity issues, where two or more variables in the model are highly correlated, making it challenging to attribute the effect of each variable separately.

Feature selection helps identify and emphasize the most important variables, providing insights into which factors have the most significant impact on the target variable. Different techniques, such as feature importance analysis, correlation analysis, and regularization, can aid in the process of selecting the most relevant variables. Methods like LIME (Local Interpretable Model-agnostic Explanations), permutation, and SHAP (SHapley Additive exPlanations) allow gain insights into variable significance and model interpretability [80].

In this study, we use the permutation method to evaluate how sensitive the model performance is to changes in the values of individual features. The permutation method for determining variable importance is model-agnostic, making it a valuable tool for understanding the relative importance of features in various types of machine learning models. It provides a clear quantitative measure of feature importance based on how much the model relies on each feature for its predictions, since if permuting a predictor variable leads to a significant drop in prediction accuracy, it indicates that this variable is a significant variable in your model. That is, the permutation accuracy measure helps you determine whether a predictor variable is genuinely associated with the response variable or if any apparent association is merely due to random chance. The importance ranking of the independent input factors in predicting the superconducting critical temperature T_c for this complex nonlinear complex issue, using the permutation method, is another significant finding of the current work (see Table 4 and Fig. 7).

Ultimately, Weighted Standard Deviation Thermal Conductivity is the most important input variable in the T_c forecasting process, according to the GS/RFR approach. Range Thermal Conductivity, the second-most important input factor, is followed by: Range Atomic Radius, Standard Thermal Conductivity, Weighted Entropy Atomic Mass, Weighted Mean Valence, Weighted Geometric Mean Valence, Weighted Entropy Atomic Radius, Range First Ionization Energy and Entropy Valence.

The first-order and second-order terms that make up the RFR–relied technique for the superconducting critical temperature T_c are indicated in a pictorial graph in Fig. 8.

The most important attributes, according to our investigation, had to do with thermal conductivity. This is expected given that transitions involving lattice phonons and electrons drive both superconductivity and thermal conductivity [8]. Ionic characteristics may also have an impact on superconductors' ability to generate ions, which is associated with movement across the crystalline lattice (pertaining to electron affinity and the first ionization energy). With regard to superconductivity, the BCS theory fits in well with this interpretation [2]. By comprehending the physico-chemical characteristics that are more closely connected to the critical temperature, the analysis of superconducting materials can be made simpler.

Overall, the RFR–relied technique has shown to be a very accurate and highly effective way for calculating the superconducting critical temperature T_c (dependent variable) as a function of various important measured physico-chemical factors, commensurate with the actual data reported in this study. Specifically, Figs. 9 to 13 indicate the comparison between the experimental and predicted T_c values employing the GS/RFR, MLP, M5 model tree, and MLR models for the test dataset.

Table 3
Coefficients of determination (R^2), correlation coefficients (r), root mean square deviation (RMSE) and mean absolute error (MAE) over the testing dataset, for the four different methods adjusted (RFR, M5 model tree, MLP and MLR) in this investigation to the training dataset.

Method	R^2	r	RMSE	MAE
Random Forest	0.9146	0.9565	9.905	5.739
M5	0.8661	0.9309	12.407	7.263
MLP	0.8277	0.9114	14.073	9.319
MLR	0.5211	0.7220	23.463	18.191

Table 4
Relative significance of the physico-chemical input variables in the best-fit RFR–relied model for the prediction of the superconducting critical temperature Tc prediction.

Variable	Weight
wtd_std_ThermalConductivity	0.7207
range_ThermalConductivity	0.6889
range_atomic_radius	0.6557
std_ThermalConductivity	0.6541
wtd_entropy_atomic_mass	0.6299
wtd_mean_Valence	-0.6288
wtd_gmean_Valence	-0.6125
wtd_entropy_atomic_radius	0.6070
range_fie	0.6050
entropy_Valence	0.6006

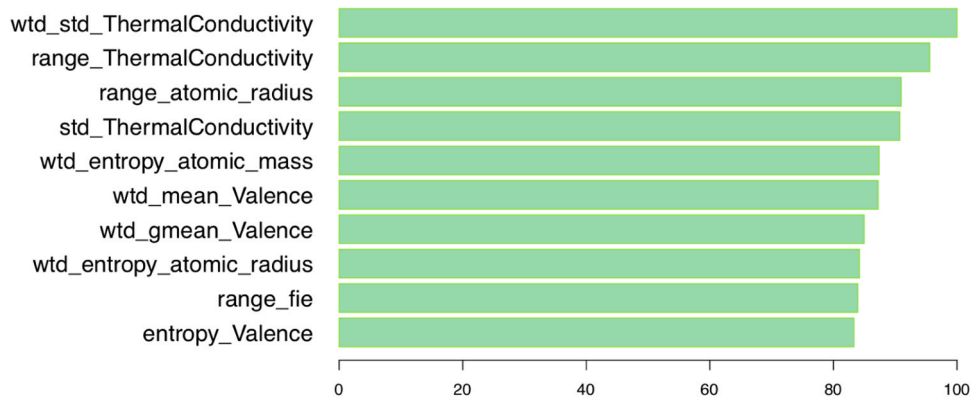

Fig. 7. Physico-chemical input variables' relative significance in foretelling the critical temperature T_c for the adjusted RFR–relied model.

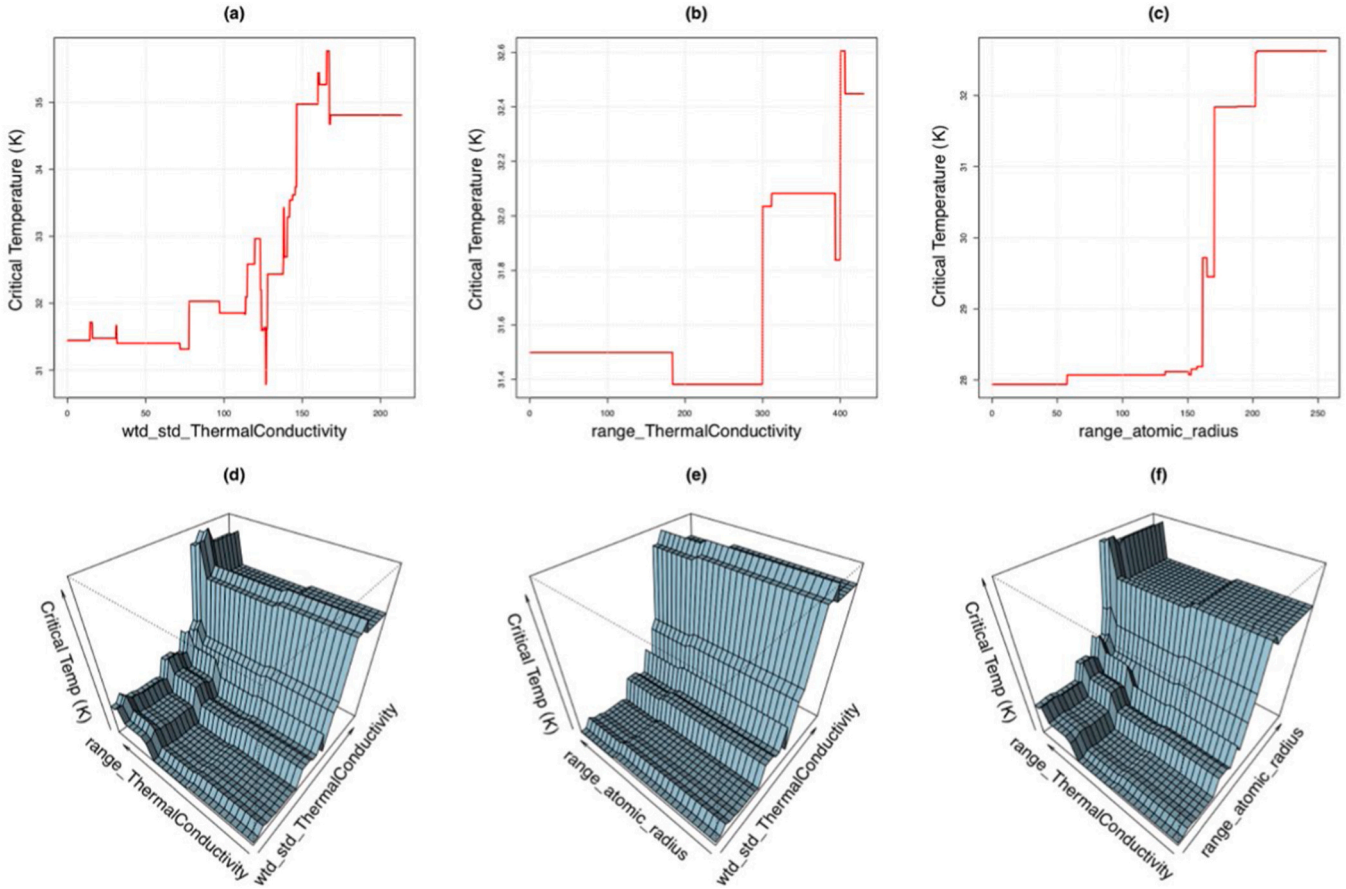

Fig. 8. Representation of the dependent superconducting critical temperature T_c variable's first-order and second-order terms for the three most significant independent input variables: (a) T_c vs. Weighted Standard Thermal Conductivity; (b) T_c vs. Range Thermal Conductivity; (c) T_c vs. Range atomic radius; (d) T_c vs. Range Thermal Conductivity and Weighted Standard Thermal Conductivity; (e) T_c vs. Range atomic radius and Weighted Standard Thermal Conductivity; and (f) T_c vs. Range atomic radius and Range Thermal Conductivity.

Consequently, to solve this nonlinear regression problem, it is fundamental to bring together the RFR procedure with the GS optimizer to produce an original hybrid strategy that is significantly more reliable and efficient than the other three regression methods. Particularly, a strong correlation between the modeled and measured T_c values was discovered. The T_c watched and foretold for the first materials in Fig. 6 are shown in Table 5.

3.3. Discussion

Relied on the aforementioned discoveries, the following key findings of this inquiry can be made:

Present analytical models that attempt to foretell the superconducting critical temperature T_c from reported data fall short of being precise enough because they oversimplify a difficult, highly nonlinear problem. Consequently, the best method for producing precise estimates of the T_c from experimental samplings is to use ML techniques, such as the hybrid GS/RFR–relied approximation used in this investigation.

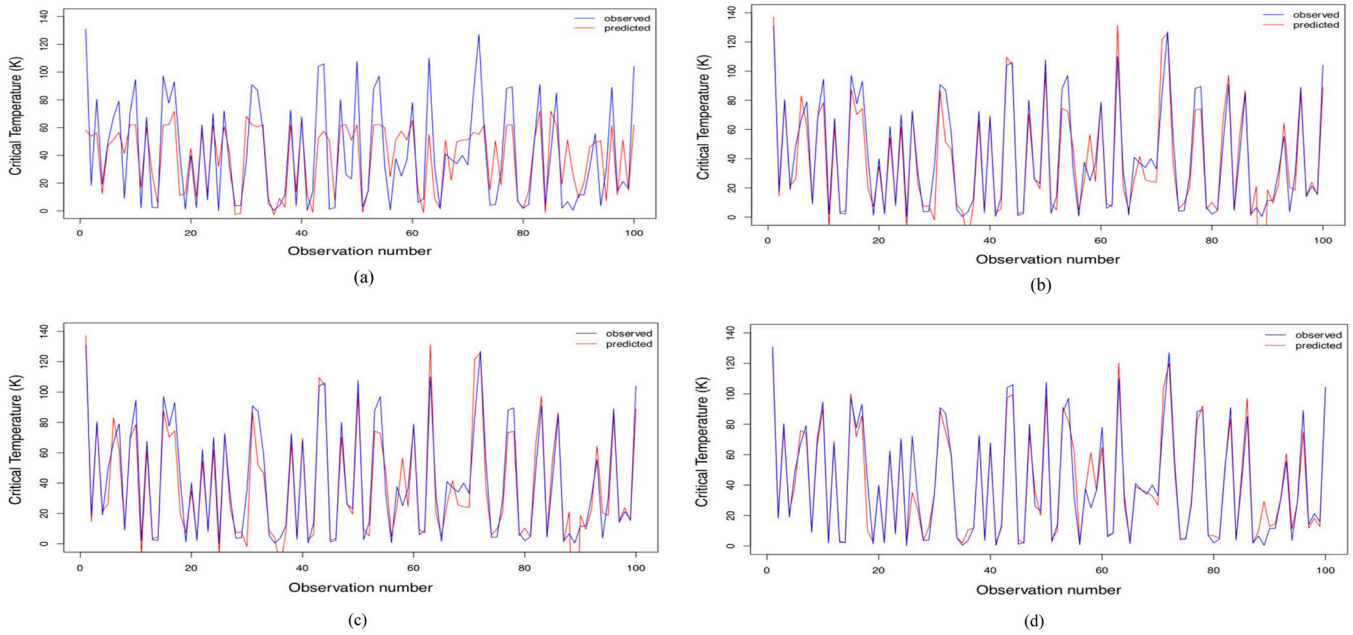

Fig. 9. Observed vs. foretold superconducting critical temperature T_c values using 100 samples from the testing dataset for four distinct models: (a) MLR model ($R^2 = 0.5211$and$r = 0.7220$); (b) MLP regression model ($R^2 = 0.8277$and$r = 0.9114$); (c) M5 model tree ($R^2 = 0.8661$and$r = 0.9309$); and (d) RFR–relied model ($R^2 = 0.9146$and$r = 0.9565$).

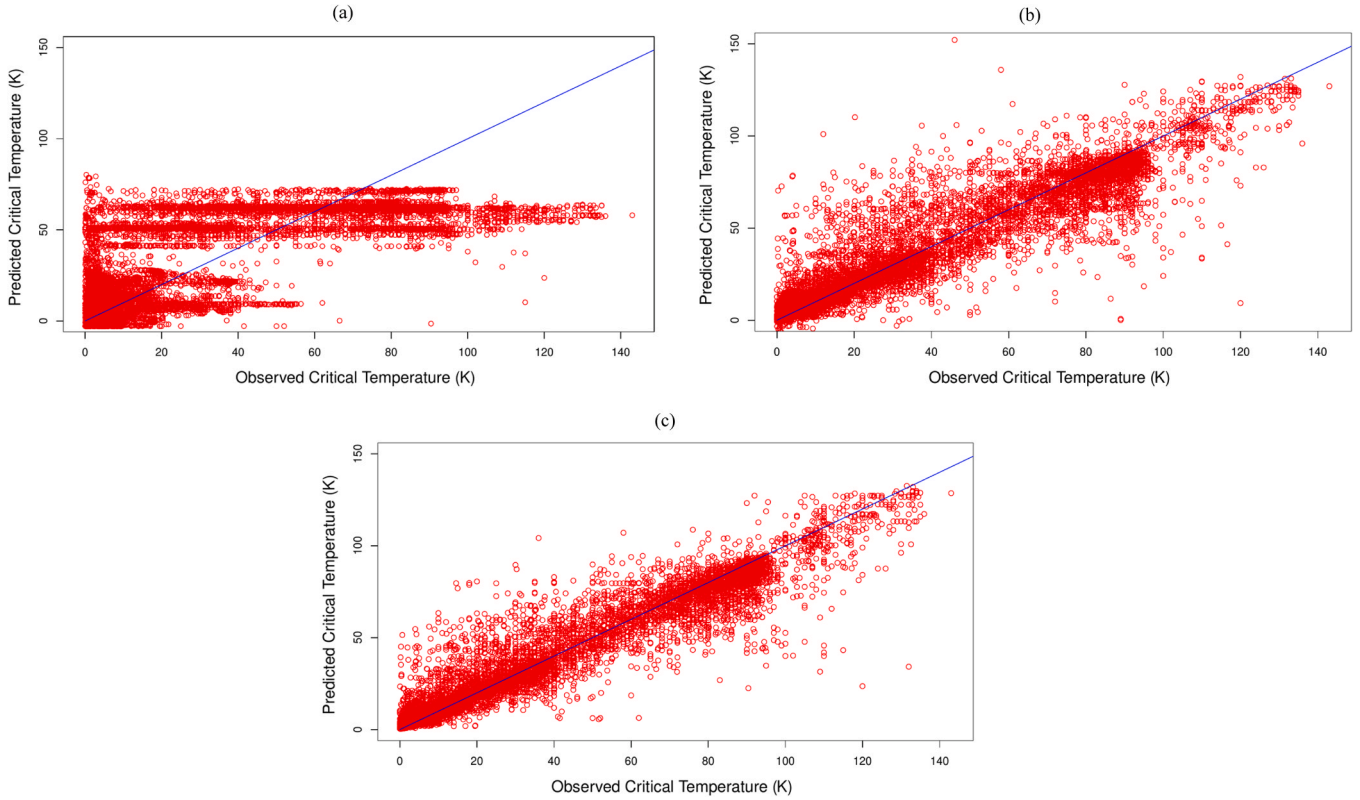

Fig. 10. Observed vs. foretold superconducting critical temperature T_c scatterplots from the testing dataset for four distinct models: (a) MLR model (b) MLP regression model (c) M5 model tree and (d) RFR–relied model.

Here, the hypothesis that a hybrid GS/RFR–relied approach can successfully validate the identification of T_c in a wide range of superconductors has been proven correct.

This RFR–relied methodology produced satisfactory coefficients of determination and correlation coefficients with valuations of 0.9146 and 0.9565, each in order, when applied to the entire experimental dataset from the T_c.

Finally, the ranking of the input variables' importance for estimating the T_c from experimental samples in various superconductors has also been determined. Particularly, it has been determined that Weighted

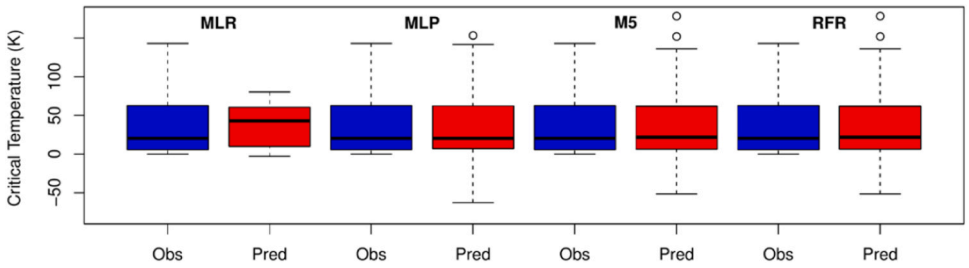

Fig. 11. Observed and foretold superconducting critical temperature T_c boxplots from the testing dataset for the four models.

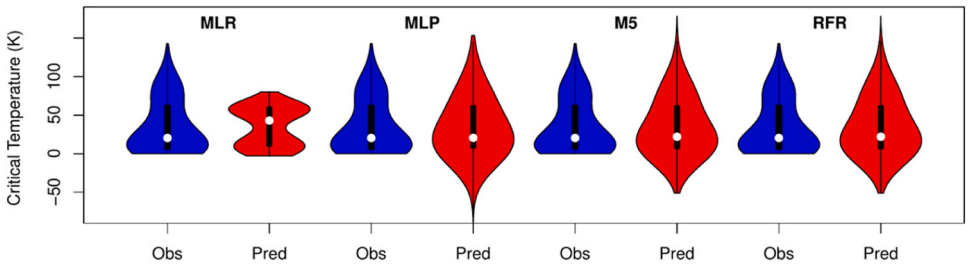

Fig. 12. Observed and foretold superconducting critical temperature T_c violinplots from the testing dataset for the four models.

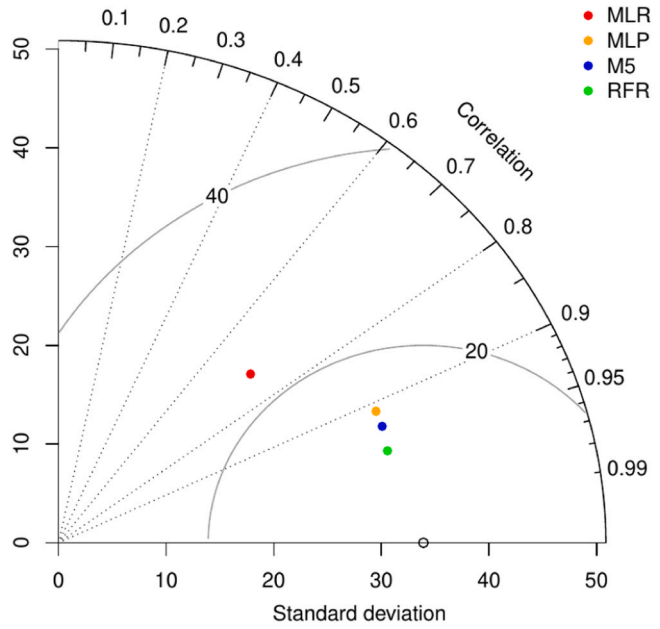

Fig. 13. Taylor diagram for the four constructed models for the superconducting critical temperature T_c using the testing dataset.

Standard Thermal Conductivity is the single most crucial variable in forecasting critical temperature T_c. Noting the following sequential order of importance is also crucial: the Range Thermal Conductivity, Range Atomic Radius, Standard Thermal Conductivity, Weighted Entropy Atomic Mass, Weighted Geometric Mean Valence, Weighted Mean Valence, Range First Ionization Energy, Weighted Entropy Atomic Radius and Entropy Valence in the obtained T_c outcome.

4. Conclusions

The superconducting critical temperature T_c of a diversity of superconductors can be accurately foretold using hybrid GS/RFR–relied approximation, using features derived from the unique physicochemical of each superconductor and/or experiment. As a result, the GS/RFR–relied approach proved to be a very reliable and workable solution to the nonlinear issue of T_c estimate from experimental samplings in various superconductors.

The model may help researchers focus their hunt for high temperature superconductors. For instance, it is possible to use the described method on a bigger database as a future development of this study [59]. In the future, researchers might use this dataset in conjunction with brand-new data (like pressure or crystal structure) to build models that are more precise.

Table 5
Tc observed and foretold for various of the first materials (or chemical compounds) in Fig. 6 for RFR model.

Material	Observed T_c (K)	Predicted T_c (K)
$Hg_1Ba_2Ca_2Cu_3O_8$	130.90	126.61
$Ca_{1.88}Na_{0.12}Cu_1Cl_2O_2$	18.40	18.65
$Pb_2Sr_2Ho_{0.5}Ca_{0.5}Cu_{2.94}Pt_{0.06}O_8$	80.40	76.75
$Na_{0.29}Hf_1N_1Cl_1$	19.00	22.09
$Eu_{1.5}Ce_{0.5}Ru_1Sr_2Cu_2O_{10}$	49.00	40.03
$Bi_2Sr_{1.5}Ca_{1.5}Cu_2O_{8.25}$	67.00	75.87
$Pb_2Sr_2Y_{0.7}Ca_{0.3}Cu_3O_8$	79.00	73.84
$Bi_2Sr_2Cu_1O_6$	9.00	9.19
$Lu_{0.7}Pr_{0.3}Ba_2Cu_3O_7$	70.00	66.17
$Eu_1Ba_2Cu_3O_7$	94.50	90.11
$Hf_{0.8}Rh_{0.2}$	1.95	2.85
$Y_{0.75}Ca_{0.25}Ba_2Cu_{2.75}Co_{0.25}O_{6.82}$	67.50	68.82
$Au_{0.07}Zr_{0.93}$	2.52	3.37
$Ge_{1.85}La_1$	2.17	1.99
$Bi_{1.6}Pb_{0.4}Sr_2Ca_{1.999}Sm_{0.001}Cu_3O$	97.10	100.02
$Y_{0.7}Ca_{0.15}Pr_{0.15}Ba_2Cu_3O_{6.95}$	77.60	71.69
$Y_1Ba_{1.75}Nd_{0.25}Cu_3O$	93.00	85.45
$Yb_{0.8}Fe_2Se_2$	42.00	9.83
In_3Te_4	1.25	2.71
$Tl_{0.9}Mo_{0.1}Sr_{1.2}La_{0.8}Cu_1O_5$	40.00	39.24

Funding

Universidad de León and Cátedra RENNOVA.

CRediT authorship contribution statement

Paulino José García–Nieto: Conceptualization, Data curation,

Formal analysis, Methodology, Investigation, Writing – original draft. **Esperanza García–Gonzalo:** Conceptualization, Data curation, Formal analysis, Methodology, Investigation, Writing – review & editing. **Luis Alfonso Menéndez García:** Conceptualization, Data curation, Formal analysis, Methodology, Investigation. **Laura Álvarez–de–Prado:** Methodology, Formal analysis, Writing – review & editing, Visualization. **Antonio Bernardo–Sánchez:** Methodology, Formal analysis, Writing – review & editing, Visualization.

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.

Acknowledgements

The University of Oviedo's Department of Mathematics generously provided computational assistance, which the authors gratefully acknowledge. Likewise, the authors would like to thank Anthony Ashworth for revising this research paper in English.

References

[1] N.W. Ashcroft, N.D. Mermin, Solid State Physics, Thomson Press Ltd,, Delhi, India, 2003.
[2] M. Tinkham, Introduction to Superconductivity, Dover Publications,, New York, 2004.
[3] C. Kittel, Introduction to Solid State Physics, John Wiley & Sons,, New York, 2005.
[4] J.F. Annett, Superconductivity, Superfluids, and Condensates, Oxford University Press,, Oxford, UK, 2004.
[5] C.P. Poole Jr., R. Prozorov, H.A. Farach, R.J. Creswick, Superconductivity, Elsevier,, Amsterdam, 2014.
[6] A.A. Abrikosov, Fundamentals of the Theory of Metals, Dover Publications,, New York, 2017.
[7] K. Hamidieh, A data-driven statistical model for predicting the critical temperature of a superconductor, Comput. Mat. Sci. 154 (2018) 346–354, https://doi.org/10.1016/j.commatsci.2018.07.052.
[8] R.P. Huebener, Conductors, Semiconductors, Superconductors: An Introduction to Solid-state Physics, Springer,, Berlin, 2019.
[9] B.T. Matthias, Empirical relation between superconductivity and the number of electrons per atom, Phys. Rev. 97 (1955) 74–76, https://doi.org/10.1103/PhysRev.97.74.
[10] R. Tibshirani, Regression shrinkage and selection via the Lasso, J. R. Stat. Soc. Ser. B Methodol. 58 (1) (1996) 267–288, https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.
[11] A.E. Hoerl, R.W. Kennard, Ridge regression: biased estimation for nonorthogonal problems, Technometrics 12 (1) (1970) 55–67, https://doi.org/10.2307/1267351.
[12] V. Vapnik, The Nature of Statistical Learning Theory, Springer Science & Business Media,, Berlin, 1999.
[13] A. Majid, A. Khan, G. Javed, A.M. Mirza, Lattice constant prediction of cubic and monoclinic perovskites using neural networks and support vector regression, Comput. Mater. Sci. 50 (2) (2010) 363–372, https://doi.org/10.1016/j.commatsci.2006.08.015.
[14] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297, https://doi.org/10.1007/BF00994018.
[15] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32, https://doi.org/10.1023/A:1010933404324.
[16] J.R. Quinlan, Simplifying decision trees, Int. J. Man-Mach. Stud. 27 (3) (1987) 221–234, https://doi.org/10.1016/S0020-7373(87)80053-6.
[17] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. R. Stat. Soc. Ser. B, Methodol. 67 (2) (2005) 301–320, https://doi.org/10.1111/j.1467-9868.2005.00503.x.
[18] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou, M. Li, J. Xie, M. Lin, Y. Geng, Y. Li, Xgboost: extreme gradient boosting, R. Package Version vol. 1 (4) (2015) 1–4.
[19] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory. MHS'95, Proceedings of the Sixth International Symposium on Micro Machine and Human Science, IEEE Publisher,, Nagoya, Japan, 1995, pp. 39–43, https://doi.org/10.1109/MHS.1995.494215.
[20] X.–G. Shao, H.–Z. Yang, G. Chen, Parameters selection and application of support vector machines based on particle swarm optimization algorithm, IET Control. Theory Appl. 23 (5) (2006) 740–743.
[21] H.–R. Zhang, Y. Zhang, D.–B. Dai, M. Cao, W.–F. Shen, Modelling and optimization of the superconducting transition temperature, Mater. Des. 92 (2016) 371–377, https://doi.org/10.1016/j.matdes.2015.12.081.
[22] Z. Pawlak, Rough sets, Int. J. Comput. Inf. Sci. 11 (5) (1982) 341–356, https://doi.org/10.1007/BF01001956.
[23] R. Hecht-Nielsen III, 3-Theory of the Backpropagation Neural Network. Neural Networks for Perception, Academic Press,, New York, 1992, pp. 65–93, https://doi.org/10.1016/B978-0-12-741252-8.50010-8.
[24] Y. Liu, H. Zhang, Y. Xu, S. Li, D. Dai, C. Li, G. Ding, W. Shen, Q. Qian, Prediction of superconducting transition temperature using a machine-learning method, Mater. Tehnol. 52 (5) (2018) 639–643, https://doi.org/10.17222/MIT.2018.043.
[25] V. Stanev, C. Oses, A.G. Kusne, E. Rodriguez, J. Paglione, S. Curtarolo, I. Takeuchi, Machine learning modeling of superconducting critical temperature, NPJ Comput. Mater. 4 (1) (2018) 1–14, https://doi.org/10.1038/s41524-018-0085-8.
[26] L. Ward, A. Agrawal, A. Choudhary, C. Wolverton, A general-purpose machine learning framework for predicting properties of inorganic materials, NPJ Comput. Mater. 2 (1) (2016) 1–7, https://doi.org/10.1038/npjcompumats.2016.28.
[27] K. Matsumoto, T. Horide, An acceleration search method of higher T_c superconductors by a machine learning algorithm, Appl. Phys. Express 12 (7) (2019), 073003, https://doi.org/10.7567/1882-0786/ab2922.
[28] B. Roter, S. Dordevic, Predicting new superconductors and their critical temperatures using machine learning, Phys. C. Supercond. 575 (2020), 1353689, https://doi.org/10.1016/j.physc.2020.1353689.
[29] M. Gaikwad, A.R. Doke, Featureless approach for predicting critical temperature of superconductors, in: 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE Publisher, Kharagpur, India, 2020, pp. 1–5, https://doi.org/10.1109/ICCCNT49239.2020.9225447.
[30] P.J. García-Nieto, E. García-Gonzalo, J.P. Paredes-Sánchez, Prediction of the critical temperature of a superconductor by using the WOA/MARS, Ridge, Lasso and Elastic-Net Machine Learning Techniques, Neural Comput. Appl. 33 (24) (2021) 17131–17145, https://doi.org/10.1007/s00521-021-06304-z.
[31] J.H. Friedman, Multivariate adaptive regression splines, Ann. Stat. 19 (1) (1991) 1–67, https://doi.org/10.1214/aos/1176347963.
[32] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67, https://doi.org/10.1016/j.advengsoft.2016.01.008.
[33] Y. Zhang, X. Xu, Predicting doped Fe-based superconductor critical temperature from structural and topological parameters using machine learning, Int. J. Mater. Res. 112 (1) (2021) 2–9, https://doi.org/10.1515/ijmr-2020-7986.
[34] Y. Zhang, X. Xu, Predicting the superconducting transition temperature of high-temperature layered superconductors via machine learning, Phys. C. Supercond. 595 (2022), 1354031, https://doi.org/10.1016/j.physc.2022.1354031.
[35] G. Revathy, V. Rajendran, B. Rashmika, P.S. Kumar, P. Parkavi, J. Shynisha, Random forest regressor based superconductivity materials investigation for critical temperature prediction, Mater. Today.: Proc. 66 (3) (2022) 648–652, https://doi.org/10.1016/j.matpr.2022.03.515.
[36] R. Genuer, J.-M. Poggi, Random Forests with R, Springer,, San Francisco, USA, 2020.
[37] C. Smith, M. Koning, Decision Trees and Random Forests: A Visual Introduction For Beginners, Blue Windmill Media, British Columbia,, Canada, 2017.
[38] M.P. Deisenroth, Mathematics for Machine Learning, Cambridge University Press,, New York, USA, 2020.
[39] T. Hastie, R. Tibshirani, J.H. Friedman, The Elements of Statistical Learning, Springer-Verlag,, New York, USA, 2003.
[40] C.M. Bishop, Pattern Recognition and Machine Learning, Springer,, Cambridge, United Kingdom, 2006.
[41] E.K.P. Chong, S.H. Zak, An Introduction to Optimization, Wiley,, New York, 2013.
[42] R.C. Eberhart, Y. Shi, J. Kennedy, Swarm Intelligence, Morgan Kaufmann,, San Francisco, USA, 2001.
[43] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization (https://dl.acm.org/doi/), J. Mach. Learn. Res. 133 (2012) 281–305, https://doi.org/10.5555/2188385.2188395.
[44] C.C. Aggarwal, Linear Algebra and Optimization for Machine Learning, Springer,, New York, 2020.
[45] T. Agrawal, Hyperparameter Optimization in Machine Learning: Make Your Machine Learning and Deep Learning Models More Efficient, Apress, New York, 2020.
[46] M. Hassoun, Fundamentals of Artificial Neural Networks, The MIT Press,, Bradford Book, Massachusetts, USA, 1995.
[47] A.J. Shepherd, Second-order Methods for Neural Networks: Perspectives in Neural Computing, Springer-Verlag,, London, 1997.
[48] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice–Hall,, Singapore, 1999.
[49] T.L. Fine, Feed-forward Neural Network Methodology, Springer-Verlag,, New York, 1999.
[50] J.R. Quinlan, Learning with continuous classes, in: Proceedings of Australian Joint Conference on Artificial Intelligence, World Scientific Press, Hobart, 1992, pp. 343–348.
[51] T.M. Mitchell, Machine Learning, McGraw-Hill Company Inc,, New York, 1997.
[52] O. Kisi, Pan evaporation modeling using least square support vector machine, multivariate adaptive regression splines and M5 model tree, J. Hydrol. 528 (2015) 312–320, https://doi.org/10.1016/j.jhydrol.2015.06.052.
[53] S. Weisberg, Applied Linear Regression, Wiley,, New York, USA, 2013.
[54] P. Roback, J. Legler, Beyond Multiple Linear Regression: Applied Generalized Linear Models and Multilevel Models in R, CRC Press,, Boca Raton, USA, 2020.

[55] Y. He, C. Chen, B. Li, Z. Zhang, Prediction of near-surface air temperature in glacier regions using ERA5 data and the random forest regression method, Remote Sens. Appl.: Soc. Environ. 28 (2022), 100824, https://doi.org/10.1016/j.rsase.2022.100824.
[56] S. Kwak, J. Kim, H. Ding, X. Xu, R. Chen, J. Guo, H. Fu, Machine learning prediction of the mechanical properties of γ-TiAl alloys produced using random forest regression model, J. Mater. Res. Technol. 18 (2022) 520–530, https://doi.org/10.1016/j.jmrt.2022.02.108.
[57] K.C. Onyelowe, T. Gnananandarao, A.M. Ebid, Estimation of the erodibility of treated unsaturated lateritic soil using support vector machine-polynomial and -radial basis function and random forest regression techniques, Clean. Mater. 3 (2022), 100039, https://doi.org/10.1016/j.clema.2021.100039.
[58] H. Jiang, L. Mei, Y. Wei, R. Zheng, Y. Guo, The influence of the neighbourhood environment on peer-to-peer accommodations: a random forest regression analysis, J. Hosp. Tour. Manag. 51 (2022) 105–118, https://doi.org/10.1016/j.jhtm.2022.02.028.
[59] SuperCon Database, Technical report, National Institute for Materials Science (NIMS), Japan, 2021.
[60] D. Dua, C. Graff, UCI machine learning repository, Technical report, University of California, Irvine, School of Information and Computer Sciences, 2019.
[61] M. Minsky, S. Papert, Perceptrons: An Introduction to Computational Geometry, MIT Press,, New York, USA, 1988.
[62] M. Olazaran, A sociological study of the official history of the perceptrons controversy, Soc. Stud. Sci. 26 (3) (1996) 611–659, https://doi.org/10.1177/030631296026003005.
[63] M. Pal, M5 model tree for land cover classification, Int. J. Remote Sens. 27 (4) (2006) 825–831, https://doi.org/10.1080/01431160500256531.
[64] M. Pal, S. Deswal, M5 model tree based modelling of reference evapotranspiration, Hydrol. Process. 23 (10) (2009) 1437–1443, https://doi.org/10.1002/hyp.7266.
[65] A. Rahimikhoob, M. Asadi, M. Mashal, A comparison between conventional and M5 model tree methods for converting pan evaporation to reference evapotranspiration for semi-arid region, Water Resour. Manag. 27 (14) (2013) 4815–4826, https://doi.org/10.1007/s11269-013-0440-y.
[66] D.P. Solomatine, Y.P. Xue, M5 model trees and neural networks: application to flood forecasting in the upper reach of the Hual River in China, J. Hydrol. Eng. 9 (6) (2004) 491–501, https://doi.org/10.1061/(ASCE)1084-0699(2004)9:6(491).
[67] J. Fox, Applied Regression Analysis and Generalized Linear Models, SAGE Publications,, Los Angeles, USA, 2015.
[68] D.C. Montgomery, E.A. Peck, G.G. Vining, Introduction to Linear Regression. Analysis, Wiley,, New York, 2021.
[69] J.V. Stone, Linear Regression: A Tutorial Introduction to the Mathematics of Regression Analysis, Sebtel Press,, London, UK, 2022.
[70] D. Freedman, R. Pisani, R. Purves, Statistics, W.W. Norton & Company,, New York, USA, 2007.
[71] G.J. Knafl, K. Ding, Adaptive Regression for Modeling Nonlinear Relationships, Springer,, Berlin, 2016.
[72] J.T. McClave, T.T. Sincich, Statistics, Pearson,, New York, 2016.
[73] L. Wasserman, All of Statistics: A Concise Course in Statistical Inference, Springer,, New York, USA, 2003.
[74] D. Simon, Evolutionary Optimization Algorithms, Wiley,, New York, USA, 2013.
[75] R.R. Picard, R.D. Cook, Cross-validation of regression models, J. Am. Stat. Assoc. 79 (1984) 575–583, https://doi.org/10.1080/01621459.1984.10478083.
[76] P.J. García–Nieto, E. García–Gonzalo, J. Bové, G. Arbat, M. Duran–Ros, J. Puig–Bargues, Modeling pressure drop produced by different filtering media in microirrigation sand filters using the hybrid ABC-MARS-based approach, MLP neural network and M5 model tree, Comput. Electron. Agr. 139 (2017) 65–74, https://doi.org/10.1016/j.compag.2017.05.008.
[77] B. Efron, R. Tibshirani, Improvements on cross-validation: The.632 + bootstrap method, J. Am. Stat. Assoc. 92 (438) (1997) 548–560, https://doi.org/10.2307/2965703.
[78] I.H. Witten, E. Frank, M.A. Hall. Data Mining: Practical Machine Learning Tools and Techniques, fourth ed..,, Morgan Kaufmann,, Burlington, Massachusetts, USA, 2016.
[79] M.A. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, ACM SIGKDD Explor 11 (1) (2009) 10–18, https://doi.org/10.1145/1656274.1656278.
[80] C. Molnar, Interpretable Machine Learning: A Guide for Making Black Box Models Explainable, Independently published, 2nd ed., 2022. christophm.github.io/interpretable-ML-book/.