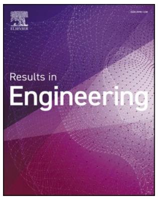
Contents lists available at ScienceDirect

Results in Engineering

journal homepage: www.sciencedirect.com/journal/results-in-engineering



Research paper

An optimized machine-learning tool to predict heat treatment response of hot-work tool steels

Venu Yarasu [*], Bojan Podgornik

Institute of Metals and Technology, Lepi pot 11, SI-1000, Ljubljana, Slovenia

ARTICLE INFO

Keywords:
Hot-work tool steels
Machine learning
Hardness
Fracture toughness
Graphical user interface

ABSTRACT

Hot-work tool steels are essential materials in high-performance tooling applications due to superior thermal fatigue resistance, strength, and toughness at elevated temperatures. Accurate prediction of mechanical properties, particularly fracture toughness (K_{IC}) and hardness (HRC), is crucial for their efficient design. However, conventional methods like tempering diagrams are often time-consuming, expensive, and lack broad applicability. This study proposes an optimized machine-learning (ML) framework for predicting HRC and K_{IC} values of hot-work tool steels based on 2.564 experimental data points involving 13 input parameters, including chemical composition, processing routes, and heat treatment conditions. A comparative analysis of ten ML algorithms-ranging from decision trees and k-nearest neighbors to ensemble methods-was conducted. The originality of this work lies in the integration of advanced ensemble learning, SHAP-based model interpretation, and the creation of a graphical user interface (GUI) for real-time property prediction. Models were validated using both conventional test/train splits and 5-fold cross-validation, with their performance evaluated using mean squared error (MSE), mean absolute error (MAE), root mean squared error (RMSE), mean absolute percentage error (MAPE), and R-squared. CatBoost and stacking achieved highest accuracy of 99% for HRC and 94% for K_{IC}, with mean absolute errors of 0.3 HRC and 3.3 MPa.m$^{-1/2}$, respectively. Additionally, the GUI tool, which integrates machine learning-derived empirical equations, showed impressive prediction accuracy (R^2 = 99% for HRC and 95% for K_{IC}), providing a valuable resource for materials engineers. This research presents a strong, data-driven approach to accelerate the development of tool steel while reducing the need for extensive experimental procedures.

1. Introduction

Hot-work tool steels are widely used in industries such as automotive, aerospace, and manufacturing due to their high hardness, wear resistance, high-temperature strength, and fracture toughness. These materials play a crucial role in processes like die casting, forging, and extrusion, where dimensional stability and thermal fatigue resistance are vital. As the performance requirements in sectors like automotive and aerospace keep increasing, the optimization of mechanical properties-particularly hardness and fracture toughness-has become critical for ensuring tool performance, durability, and service life [1–7]. Traditionally, optimizing the mechanical properties of hot-work tool steels has relied on extensive experimental trials involving variations in chemical composition, processing routes, and heat treatment conditions. However, this conventional approach is time-consuming, costly, and often inefficient, especially when dealing with complex property interdependencies. The limitations of empirical methods have highlighted the need for faster and more reliable predictive frameworks in alloy design and processing optimization [1–3].

The performance of tool steels is influenced by a well-balanced chemical composition [8], the methods of production route [9,10], and particularly heat treatment process, which plays a crucial role in determining the final microstructure and mechanical characteristics [11–13]. Critical properties such as temper resistance, tensile strength, wear and fatigue resistance, and most importantly, hardness and fracture toughness, must be balanced to achieve stability at elevated temperatures and resistance to cracking under operational conditions [13–16]. The relationship among processing parameters, microstructure, and the resultant properties in hot-work tool steels is complex and multifaceted, rendering traditional optimization methods particularly difficult.

Recent advancements in materials science have resulted in the

[*] Corresponding author.
E-mail addresses: venu.yarasu@imt.si, venu.yarasu@gmail.com (V. Yarasu), bojan.podgornik@imt.si (B. Podgornik).

https://doi.org/10.1016/j.rineng.2025.105260
Received 31 March 2025; Received in revised form 6 May 2025; Accepted 8 May 2025
Available online 8 May 2025

Table 1
Summary of selected ML-based studies for property prediction in materials science.

Inputs	Outputs	Dataset size/Test-Train Ratio	Accuracy/ Error	ML technique	Ref.
Chemical composition, heat treatment parameters	Hardness, fracture toughness	NA	95%	Feedforward ANN	[7]
Pulse on time, wire span, servo gap voltage	Cutting velocity, surface roughness	20 / NA	0–6% error	RSM + ANN	[19]
Composition, heat treatment temps, microstructural descriptors	hardness, yield strength	90 / 10	91%	RF, XGB, ANN, GBM	[22]
Cryogenic process, current, pulse duration	Electrode and work piece wear	176 / NA	≥ 99%	AutoML	[23]
Composition, heat treatment temperatures, inclusion parameters	Fatigue strength, tensile strength, fracture strength, hardness	360 / NA	≥ 98%	RF, Linear Least-Square (LLS), KNN, ANN	[24]
Preplaced aggregate concrete (cement, sand, water, gravel, etc.)	Compressive and tensile strength	1500+ / NA	≥ 97%	AdaB, GBM, XGB, ETR, SVM, ANN, RAGN-R	[26]
Structural material (nanomaterial length, diameter, strength, modulus, etc.)	Compressive, tensile, and flexural strength	NA, 80 / 20	98% ≥	RF, RR, DT, KNN, MLP, XGB, etc.	[27]
Fiber-reinforced concrete beams (fiber length, type, density, ratio, etc.)	Load capacity, ductility	193 / NA	≥ 97%	RF, GB, NN	[28]
Structural material (fiber aspect ratio, volume fraction, water, curing temperature, etc.)	Compressive strength	538 / NA	≥ 98%	BR, XGB, GBM, LGBM, ETR, RF, KNN	[29]
Laser heat treatment of AISI H13 tool steel	Hardness distribution	71 images / NA	≥ 94%	Deep Learning, CNN	[30]
D2 Steel (composition, heat treatment, tempering temperatures)	Hardness	54 images / 80 / 20	≥ 99%	DT, AdaB, XGB, RF	[32]

creation of more advanced techniques for analyzing and predicting the properties of materials, aimed at addressing these challenges. Computational techniques like finite element analysis (FEA) and phase-field modeling have given insights into the microstructural evolution during heat treatment. For example, Wrobel et al. did numerical analysis of quenching phenomena using complex model of hardening of hot-work tool steel. In this study the numerical algorithm of thermal phenomena is based on solving the heat transfer equations using finite element method [17]. Similarly, Eser et al. did FEM simulations to prepare multiscale models for tempering of AISI H13 hot-work tool steel to predict the microstructural evolution and mechanical properties [14]. However, these methods can be computationally intensive and require detailed knowledge of the underlying physical phenomena, making them less accessible for rapid design and property prediction. Recently, statistical techniques including Response Surface Methodology (RSM), Taguchi optimization, and Grey Relation Analysis have been employed to enhance heat treatment and processing parameters [18,19]. For instance, Manoj et al. utilized RSM alongside artificial neural networks (ANN) to refine wire EDM parameters, demonstrating that machine learning models provided greater accuracy compared to conventional approaches [19].

In recent years machine learning (ML) has become a powerful tool in materials science to uncover the complex relationships between input variables and material properties such as microstructure, hardness, toughness, tribological and erosion properties [20–29]. ML models, such as random forests (RF), gradient boosting machines (GBM), artificial neural networks (ANN), support vector machines (SVM), and extreme gradient boosting (XGB), have shown significant potential in predicting mechanical and tribological properties, thereby decreasing the dependence on expensive experimental procedures. For instance, Kazemi et al. employed ML models like RF, ANN, and GBM to predict the mechanical properties of advanced concretes such as fiber-reinforced and aggregate-based concrete with high accuracy [26–29]. In another study, Yang and Dou applied ML techniques to estimate the hardness and yield strength of oxide dispersion-strengthened (ODS) steels, with XGB models achieving the best performance in terms of R^2 and mean absolute error (MAE) [22]. Similarly, Gui et al. used support vector optimization to design carbon steels with improved strength and ductility [25].

Despite the progress made in the field, the application of ML in the design and property prediction of hot-work tool steels is still quite limited. Podgornik et al. [7] implemented a feedforward backpropagation neural network to enhance the hardness and toughness of these materials. Additionally, Oh and Ki developed a convolutional neural network (CNN) model that accurately predicts hardness profiles from laser heat treatment images of AISI H13, achieving a prediction accuracy of 94.4% [30]. Several previous studies have applied ML models for similar analyses of tool steels. For example, Kahrobaee et al. applied artificial neural networks to forecast the magnetic hysteresis of AISI D2 steel based on heat treatment parameters such as austenitizing and tempering temperatures. In another instance, Pillai and Karthikeyan et al. [31] utilized support vector machines to estimate time-temperature-transformation curves for comparable steels. Furthermore, another investigation [32] successfully predicted the Vickers hardness of D2 steel from indentation imprint images without relying on diagonal measurements; this study compared various methods, including decision trees, adaptive boosting, extreme gradient boosting, and random forest, concluding that random forest yielded the

Table 2
Composition (wt.%) and manufacturing process conditions used in the current study.

C	Si	Mn	Cr	Mo	V	Ni	W	N	Process
0.36	0.22	0.25	5.02	1.25	0.43	0.00	0.00	0.00	ESR
0.37	0.25	0.26	4.96	1.25	0.43	0.00	0.00	0.00	CoM
0.38	1.00	0.40	5.10	1.25	0.40	0.00	0.00	0.00	CoM
0.37	0.97	0.40	5.33	1.20	0.33	0.00	0.00	0.00	CoM
0.38	1.10	0.40	5.00	1.30	0.40	0.00	0.00	0.00	ESR
0.38	0.20	0.55	5.00	1.75	0.55	0.00	0.00	0.00	ESR
0.38	1.00	0.40	5.30	1.30	0.40	0.00	0.00	0.00	CoM
0.35	0.20	0.50	5.00	2.30	0.60	0.00	0.00	0.00	PM
0.36	0.25	0.40	5.20	1.90	0.55	0.00	0.00	0.00	CoM
0.53	0.27	0.27	4.93	2.94	0.59	0.55	0.00	0.00	ESR
0.36	0.30	0.30	5.00	2.30	0.60	0.00	0.00	0.00	ESR
0.37	0.25	0.45	4.90	1.61	0.59	1.60	0.01	0.02	ESR
0.37	1.05	0.38	4.96	1.27	0.38	0.00	0.00	0.00	PM

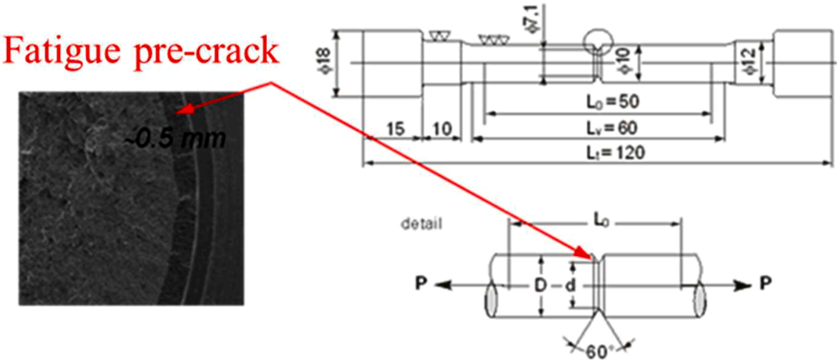


Figure 1. Dimensions of Circumferentially Notched and Fatigue Pre-Cracked Tensile Bar specimens (CNPTB).

best results. However, many of these studies are constrained by limited datasets, the use of single ML models, or a restricted number of input features. Additionally, the practical implementation of user-friendly prediction tools remains largely unaddressed. To fill these gaps, the current study aims to develop and compare ten different machine learning models to predict two essential properties-hardness (HRC) and fracture toughness (K_{IC}) in hot-work tool steels. This research utilizes a dataset containing 2.574 records that encompass chemical composition, processing methods, and heat treatment parameters. A wide range of ML models, including linear, tree-based, boosting, neural, and ensemble methods, is employed, with comparative evaluations conducted using training, testing, and 5-fold cross-validation metrics. Moreover, feature importance analysis and SHAP-based model interpretations are integrated to provide insights into the relationships between structure and properties. This integrated approach offers a practical and efficient pathway for tool steel designers and quality control engineers to predict performance outcomes and optimize hot-work tool steels. Table 1 summarizes key recent ML-based studies in materials science, highlighting the novelty and relevance of this work.

2. Material and methods

2.1. Materials and treatments

The current study involved thirteen commercial hot-work tool steels, each steel with different carbon and alloying elements in wt.% as shown in Table 2. These steels were sourced from commercial production using various manufacturing technologies, including conventional methods (CoM), electro slag remelting (ESR), and powder metallurgy (PM), and were supplied in a soft-annealed state. The vacuum hardening process was performed at temperatures between 990°C and 1050°C, followed by a single-step quenching in a horizontal vacuum furnace using uniform high-pressure nitrogen gas at 1.05 bar. The hardening treatment comprised preheating, austenitizing, and quenching. Preheating was done at 650°C and 850°C for 10 minutes to ensure a uniform temperature distribution between the surface and core. After the hardening process, double tempering was carried out at temperatures ranging from 540°C to 640°C for each group of hot-work tool steel, with each tempering cycle lasting 2 hours. All vacuum heat treatments followed recommendations from tool steel suppliers.

2.2. Fracture toughness & Hardness

Circumferentially notched tensile bar samples, as illustrated in Figure 1, were prepared from rolled or forged bars through CNC machining. After that, these samples were subjected to fatigue pre-cracking under rotational bending at a load of 65N−70N for 5 minutes, which resulted in a pre-crack length between 0.2 and 0.5 mm in the V-notch. The Circumferentially Notched and Fatigue Pre-Cracked Tensile Bar specimens (CNPTB) for each type of steel were then heat-treated as previously described.

For the assessment of hardness and fracture toughness, CNPTB specimens were used [33]. Rockwell hardness tests were performed on the circumferential surface (12 mm diameter) of both fractured sections of the CNPTB specimen using a Wilson Rockwell B 2000 machine, following the ISO 6508-1:2016 [34] standard protocols. Fracture toughness was evaluated using fatigue pre-cracked specimens according to the plain strain model. A minimum of nine tests, conducted identically, were averaged to measure the fracture toughness (K_{IC}). Tensile loading to fracture was performed at room temperature with an Instron 1255 tensile test machine at a crosshead speed of 1.0 mm/min. Finally, the area of brittle fracture was measured to assess fracture toughness.

Fracture toughness was calculated using Eq. 1 [35]:

$$K_{IC} = \frac{P}{D^{3/2}}\left(-1.27 + 1.72\left(\frac{D}{d}\right)\right) \tag{1}$$

Here, P refers to the load at fracture, D stands for the outside diameter of the non-notched section (10 mm), and d represents the diameter of the brittle fractured area. As mentioned, minimum of nine specimens were prepared for each test group, and the diameter of the brittle fractured area was measured for all specimens.

2.3. Data preparation and processing

To obtain an accurate correlation between the chemical composition, processing routes, heat treatment conditions, hardness, and fracture toughness, the collected data were verified for any missing values and outliers. Missing values were verified using visual inspection, and the Interquartile Range (IQR) method was applied to identify outliers, resulting in the identification of 10 outliers within the current dataset. The IQR method is a robust statistical technique used to identify potential outliers by establishing a range that encompasses the central 50% of the data. Following the removal of these outliers, the refined dataset comprises 2.564 instances derived from experimental studies conducted at the IMT. This dataset encompasses thirteen varieties of hot-work tool steels, each characterized by distinct chemical compositions (numerical), three different manufacturing processes (categorical), eight varying hardening temperatures (numerical), twelve tempering temperatures (numerical), and two critical properties: hardness and fracture toughness (numerical). A summary of the input and output features utilized for the development of the machine learning model is provided below.

The input variables are:

- Chemical composition: Carbon (C), Silicon (Si), Manganese (Mn), Chromium (Cr), Molybdenum (Mo), Vanadium (V), Nickel (Ni), Tungsten (W), Nitrogen (N)
- Processing method: Electro Slag Remelting (ESR), Conventional, and Powder Metallurgy (PM)

Table 3
Summary statistics of variables used in the dataset.

Feature	Min.	Max.	Mean.	Standard Error	Mode	Median	Standard Deviation
C	0.25	0.52	0.28	0.001	0.28	0.27	0.05
Si	0.20	1.10	0.54	0.008	0.25	0.25	0.39
Mn	0.20	0.55	0.39	0.002	0.40	0.40	0.10
Cr	4.90	5.22	5.05	0.002	5.00	5.00	0.10
Mo	1.20	2.94	1.59	0.009	1.25	1.25	0.45
V	0.22	0.60	0.47	0.002	0.40	0.42	0.11
Ni	0.00	1.60	0.17	0.009	0.00	0.00	0.48
W	0.00	0.01	0.00	0.000	0.00	0.00	0.00
N	0.00	0.02	0.00	0.000	0.00	0.00	0.01
Process	0.00	2.00	0.75	0.013	1.00	1.00	0.66
Hardening (°C)	990.00	1050.00	1016.26	0.346	1020.00	1020.00	17.54
Tempering (°C)	540.00	640.00	588.98	0.496	620.00	590.00	25.10
HRC	26.75	59.87	48.26	0.098	52.20	49.40	4.98
K_{IC}	16.65	111.12	48.30	0.459	22.22	44.11	23.26

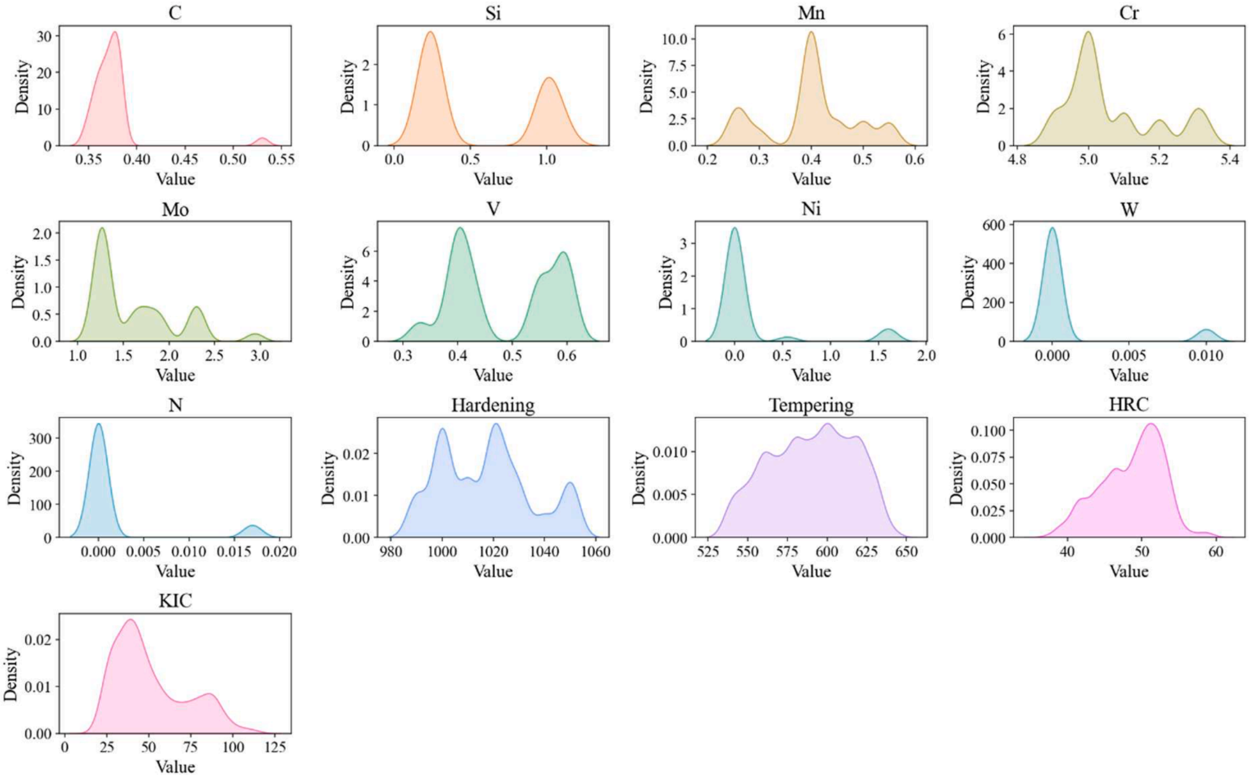


Figure 2. KDE distribution of features: before min-max normalization.

- Heat treatment parameters: Hardening temperature and Tempering temperature

The output variables are:

- Hardness (HRC)
- Fracture toughness (K_{IC})

For the processing of categorical features, the Label Encoding method was applied, converting the three processing routes-conventional, ESR, and powder metallurgy-into integer values of 0, 1, 2, respectively.

Additionally, the various descriptors can differ significantly in their values, potentially leading to misleading results during modeling. To address this, the Min-Max normalization method was adopted to preprocess the descriptors to eliminate the influence of numerical differences on the prediction performance of regression models. The Min-Max normalization method is represented by the following Eq. 2 [36], allowing all features to be mapped to the range [0,1].

$$y = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{2}$$

Where X_{min} and X_{max} represent the minimum and maximum values for feature X. In this study, the dataset contains twelve features and two target variables, and the descriptive statistics of each feature and target variables for averaged results before normalization is shown in Table 3. Furthermore, the Kernel Density Estimation (KDE) distributions of features were presented in Figure 2. KDE, or Kernel Density Estimation, is a non-parametric technique utilized for estimating the probability density function of a continuous variable. This method generates a smoothed curve that illustrates the data distribution, facilitating the identification of characteristics such as skewness, modality, and outliers. In this research, KDE plots were employed to depict the distribution of input features prior to normalization, as illustrated in Figure 2. This visualization is instrumental in comprehending the initial spread and shape of the data prior to any scaling transformations.

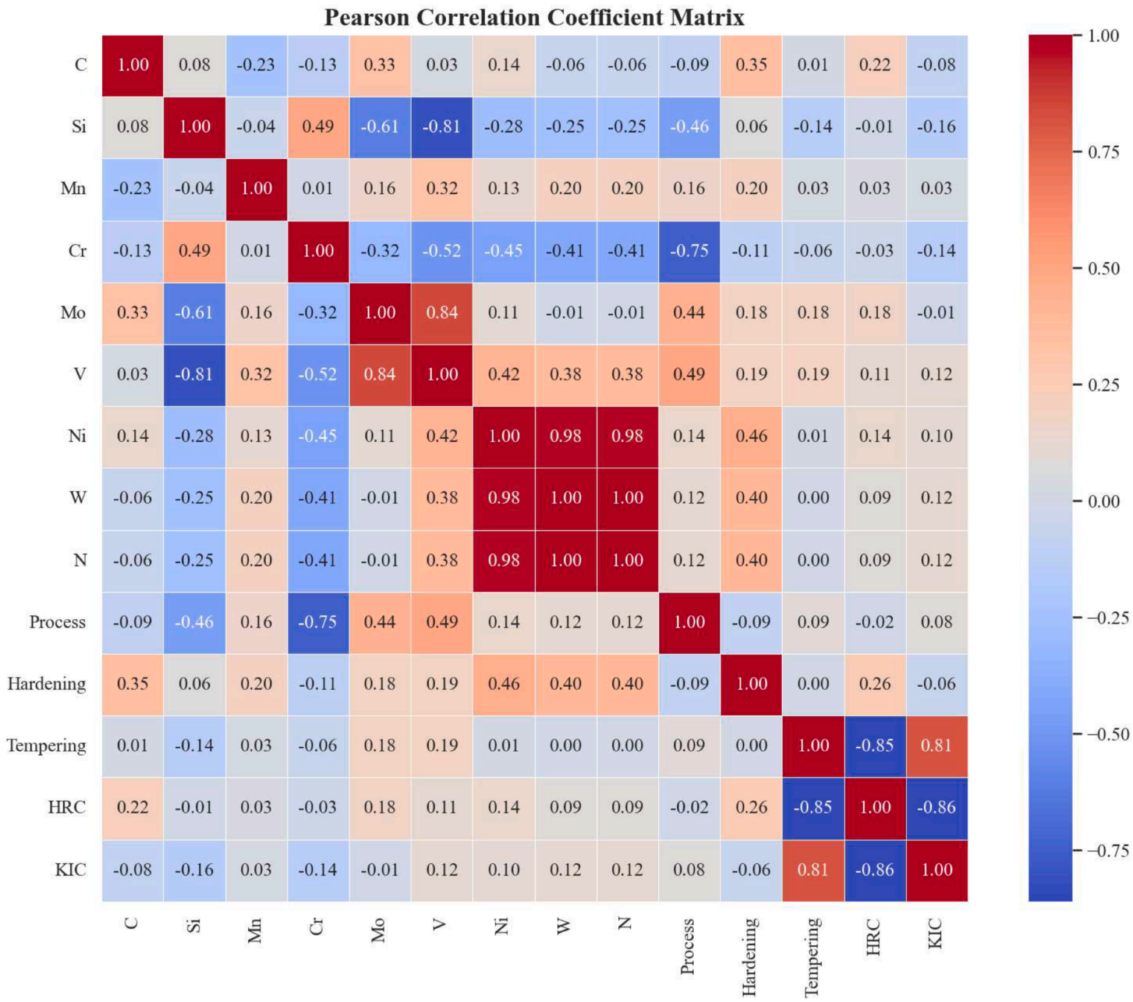


Figure 3. Pearson Correlation Matrix: showing linear relationships between variables.

The Pearson Correlation Matrix (PCM), illustrated in Figure 3, was employed to investigate the linear relationships between each input feature and the output variables, specifically hardness and fracture toughness. The Pearson correlation coefficient (r) serves as a statistical indicator that ranges from -1 to $+1$. A coefficient approaching $+1$ signifies a robust positive linear correlation, meaning that an increase in one variable corresponds with an increase in the other. In contrast, a coefficient close to -1 indicates a strong negative linear correlation, where an increase in one variable results in a decrease in the other. A coefficient near 0 suggests a minimal linear relationship. This analysis facilitates the identification of features that may significantly impact the output variables, thereby assisting in the initial selection of relevant features.

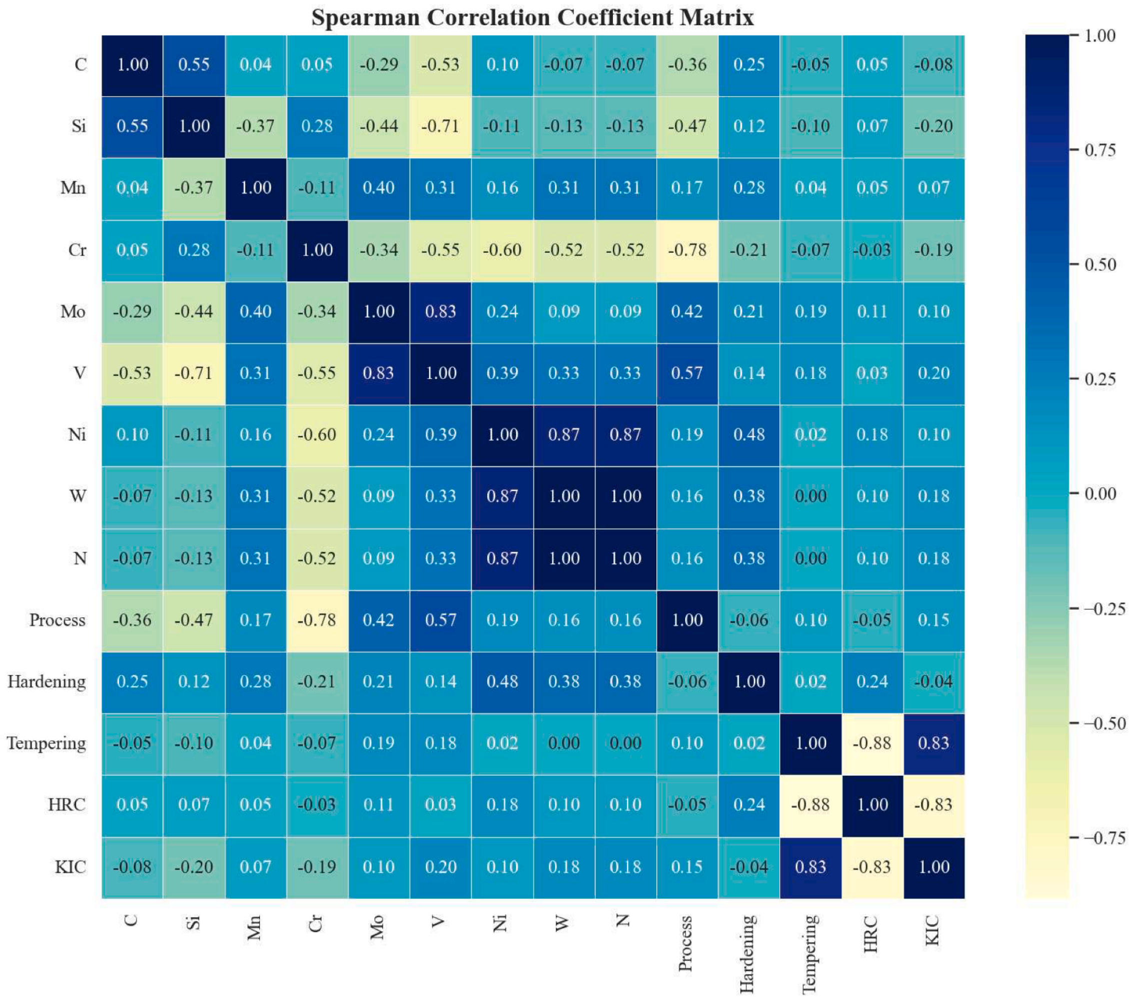
Based on the PCM analysis, tempering shows the strong negative linear correlation with HRC, followed by weak positive correlations with hardening and carbon. For K_{IC}, tempering emerges as the most significant linear predictor with a positive correlation, while vanadium, nickel, tungsten, and nitrogen exhibit weak positive linear relationships. In contrast, elements such as vanadium, nickel, tungsten, and nitrogen reveal weak positive linear associations, whereas silicon and chromium present weak negative correlations. These findings suggest that tempering is a critical linear predictor for both HRC and K_{IC}, while elemental composition features generally show weaker linear associations.

Additionally, to validate the suitability of the selected input variables, the Spearman Correlation Matrix (SCM) is constructed, as shown in Figure 4. The PCM evaluates the linear relationships between continuous variables and is commonly used to quantify direct proportionality or inverse trends. In contrast, the SCM evaluates monotonic relationships (both linear and non-linear) by assessing the rank-order correlations between variables.

The rationale for using both matrices lies in the fact that some relationships between input features and target outputs (HRC and K_{IC}) may not follow a strictly linear pattern. While the PCM highlights linear dependencies, the SCM helps detect any monotonic but non-linear associations. By comparing both matrices, we ensured a comprehensive understanding of the data structure. For instance, SCM highlights similar trends like PCM, with tempering again being a dominant monotonic predictor for HRC (negative correlation), and hardening, nickel, and molybdenum showing weak positive monotonic associations. For K_{IC}, tempering exhibits a strong positive monotonic correlation, while vanadium, tungsten, and nitrogen reveal slightly stronger positive monotonic correlations compared to their Pearson values. Silicon and chromium continue to show weak negative monotonic relationships. These findings reinforce the importance of tempering in predicting both target properties and suggest the presence of non-linear monotonic relationships for certain elemental features, particularly in relation to K_{IC}. Notably, all input variables exhibited non-zero correlations in both PCM and SCM, justifying their inclusion in the regression modeling process.

3. Machine learning algorithms

Regression models were developed using the Scikit-learn machine learning library to predict the hardness and fracture toughness of hot-



	C	Si	Mn	Cr	Mo	V	Ni	W	N	Process	Hardening	Tempering	HRC	KIC
C	1.00	0.55	0.04	0.05	-0.29	-0.53	0.10	-0.07	-0.07	-0.36	0.25	-0.05	0.05	-0.08
Si	0.55	1.00	-0.37	0.28	-0.44	-0.71	-0.11	-0.13	-0.13	-0.47	0.12	-0.10	0.07	-0.20
Mn	0.04	-0.37	1.00	-0.11	0.40	0.31	0.16	0.31	0.31	0.17	0.28	0.04	0.05	0.07
Cr	0.05	0.28	-0.11	1.00	-0.34	-0.55	-0.60	-0.52	-0.52	-0.78	-0.21	-0.07	-0.03	-0.19
Mo	-0.29	-0.44	0.40	-0.34	1.00	0.83	0.24	0.09	0.09	0.42	0.21	0.19	0.11	0.10
V	-0.53	-0.71	0.31	-0.55	0.83	1.00	0.39	0.33	0.33	0.57	0.14	0.18	0.03	0.20
Ni	0.10	-0.11	0.16	-0.60	0.24	0.39	1.00	0.87	0.87	0.19	0.48	0.02	0.18	0.10
W	-0.07	-0.13	0.31	-0.52	0.09	0.33	0.87	1.00	1.00	0.16	0.38	0.00	0.10	0.18
N	-0.07	-0.13	0.31	-0.52	0.09	0.33	0.87	1.00	1.00	0.16	0.38	0.00	0.10	0.18
Process	-0.36	-0.47	0.17	-0.78	0.42	0.57	0.19	0.16	0.16	1.00	-0.06	0.10	-0.05	0.15
Hardening	0.25	0.12	0.28	-0.21	0.21	0.14	0.48	0.38	0.38	-0.06	1.00	0.02	0.24	-0.04
Tempering	-0.05	-0.10	0.04	-0.07	0.19	0.18	0.02	0.00	0.00	0.10	0.02	1.00	-0.88	0.83
HRC	0.05	0.07	0.05	-0.03	0.11	0.03	0.18	0.10	0.10	-0.05	0.24	-0.88	1.00	-0.83
KIC	-0.08	-0.20	0.07	-0.19	0.10	0.20	0.10	0.18	0.18	0.15	-0.04	0.83	-0.83	1.00

Figure 4. Spearman Correlation Matrix: showing monotonic relationship between the variables.

work tool steels. This library is an open-source Python framework that offers a wide range of advanced machine learning methodologies, including ensemble learning, boosting techniques, tree-based algorithms, neural networks, and k-nearest neighbors, which are appropriate for medium-scale supervised and unsupervised learning tasks [37]. In total, ten machine learning algorithms were utilized in the creation of the predictive regression models, comprising Linear Regression (LR), Decision Tree (DT), Random Forest (RF), Gradient Boosting Machine (GB), Extreme Gradient Boosting (X.GB), CatBoost (CatB), AdaBoost (AdaB), K-nearest Neighbors (kNN), and Stack of ensemble models.

3.1. Linear regression (LR)

Linear regression (LR) represents the most fundamental type of statistical model designed to identify the linear relationship between a dependent variable and one or more independent variables. When the model incorporates a single independent variable, it is referred to as simple linear regression, whereas the inclusion of multiple independent variables designates it as multiple linear regression. Statistically, the relationship between the dependent variable and one or more independent variables can be expressed through linear regression, which seeks to determine the optimal linear correlation between input and output variables by minimizing the discrepancy between predicted and actual values [38]. The linear regression is represented by Eq. 3:

$$Y = b0 + b1X1 + b2X2 + \ldots + bnXn + \varepsilon \tag{3}$$

In this equation, X1, X2, …, Xn denote the independent variables, Y signifies the dependent variable, b0 represents the intercept, b1, b2, b3, …, bn are the coefficients corresponding to the independent variables, and ε is the error term. Within the framework of machine learning algorithms, the process involves utilizing a cost function or loss function, which is minimized through various optimization techniques to enhance the accuracy of the machine learning model. Nevertheless, the performance of linear regression is constrained by the absence of hyperparameters when compared to other machine learning models.

3.2. Decision tree (Tree)

The decision tree represents one of the most fundamental machine learning models frequently utilized in both classification and regression tasks. It initiates from a root node, which is determined by the most influential attribute, assessed through a purity index based on information gain. The tree subsequently expands into branches that perform tests on various attributes, guided by their respective information gains. Each branch may further divide into sub-branches or terminate as leaf nodes. The terminal points of the tree, known as leaves, signify the outcomes or decisions derived from the training data, represented either as a class label or a continuous value [39,40]. The prediction formula for a decision tree is expressed in Eq. 4:

$$Prediction^{RFR} = \frac{1}{K}\sum_{k=1}^{K} h_k(x) \tag{4}$$

In this equation, K denotes the total number of decision trees, while $h_k(x)$ represents the average prediction across these K trees.

The hyperparameters relevant to the configuration of a decision tree regression model are outlined as follows:

- N estimators: This parameter indicates the number of trees to be trained to construct the forest, which is determined based on the dataset's characteristics, including the number of instances, attributes, and features. Typically, this value ranges from 10 to over 500 trees, and for the current dataset, the full range has been used in hyperparameter tuning.
- Max depth: This parameter specifies the maximum number of splits allowed for any branch of the tree until all leaves become pure nodes. It can range from 0 to 100. If set to 'none', all nodes will continue to expand until every leaf is pure or meets the minimum sample split criteria. For the current dataset, due to its smaller and more precise nature, the tree was allowed to grow fully by setting max depth to 'none'.
- Min samples split: This parameter defines the minimum number of instances required to split a branch node, with a default value typically set at min samples split = 5.
- Min samples leaf: This parameter indicates the minimum number of instances needed to form a leaf node, which helps to smooth the regression line; the standard value is generally set at 2.

3.3. Random forest (RF)

Random forest (RF) is an ensemble-based supervised machine learning technique that constructs multiple decision trees and aggregates their outputs to produce a final prediction. Each decision tree utilizes principles of information gain or entropy for classification tasks and mean squared error (MSE) for regression tasks, thereby enhancing the predictive accuracy of the trees. Each tree is trained on a randomly chosen subset of both the input data and the features, which mitigates the risk of overfitting and enhances the model's generalization capabilities. The fundamental approach involves generating numerous decision trees through random selection of both attributes and samples, subsequently combining their predictions to arrive at a final output [41, 42]. In RF regression, the final prediction is calculated using the same formula as in the decision tree model. Additionally, the hyperparameters pertaining to the random forest model were largely consistent with those employed in the decision tree model, as detailed in Section 3.1.2.

3.4. Multi-layer perception (MLP)

Multi-layer perceptron (MLP) represents a fundamental architecture in neural networks characterized by the presence of multiple hidden layers, each containing distinct neurons. These hidden layers are interconnected, with the initial hidden layer receiving input from the dataset and the final hidden layer producing the output dataset. The MLP employs various optimization algorithms in conjunction with the back-propagation technique to facilitate learning. The nodes in the input layer process the incoming data, while the nodes in the output layer generate the final results. The intermediate layers apply non-linear activation functions to modify the input, thereby creating progressively intricate representations of the data [43–46]. The mathematical representation of an MLP can be articulated through the following Eqs. 5–10:

$$Z1 = f(W1X + b1) \tag{5}$$

$$a1 = g(z1) \tag{6}$$

$$z2 = f(W2a1 + b2) \tag{7}$$

$$a2 = g(z2) \tag{8}$$

$$zk = f(Wk * ak - 1 + bk) \tag{9}$$

$$Y = g(zk) \tag{10}$$

In these equations, X denotes the input data, b and W signify the network's weights and biases, f and g represent the activation functions for the hidden and output layers, respectively, and Y indicates the network's output.

3.5. Gradient boosting machine (GB)

A gradient boosting machine (GBM) represents a robust ensemble machine-learning technique that integrates multiple weak learners, specifically decision trees, to form a more competent learner. This ensemble methodology enhances the model's stability and predictive accuracy by amalgamating various individual models. In each iteration, a new decision tree is incorporated into the ensemble, with a primary focus on minimizing the errors (residuals) produced by the preceding model. The ultimate prediction generated by a GBM is the cumulative result of the predictions from all individual trees within the ensemble as shown in Eq. 11 [47,48]:

$$F_n(x_t) = \sum_{i=0}^{n} f_i(x_t) \tag{11}$$

Here, x_t denotes the independent variable at each time step t, while fi (x_t) represents the weak learners or decision trees trained during each iteration.

The effectiveness and performance of the gradient boosting model are governed by two principal hyperparameters. The first is the number of estimators (n_estimators), which specifies the total number of decision trees that comprise the final GBM ensemble. Typically, an increase in the number of estimators enhances accuracy, particularly for intricate datasets. However, an excessive number of trees may lead to overfitting, where the model begins to memorize the training data rather than discerning general patterns, resulting in diminished performance on new, unseen data. The second parameter is the learning rate (η), which dictates the magnitude of the step taken when incorporating a new decision tree into the ensemble, thereby influencing the contribution of each tree to the overall prediction. Elevated learning rates facilitate quicker learning but may induce larger fluctuations in the loss function during training. Conversely, lower learning rates yield smaller, more cautious updates, potentially leading to a smoother training experience and improved generalization of the model's performance on unseen data. Additionally, other parameters, such as maximum depth (max_-depth), regulate the complexity of individual trees. A greater depth permits more intricate splits and enhances accuracy for complex datasets, yet it also heightens the risk of overfitting (with a typical range of 3-8). In contrast, a reduced depth limits complexity.

3.6. Xtreme gradient boosting (X.GB)

Xtreme Gradient Boosting (X.GB) was introduced as an advancement of the gradient boosting algorithm by Chen et al. [49]. This method represents an ensemble supervised machine learning approach that is both highly optimized and capable of parallel processing, distinguishing it from traditional gradient boosting. In each iteration of X.GB, the residuals from the previous predictor are utilized for calibration. X.GB exhibits several advantages over conventional gradient boosting, including more efficient tree partitioning, the generation of random hidden nodes, shorter leaf nodes, and the ability to perform out-of-core predictions. Additionally, X.GB incorporates a regularization term within its loss function to mitigate the risk of overfitting. The parallelization of the boosting process significantly reduces training time, allowing the algorithm to leverage multiple CPU cores. Consequently, X.GB is applicable in various engineering simulations, delivering rapid and dependable outcomes [48–50]. The primary benefits of the X.GB technique stem from its speed, scalability, and support for parallel

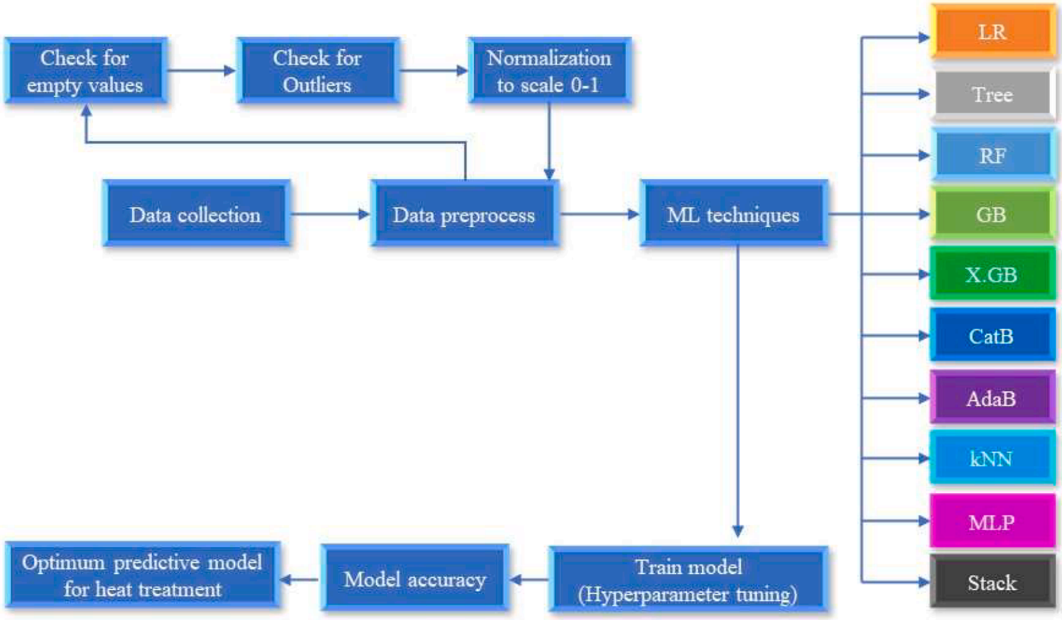


Figure 5. Workflow of Machine learning models development.

computing. In this boosting framework, the model is trained incrementally. For predicting the i^{th} data point during the t^{th} iteration, the function ft is added to minimize the following objective function [51] as shown in Eq. 12:

$$L^{(t)} = \sum_{i=1}^{n} \frac{l}{\left(y_i,\ y_i^{(i-1)} + f_t(x)_i\right) + \Omega(f_t)} \tag{12}$$

where Ω serves to penalize model complexity in X.GB, l denotes the loss function that quantifies the discrepancy between the predicted value (ŷi) and actual value (y_i), and xi represents the input vector.

3.7. CatBoost (CatB)

CatBoost (CatB) is an advanced gradient-boosting algorithm specifically engineered to effectively manage categorical data. A significant benefit of the CatB model is its capability to automatically process categorical variables without necessitating preprocessing steps. Furthermore, this algorithm is adept at handling datasets characterized by numerous categorical features. Utilizing decision trees as its foundational model, CatBoost employs the gradient descent technique to enhance predictive accuracy. The algorithm also integrates inherent regularization strategies to mitigate the risk of overfitting, which may lessen the requirement for extensive hyperparameter optimization. Like Gradient Boosting Machines (GBM), CatBoost provides a variety of hyperparameters, including the number of estimators, learning rate, and the depth of individual trees. Moreover, its built-in regularization mechanisms help manage model complexity and further reduce the likelihood of overfitting [52,53].

3.8. AdaBoost (AdaB)

AdaBoost represents a distinct boosting algorithm characterized by its unique methodology. The Adaptive Boosting Regression technique involves the training of a sequence of weak learners, specifically utilizing decision trees. Initially, all data points are assigned equal weights during the training of a weak learner. In subsequent iterations, these weights are modified in response to the accuracy of predictions made. This iterative process continues, culminating in a final prediction that is derived from a weighted aggregation of the outputs from each learner. Notably, AdaB prioritizes challenging cases during its iterations, thereby enhancing the overall efficacy of the model [54].

3.9. K-Nearest neighbors (kNN)

The k-nearest neighbor (kNN) algorithm is a non-parametric approach utilized for both classification and regression tasks. It estimates the value for a new data point by referencing the values of its k closest data points within the training dataset. This technique relies on "feature similarity" to project the values of prospective data points. The kNN model functions by calculating the distance between a new observation and all existing observations in the training set. The Euclidean distance is the most frequently employed metric, defined by Eq. 13 [55]:

$$d(x_i x_j) = \sqrt{\sum_{k=1}^{p} \left(x_{ik} - x_{jk}\right)^2} \tag{13}$$

In this equation, p represents the number of input features, x_{ik} indicates the value of the kth input feature for the i^{th} observation, and x_{jk} signifies the value of the k^{th} feature for the j^{th} observation.

After computing the distances, the kNN algorithm identifies the k neighbors with the smallest distances. The predicted value for the new observation is then determined by averaging (or taking the median of) the target variable values of these k nearest neighbors. A significant advantage of kNN regression lies in its straightforwardness and ease of interpretation. Nonetheless, it is essential to choose an appropriate value for k, as selecting values that are too small or too large can lead to overfitting or underfitting, respectively. Additionally, the efficacy of kNN regression may be compromised in high-dimensional datasets or when the data exhibits a complex structure.

3.10. Stack models

Stacking represents an ensemble learning methodology that integrates several machine learning models to enhance the overall accuracy and robustness of predictions. The term "stacking" derives from the process of layering individual models to construct a cohesive, singular model. The fundamental principle of stacking involves training a collection of base models on the initial training dataset, subsequently

Table 4
List of hyperparameters used for ML model development: HRC.

Model	Hyperparameter
LR	N/A
Tree	’min_samples_split’: 2, ’max_depth’: None
RF	’n_estimators’: 300, ’max_depth’: 20
GB	’n_estimators’: 300, ’max_depth’: 5, ’learning_rate’: 0.1
X.GB	’n_estimators’: 300, ’max_depth’: 5, ’learning_rate’: 0.1
CatB	’learning_rate’: 0.1, ’iterations’: 300, ’depth’: 7
AdaB	’n_estimators’: 200, ’learning_rate’: 1.0
KNN	’n_neighbors’: 3, ’metric’: Euclidean, ’weight’: by distances
MLP	’learning_rate_init’: 0.001, ’hidden_layer_sizes’: (100, 50), ’activation’: ’relu’
Stack	N/A

Table 5
List of hyperparameters used for ML model development: K_{1C}.

Model	Hyperparameter
LR	N/A
Tree	’min_samples_split’: 2, ’max_depth’: None
RF	’n_estimators’: 100, ’max_depth’: 20
GB	’n_estimators’: 300, ’max_depth’: 5, ’learning_rate’: 0.1
X.GB	’n_estimators’: 300, ’max_depth’: 5, ’learning_rate’: 0.1
CatB	’learning_rate’: 0.1, ’iterations’: 200, ’depth’: 7
AdaB	’n_estimators’: 200, ’learning_rate’: 0.1
KNN	’n_neighbors’: 3, ’metric’: Euclidean, ’weight’: by distances
MLP	’learning_rate_init’: 0.01, hidden_layer_sizes: (100), activation: ’tanh’
Stack	N/A

utilizing these models to generate predictions on a separate validation dataset. These predictions are then employed as input features for a higher-order model, referred to as the meta-model, which is specifically trained to yield the final predictions. This meta-model is designed to leverage the unique strengths and limitations of the base models, thereby facilitating more precise predictions [56,57].

A significant advantage of stacking lies in its capacity to elevate the performance of the individual base models by fostering inter-model learning. The meta-model effectively synthesizes the predictions from the base models, taking into account their diverse strengths and weaknesses. In summary, stacking is a formidable ensemble learning strategy that enhances the efficacy of machine learning models. It proves particularly beneficial in scenarios where the base models exhibit varying types of errors or strengths, enabling them to collaborate and generate more accurate outcomes [58,57].

3.11. Model training and evaluation

The characteristics outlined in Table 3 were chosen as the input variables for the models, while the dataset pertaining to hardness and fracture toughness served as the output for the predictive model. Figure 5 illustrates the comprehensive roadmap for developing predictive models using various machine learning models.

Various ML models, including LR, Tree, RF, GB, X.GB, CatB, AdaB, kNN, MLP, and Stack models, were employed to establish the relationship between the primary input variables and the outcomes of hardness and fracture toughness. The effectiveness of predictive models is heavily reliant on the selection of the appropriate model class and the tuning of hyperparameters within that class. To enhance model performance, it is essential to adjust the hyperparameters of each ML algorithm. There are several optimization methods available for hyperparameter searching, such as Grid Search, Random Search, and Bayesian optimization techniques [59,60]. In this research, hyperparameter tuning was conducted using Random Search Cross-Validation and Bayesian methods, facilitated by the Scikit-learn library. Additionally, the performance of models was evaluated to compare the efficacy of Random Search Cross-Validation and Bayesian techniques, aiming to identify the optimal hyperparameter settings for the current dataset. The specific hyperparameter configurations for the different machine learning models derived from the Random Search CV approach are presented in Tables 4 and 5.

To mitigate overfitting and enhance the generalizability of the models, several strategies were implemented. Initially, 5-fold cross-validation was employed to assess the stability of each model across various data subsets. This approach ensures that the model does not merely memorize the training data but is capable of generalizing to new, unseen instances. Additionally, models that are particularly susceptible to overfitting, such as decision trees and neural networks, underwent careful regularization through hyperparameter optimization (for instance, by adjusting tree depth, establishing minimum samples per leaf, and applying regularization as necessary). Furthermore, a comparative analysis of the R^2 values and error metrics (MAE, MAPE, MSE, RMSE) across training, testing, and cross-validation phases provided insights into potential overfitting. A significant gap between training and testing performance was identified as a warning sign for overfitting, prompting further adjustments or refinements to the model.

3.12. Performance criteria

The efficacy of the developed model was evaluated during both the training, testing, and validation stages through the application of widely recognized statistical metrics. These include the Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2). The MAE quantifies the absolute deviation between predicted and actual outcomes; the MSE, which measures the average squared deviation between predicted and actual outcomes; RMSE, the square root of MSE, offers an interpretable measure of spread in residuals, and the coefficient of determination (R^2), which reflects the proportion of variance in the dependent variable that is accounted for by the regression model. MAPE expresses the prediction error as a percentage of actual values, offering a scale-independent measure that facilitates model comparison across datasets. The MAE is expressed in the same units as the variable being predicted, which limits its utility in comparing the performance of different regression models across various datasets. Notably, MAE is robust against the influence of outliers, often referred to as extreme values. The MSE can take on values ranging from 0 to infinity, with lower values being preferable. The R^2 value, which ranges from 0 to 1, indicates the degree to which the model’s trends align with the actual data movements. Enhanced predictive accuracy is associated with lower MAE, MAPE, MSE, and RMSE values, while a higher R^2 value signifies that the model successfully captures the variability within the data. The formulas for these statistical criteria are as shown in Eq. 14-18:

$$R^2 = \sum_{i=1}^{n} (\widehat{y}_i - y_i)^2 \Big/ \sum_{i=1}^{n} (y_i - \overline{y}_i)^2 \tag{14}$$

$$MAE = \sum_{i=1}^{n} \frac{(\widehat{y}_i - y_i)}{n} \tag{15}$$

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (\widehat{y}_i - y_i)^2 \tag{16}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\widehat{y}_i - y_i)^2} \tag{17}$$

$$MAEP = \sum_{i=1}^{n} \frac{(\widehat{y}_i - y_i)}{n} \times 100 \tag{18}$$

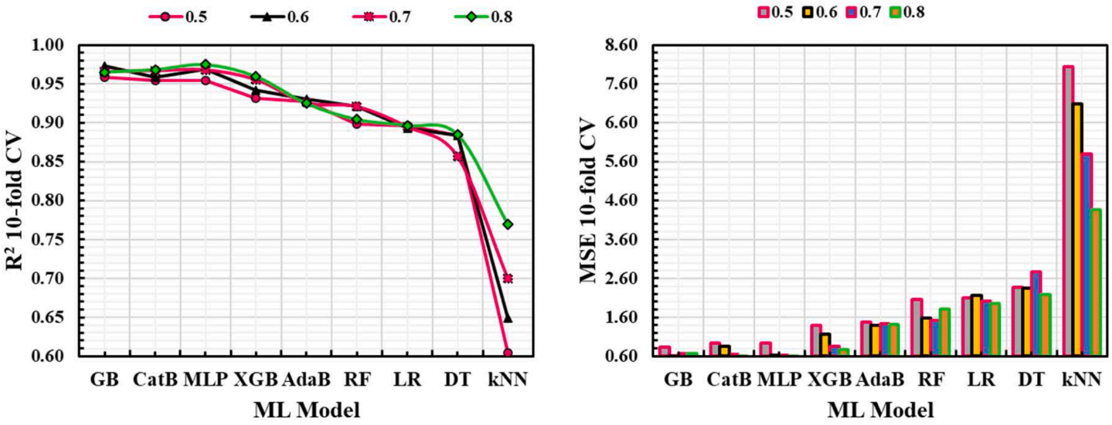


Figure 6. Model accuracy at different data split ratios for hardness (HRC).

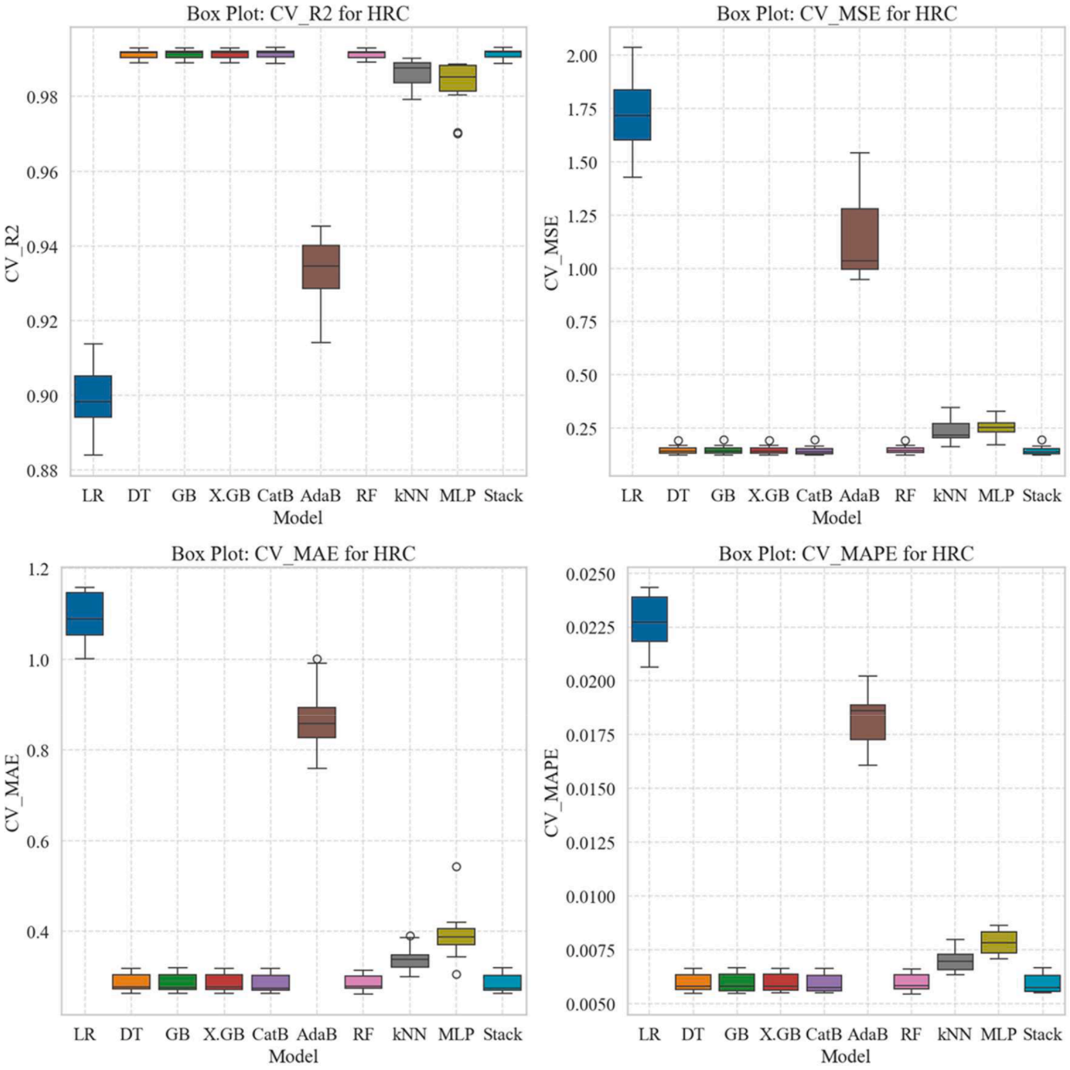


Figure 7. Model accuracy after 5-fold cross-validation for hardness (HRC): top: R^2, MSE, and bottom: MAE, MAPE.

4. Results and discussion

This section details the outcomes of predictive modeling of hardness and fracture toughness through machine-learning algorithms, addressing each output response variable individually. Initially, the impact of various data split ratios, specifically 0.5/0.5, 0.6/0.4, 0.7/0.3, and 0.8/0.2, has been investigated to determine the optimal split ratio for predicting both HRC and K_{IC}.

The motivation for testing various data split ratios comes from the fact that an incorrect split can overfit the model or fail to represent the

Table 6
Summary of model performance metrics for hardness (HRC) after 5-fold CV.

Hyperparameter	Model	R^2	MAE	MSE	RMSE	MAPE	Training_Time (s)
Bayesian	Stack	0.992	0.284	0.146	0.381	0.006	9.705
	CatB	0.991	0.285	0.146	0.381	0.006	108.879
	X.GB	0.991	0.286	0.147	0.382	0.006	75.607
	GB	0.991	0.286	0.147	0.382	0.006	95.178
	DT	0.991	0.286	0.147	0.383	0.006	64.775
	RF	0.991	0.286	0.148	0.384	0.006	98.927
	kNN	0.986	0.341	0.236	0.482	0.007	50.656
	MLP	0.980	0.450	0.346	0.586	0.009	3.173
	AdaB	0.932	0.855	1.142	1.065	0.018	76.606
	LR	0.899	1.092	1.734	1.315	0.023	0.016
Random Search	Stack	0.992	0.285	0.145	0.380	0.006	18.465
	CatB	0.991	0.285	0.146	0.381	0.006	11.108
	X.GB	0.991	0.286	0.147	0.382	0.006	13.343
	GB	0.991	0.286	0.147	0.382	0.006	4.078
	DT	0.991	0.286	0.147	0.383	0.006	7.171
	RF	0.991	0.286	0.148	0.384	0.006	0.141
	kNN	0.986	0.341	0.236	0.482	0.007	52.031
	MLP	0.983	0.395	0.254	0.503	0.008	0.062
	AdaB	0.934	0.869	1.150	1.068	0.018	1.266
	LR	0.899	1.092	1.734	1.315	0.023	0.000

complexity of the underlying data distribution. The ideal split should include enough training data to support model learning while also retaining a representative sample for analyzing unseen data. As illustrated in Figure 6, the model accuracy for HRC across different data split ratios indicates that a ratio of 0.8 yielded optimum performance compared to the other ratios. This observation holds true for K_{IC} as well; consequently, a data split ratio of 0.8 was employed for the training and testing of the machine learning models for both HRC and K_{IC}. Furthermore, predictive models were developed, and an analysis was performed to examine any possible overfitting associated with the fine-tuning of their hyperparameters. Ultimately, a comparison will be carried out to assess the accuracy of the machine-learning algorithms, resulting in recommendations for the most suitable model for both target features. Lastly, the top-performing model was validated using a dataset that representative of real-world distributions, including cases outside the original training data range. This step was critical for determining the robustness and generalizability of the model when exposed to new, unseen data [61]. This dual evaluation methodology facilitates a comprehensive assessment of the model's capacity for generalization and its accuracy in relation to data that extends beyond the training set. Typically, models are regarded as highly predictive when the R-squared value surpasses 0.80, accompanied by minimized error metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE).

4.1. Performance comparison of different ML models

4.1.1. Hardness (HRC)

Figure 7 illustrates the R^2, MSE, MAE, and MAPE values for all models after 5-fold cross validation comparing the accuracy of the ten predictive models for HRC. The assessment of the machine learning model's performance was carried out using the data from the validation set.

The evaluation of different machine learning models for the prediction of HRC reveals several significant insights. Notably, the top three models following a 5-fold validation process were the Stack, CatB, and X. GB models, achieving R^2 values of 0.992, 0.991, and 0.991, respectively. Furthermore, the GB and RF models exhibited impressive performance, each achieving R^2 values of 0.991. In other words, these advanced ML techniques have achieved 99% accuracy for HRC. The Stack and CatB models have proven to be the top performers, exhibiting minimal variance among training, testing, and 5-fold validation R^2 values, thereby indicating robust model generalization. These models are particularly effective in delivering accurate predictions, especially when working

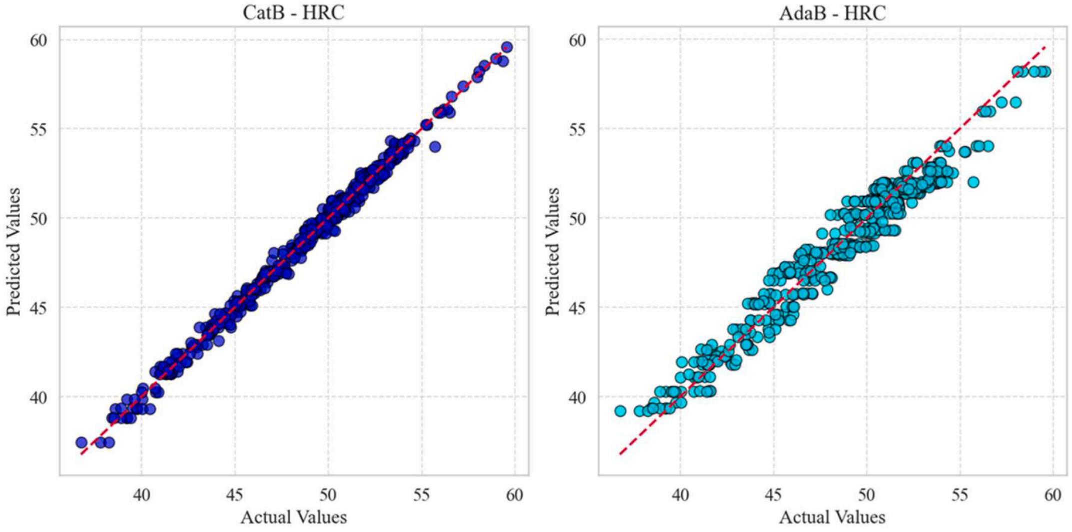


Figure 8. Hardness values (HRC) for experimental and predicted results for CatBoost and AdaBoost.

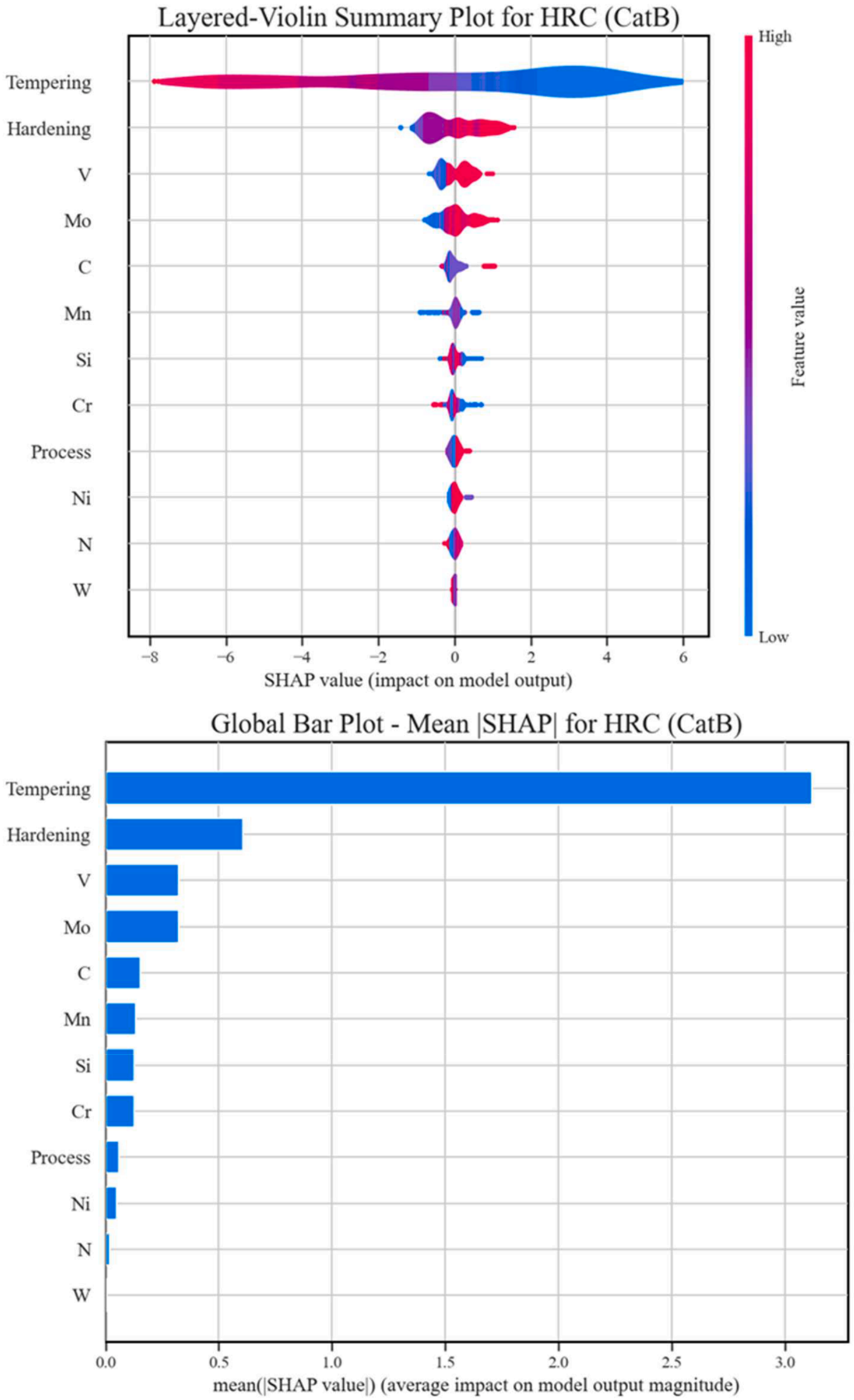


Figure 9. The importance of each feature to the HRC based on the SHAP (Shapley additive explanations) analysis: a. SHAP summary-top, b. mean SHAP-bottom.

with large datasets that feature complex, nonlinear relationships among multiple variables. The CatB model, in particular, is advantageous for datasets with diverse scales and can effectively accommodate both categorical and numerical features, making it more robust to outliers and noise than other models [57,62]. In contrast, the LR and AdaB models exhibited moderate performance with lower R^2 values, indicating their limitations in effectively capturing the complex patterns present in the data.

In terms of error metrics, the MSE, MAE, and MAPE values clearly highlight the differences in model performance. For example, the three models Stack, CatB, and X.GB demonstrate low MSE values of 0.145, 0.146, and 0.147 HRC, respectively and MAE values of 0.380, 0.381, and 0.382, respectively, indicating their strong predictive accuracy. The MAPE for the top 5 best-performing models is 0.006 (6%). In comparison, models like AdaBoost and Linear Regression exhibit greater error metrics, with MSE values of 1.1 and 1.7 HRC, respectively, and MAE values of 0.9 and 1.1 HRC, respectively. The MAPE for these underperforming models is 0.02 (20%). These findings suggest that the ensemble methods CatB and Stack stand out as the most effective models for predicting HRC, show exceptional performance and ability to generalize [57,62]. The high R^2 values and low error metrics across training, testing, and cross-validation datasets further emphasize their

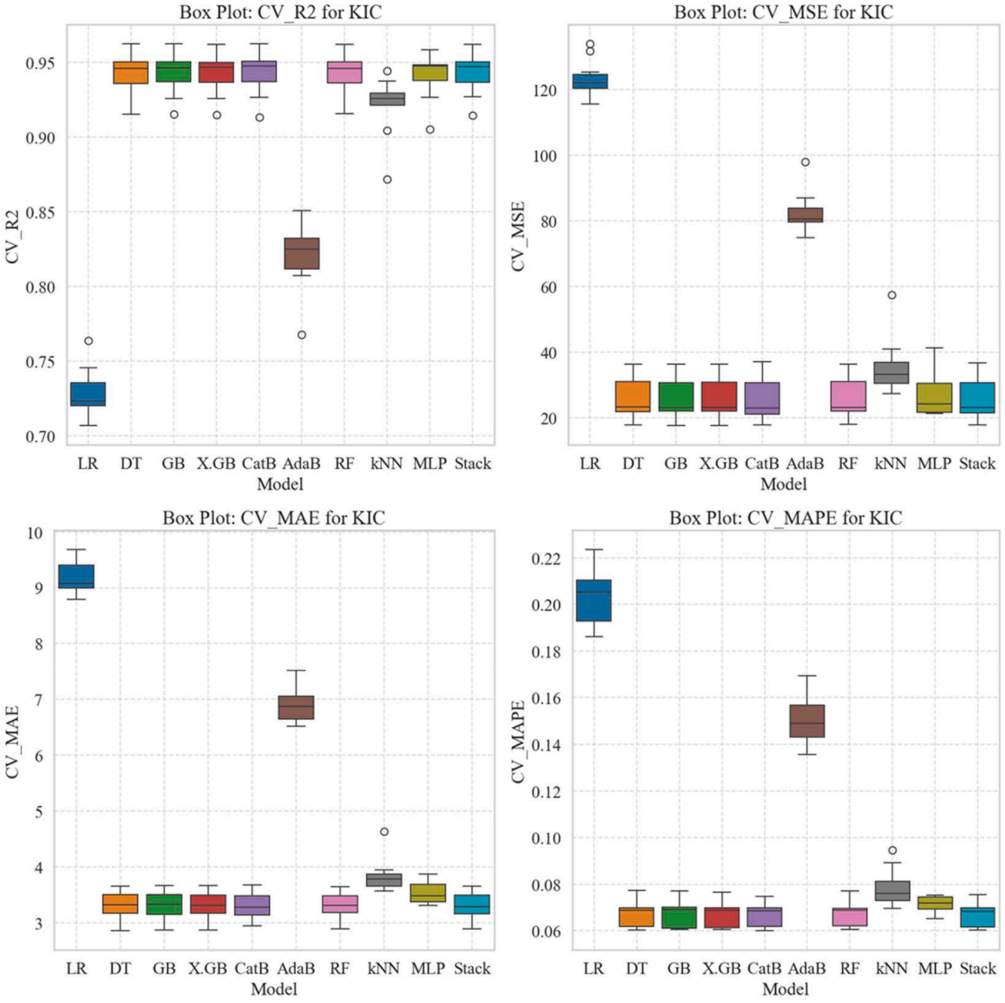


Figure 10. Model accuracy after 5-fold cross-validation for fracture toughness (K_{IC}): top: R^2, MSE, and bottom: MAE, MAPE.

effectiveness in predictive modeling. On the other hand, simpler models like linear regression and AdaB, while they deliver acceptable results, do not measure up to the advanced algorithms needed for enhanced predictive accuracy. Moreover, Table 6 presents a summary of the error metrics derived from both Bayesian and Random Search CV, along with their respective training times. This indicates that Random Search CV is less computationally demanding while yielding optimal hyper-parameters. Furthermore, advanced machine learning techniques, such as ensemble methods, require extended training durations.

The experimental and the predicted HRC for the top performing CatB model and the low performed AdaB model are shown in Figure 8. Based on the plots, CatB model showing a strong correlation between predicted and actual values with minimal scatter, suggesting high accuracy and low prediction error. AdaB model, while showing a positive correlation, exhibit more variability in their predictions. The CatB model's impressive predictive capabilities, marked by high R^2 values and minimal error metrics, underscore its proficiency in capturing the complex relationships inherent in hot-work tool steel designs. The analysis of actual and predicted values of HRC and their tight clustering around the regression line further confirms the model's strength and adaptability, establishing it as an essential tool for optimizing material properties while minimizing the need for extensive experimental trials.

Another critical aspect of this study was to understand which features contributed the most to predicting the hardness. Machine learning models, sometimes known as black-box models, provide accurate predictions for regression tasks but are uninterpretable. To address this and understand the ML prediction, further SHAP analysis was used in this study to provide a global interpretation of feature importance through the top-performing CatB model. SHAP explain the ML model prediction based on game theory. For instance, inputs variables are taken as players, and output variables or predictions are referred to as payout. The contribution of each player in the game can be calculated with the help of SHAP [52].

Figure 9 illustrates the impact of each feature, with SHAP analysis considering the SHAP summary and mean values. On the summary (Figure 9**a**) the features are ordered by their effect on prediction, but we can also see how higher and lower values of the feature will affect the results. The horizontal axis represents the SHAP value, while the color represents the magnitude of the feature (low or high), when compared to other observation. In this case, higher tempering temperature has a negative impact on the prediction HRC, while lower values have a positive impact. The mean SHAP in Figure 9**b**, orders the features from the highest to the lowest effect on the prediction. It considers the absolute SHAP value, thus whether the feature has a positive or negative impact on the prediction is ignored. Considering the mean SHAP, tempering temperature exerts the most substantial influence on HRC, followed by hardening temperature and the contents of V, Mo, C, Mn, Si, Cr, Process, Ni, N, and W. In SHAP analysis, input parameter significance is determined by their contribution to predictive outcomes based on the dataset's range and variability. In this case, the dataset predominantly represents a narrow carbon range, which may cause the effect of carbon appear less substantial compared to other elements with broader variability or strong correlations within the dataset. Nevertheless, the justification for influence of input parameters on HRC is consistent with

Table 7
Summary of model performance metrics for fracture toughness (K_{IC}) after 5-fold CV.

Hyperparameter	Model	R^2	MAE	MSE	RMSE	MAPE	Training_Time (s)
Bayesian	CatB	0.944	3.298	25.488	5.012	0.067	101.622
	Stack	0.944	3.297	25.518	5.017	0.067	9.729
	GB	0.943	3.305	25.642	5.029	0.067	91.255
	X.GB	0.943	3.304	25.651	5.030	0.067	76.503
	RF	0.943	3.304	25.722	5.038	0.067	98.600
	DT	0.943	3.307	25.705	5.036	0.067	60.937
	kNN	0.921	3.843	35.622	5.934	0.079	48.609
	MLP	0.910	4.534	40.075	6.306	0.092	3.298
	AdaB	0.821	6.950	81.799	9.038	0.150	66.468
	LR	0.728	9.182	123.256	11.099	0.204	0.000
Random Search	CatB	0.944	3.298	25.488	5.012	0.067	18.465
	Stack	0.944	3.299	25.544	5.019	0.067	11.108
	GB	0.943	3.305	25.642	5.029	0.067	13.343
	X.GB	0.943	3.304	25.651	5.030	0.067	4.078
	RF	0.943	3.305	25.652	5.031	0.067	7.171
	DT	0.943	3.307	25.705	5.036	0.067	0.141
	kNN	0.941	3.542	26.893	5.153	0.071	52.031
	MLP	0.921	3.843	35.622	5.934	0.079	0.062
	AdaB	0.821	6.907	82.153	9.057	0.150	1.266
	LR	0.728	9.182	123.256	11.099	0.204	0.000

fundamental material science principles and aligns with the given dataset conditions.

4.1.2. *Fracture toughness (K_{IC})*

Figure 10 presents the comparative performance of several machine learning models for K_{IC} prediction. The evaluation metrics employed include R^2, MSE, MAE, and MAPE with dual hyperparameter tuning applied to enhance model efficacy. The evaluation of machine learning model performance was conducted based on the data in the validation set. The analysis of the various ML models for predicting K_{IC} highlights several key findings. For instance, the top three models after undergoing 5-fold validation, were identified as the CatB, Stack, and GB models, yielding R^2 values of 0.944, 0.944, and 0.943, respectively (with 94% accuracy) and the MAPE for these models is 0.067 (7%). Furthermore, X. GB and RF also demonstrated commendable performance, with R^2 values of 0.943. Moreover, the CatB and Stack models emerged as the leading performers, exhibiting minimal variance among training, testing, and 5-fold validation R^2 values, thereby indicating robust model generalization. Conversely, LR, AdaB, and MLP show moderate performance with lower R^2 values, indicating their limitations in capturing complex patterns in the data.

In terms of error metrics, the MSE and MAE values further underscore the disparity in model performance. For instance, CatB and Stack demonstrate low MSE (25.5 MPa.m$^{1/2}$) and MAE (approximately 3.3 MPa.m$^{1/2}$) following 5-fold cross-validation, emphasizing their high prediction accuracy. In contrast, LR, AdaB, and MLP exhibits higher error values of MSE (123.3, 82.1, 35.6 MPa.m$^{1/2}$, respectively) alongside consistent MAE values of 5.0 MPa.m$^{1/2}$. Additionally, for the K_{IC} metric, the MAPE for the top five performing models is 0.07 (7%), while the underperforming models, including MLP, AdaB, and LR, present MAPE values of 0.08 (8%), 0.15 (15%), and 0.02 (20%), respectively. These results suggest that the ensemble techniques CatB and Stack emerge as the most promising models for K_{IC} prediction, given their superior performance and generalization capabilities. The high R^2 values and low error metrics across training, testing, and cross-validation sets underscore their efficacy in predictive modelling tasks. Conversely other models like LR, AdaB, and MLP, while providing reasonable performance, fall short in comparison, highlighting the need for advanced algorithms to achieve higher predictive accuracy. Table 7 presents a comprehensive overview of the error metrics derived from both Bayesian and Random Search Cross-Validation, along with their respective training times. The findings suggest that Random Search is less computationally demanding while yielding optimal hyperparameters. Additionally, the advanced ML techniques such as ensemble

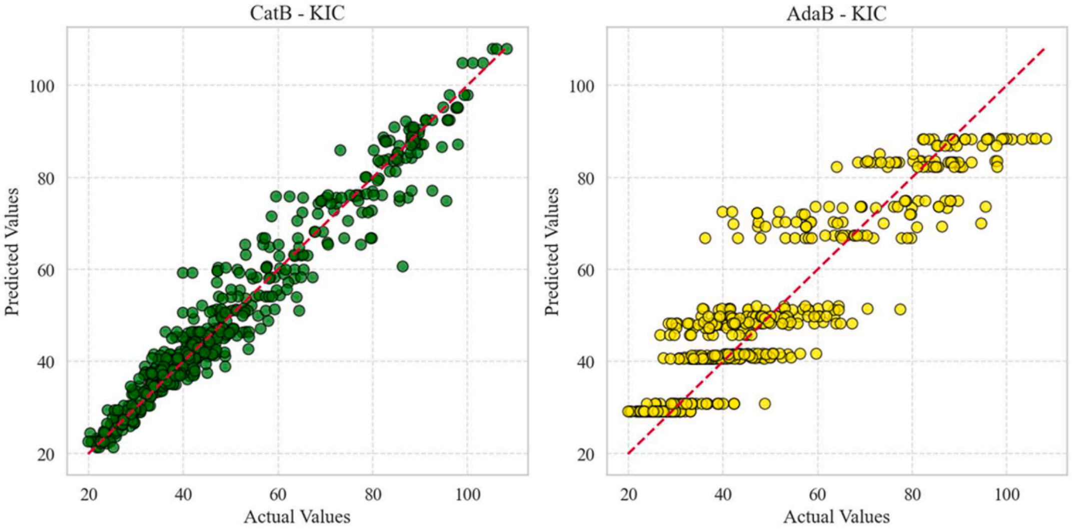


Figure 11. Fracture toughness values (K_{IC}) for experimental and predicted results for CatBoost, and AdaBoost.

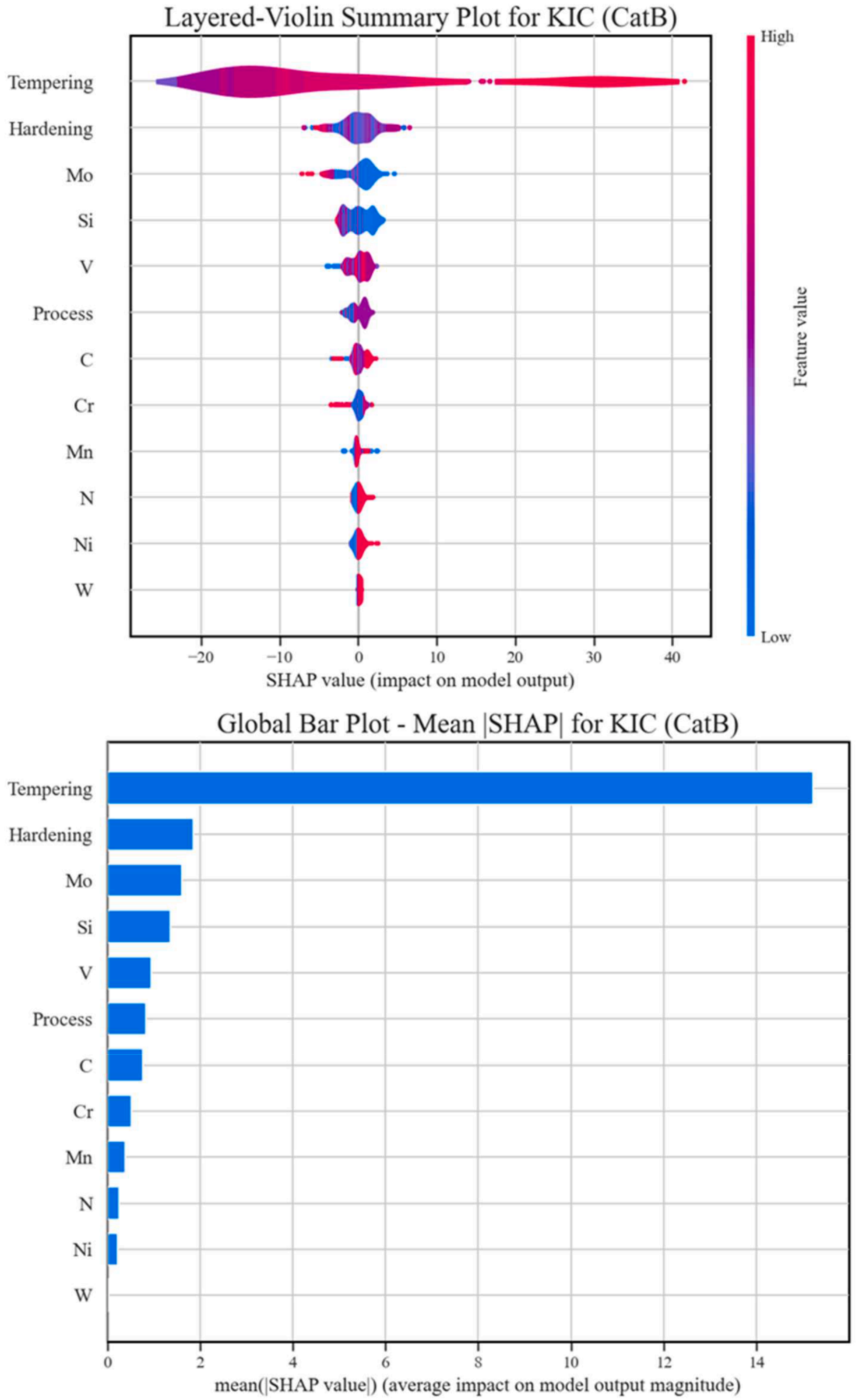


Figure 12. The importance of each feature to the K_{IC} based on the SHAP (Shapley additive explanations) analysis. a. SHAP summary-top, b. mean SHAP-bottom.

methods require longer training durations.

In the case of K_{IC}, the experimental and the predicted values are shown in Figure 11. These plots demonstrates that the ensemble models CatB show minimal scatter between actual and predicted values than simpler models like AdaB. This suggests that the advanced ensemble methods might be better at capturing non-linear and complex interactions for both HRC and K_{IC}.

To understand the ML K_{IC} prediction, SHAP analysis was used to provide a global interpretation of feature importance for the top-performing CatB model. Figure 12 illustrates the impact of each feature, with SHAP analysis considering the SHAP summary and mean indicators for K_{IC}. On the SHAP summary (Figure 12**a**) the features are ordered by their effect on prediction, in this case, higher tempering temperature has a positive impact on the prediction K_{IC}, while lower values have a negative impact. The mean SHAP in Figure 12**b**, shows that the tempering temperature exerts the most substantial influence on K_{IC}, followed by hardening temperature and the contents of Mo, Si, V, Process, C, Cr, Mn, N, Ni, and W. As discussed earlier in the context of HRC, SHAP analysis significance is influenced by dataset range and variability. The same principle applies to K_{IC} prediction, where the narrow range of C or other elements may constrain its observed effect.

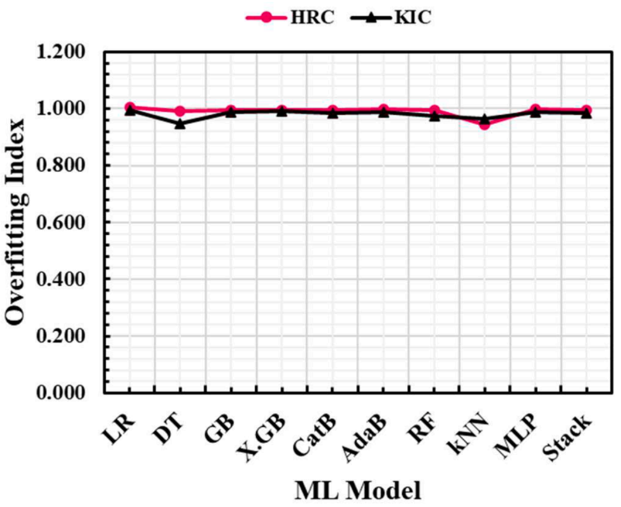


Figure 13. Overfitting index values for the machine learning techniques used in the study.

4.2. Model overfitting

In machine learning, overfitting occurs when the model demonstrates high accuracy on the training set but exhibits deteriorating performance on the testing set. To assess this tendency towards overfitting, an overfitting index is employed, which serves as a metric for evaluating the performance of machine learning models. An index value approaching 1 indicates a more effective model, while significant overfitting is suggested when the index falls below 0.77 [63]. The overfitting index (OI) is defined as shown in Eq. 17:

$$\textit{Overfitting index} = R_{test}/R_{train}$$

R_{test} and R_{train} represent the coefficient of determination for the test set and the training set, respectively. Figure 13 illustrates the overfitting index of machine learning models concerning the HRC and K_{IC}. The graph indicates that the indices for both HRC and K_{IC} are approximately equal to 1, significantly exceeding the threshold limit of 0.77. This observation suggests a minimal likelihood of overfitting in the machine learning models, which is essential for accurate property prediction and the design of new materials.

4.3. Model validation with real-world data

In this study, when the top-performing model (CatB) was evaluated with data beyond the range of the training dataset, one important finding appeared. The model was initially trained on data with carbon concentration of 0.35% to 0.38%. When attempting to forecast the HRC and K_{IC} of a sample containing 0.53% carbon, the model performance was poor, with an MAE of 5.0 HRC, 25.4 MPa.m$^{1/2}$, and R^2 values of -0.33 and -0.59 for HRC and K_{IC}, respectively. However, when a small portion of training data was supplemented with samples containing 0.53% carbon, the model performance improved drastically, with an MAE of 0.84 HRC, 3.1 MPa.m$^{1/2}$, and an R^2 value of 0.94 and 0.95 for HRC and K_{IC}, respectively, for unseen data. To highlight the strong predictive performance of the CatB model, a comparison of residuals with and without the 0.53% data was performed, as shown in Figure 14.

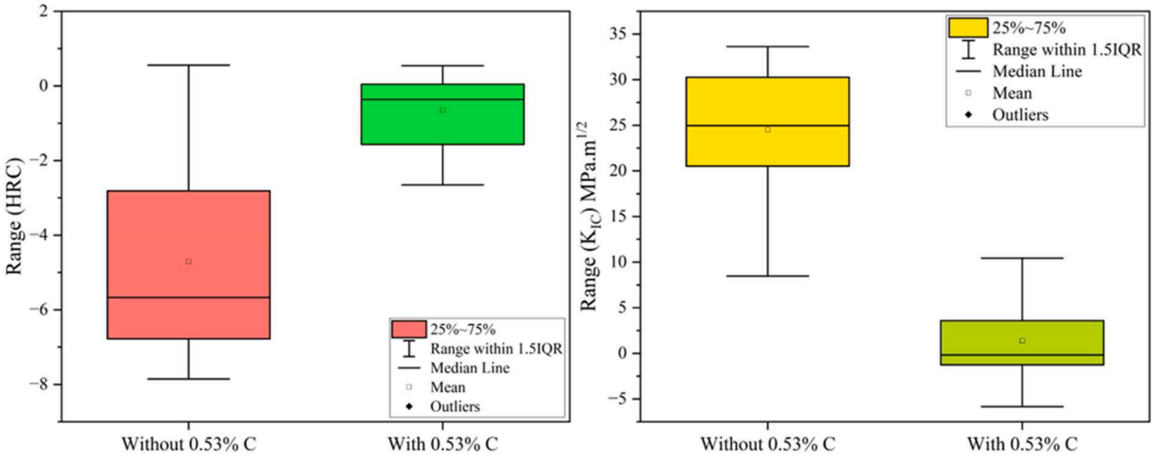


Figure 14. Model prediction results (residual distribution) for HRC and K_{IC}: CatB Model.

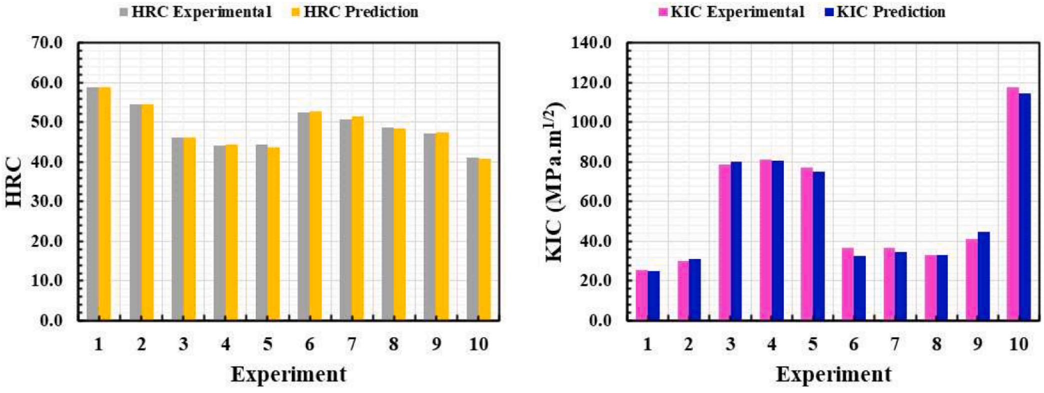


Figure 15. The GUI predictions (actual vs. prediction) of HRC and K_{IC} for the hot-work tool steels.

The figure shows that the projected HRC and K_{IC} of the CatB model are closely distributed around the zero, with just a small amount of data exhibiting bias, demonstrating the benefits of data range. This finding emphasizes an important aspect of machine learning in material science: data diversity and representativeness are key to building reliable predictive models.

4.4. Graphical user interface (GUI)

The existing models offer a useful means for predicting hardness and fracture toughness; however, their manual implementation is time-intensive and susceptible to inaccuracies due to their intricate nature. To tackle these issues, this study presents a user-friendly graphical user interface (GUI) model that consolidates data entry and model calculations into one interactive platform. This advancement removes the necessity for manual computations, greatly saving time and effort while improving both accuracy and ease of use [64]. Users can enter parameters and obtain predictions instantly, making this tool accessible for engineers and researchers without the need for advanced math or programming skills. The graphical user interface (GUI) was created using the Streamlit library in Python, enabling an efficient computation process and delivering immediate prediction results for HRC and K_{IC}. The equations integrated into the interface are based on the CatB model, which demonstrated the most impressive performance metrics, achieving an R^2 value of 0.99 (99% accuracy) for HRC with a mean absolute error (MAE) of 0.27 HRC, and an R^2 value of 0.95 (95% accuracy) for K_{IC} with a MAE of 3.1 MPa.m$^{1/2}$. To validate the usability and efficacy of the GUI, we conducted tests with ten random experimental points from the dataset to predict the HRC and K_{IC} and found the application was functional, revealing a strong correlation between the predicted values and the experimental findings, as shown in Figure 15. The developed GUI access can be utilized here using the following link: https://machinelearningproject0.streamlit.app/. Additionally, the codes used for the model development can be accessed through the GitHub link provided in the data availability statement.

4.5. Limitation and future recommendations of this work

Although this study demonstrates the effectiveness and reliability of CatBoost model in predicting hardness and fracture toughness while identifying areas for improvement. The current dataset, though sufficient for initial model development, could be expanded with diverse experimental results to enhance generalizability of the model beyond the training dataset. Additionally, the focus on other material properties such as fatigue and tribological properties remain unexplored. To address these limitations, future research should prioritize expanding and standardizing datasets through new experimental studies conducted under consistent conditions to improve reliability and applicability. Incorporating additional mechanical properties, such as fatigue strength and tribological properties, would enable the development of a more comprehensive predictive framework for hot-work tool steels. By addressing these areas, future research can build on the strengths of this study, paving the way for more versatile, accurate, and practical tools for hot-work tool steel design and optimization.

5. Conclusions

In this work, ten ML models were trained, tested, and validated using 5-fold cross validation technique to predict the mechanical properties (HRC and K_{IC}) of hot-work tool steels using the experimental data collected from the IMT. The key findings of the study are summarized as follows:

- The CatBoost and Stacking models demonstrated superior performance compared to other machine learning algorithms in predicting HRC and K_{IC}. Specifically, these models attained an accuracy of 99% for HRC and 94% for K_{IC}, reflecting their high precision. Furthermore, they exhibited the lowest error metrics, with CatBoost and Stacking achieving MAE values of 0.3 for HRC and 3.3 MPa·m$^{1}/^{2}$ for K_{IC}.
- The performance of the models was notably influenced by the data split ratio, with a 0.8/0.2 split ratio being optimal across various models, demonstrating the importance of an appropriate data split for model reliability.
- The tempering temperature emerged as the primary factor influencing the predictions of both HRC and K_{IC}. Additional notable factors comprised hardening temperature and the concentrations of elements such as V, Mo, C, Mn, Si, Cr, Process, Ni, N, and W for HRC, while for K_{IC}, the significant elements included Mo, Si, V, Process, C, Cr, Mn, N, Ni, and W.
- The minimal difference observed between the R^2 values of training, testing, and 5-fold cross-validation for the top-performing models (CatBoost, Stack) signifies robust generalization abilities. This implies that these models are less susceptible to overfitting and yield precise predictions on unseen data.
- The initial failure of the models to predict HRC and K_{IC} for steel with a carbon content of 0.53% (outside the original training range of 0.35% to 0.38%) highlighted the challenges of extrapolating beyond the trained feature range. However, adding data with higher carbon content significantly improved model performance, resulting in reduced error metrics and improved R^2.
- The graphical user interface (GUI) developed using the optimized CatBoost model demonstrated exceptional performance, achieving an R^2 value of 0.99 (indicating 99% accuracy) for predicting hardness (HRC) with a mean absolute error (MAE) of 0.27 HRC, and an R^2 value of 0.95 (indicating 95% accuracy) with an MAE of 3.1 MPa·m$^{1}/^{2}$ for K_{IC} prediction, providing a reliable and user-friendly tool for predicting mechanical properties.

Code Availability Statement

The codes used for this study are available from the GitHub link at: https://github.com/venuyarasu/machine.learning.project0.

CRediT authorship contribution statement

Venu Yarasu: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Bojan Podgornik:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition, Data curation, Conceptualization.

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.

Acknowledgements

The authors acknowledge the technical and scientific staff assistance of Institute of Metals and Technology, Slovenia and the financial support from the Slovenian Research Agency (research core funding No. P2-0050).

Data availability

Data will be made available on request.

References

[1] C.Y. Chen, I. Chiang, Y.C. Kang, The effects of austenitization temperature on the microstructural characteristics and mechanical properties of Cr–Mo–V hot work tool steels with different nitrogen contents, J. Mater. Res. Technol. 30 (2024) 9115–9129, https://doi.org/10.1016/j.jmrt.2024.05.264.

[2] Q. Zhou, X. Wu, N. Shi, J. Li, N. Min, Microstructure evolution and kinetic analysis of DM hot-work die steels during tempering, Mater. Sci. Eng. A 528 (2011) 5696–5700, https://doi.org/10.1016/j.msea.2011.04.024.

[3] C. Ma, Z. Xia, Y. Guo, W. Liu, X. Zhao, Q. Li, W. Qi, Y. Zhong, Carbides refinement and mechanical properties improvement of H13 die steel by magnetic-controlled electroslag remelting, Journal of Materials Research and Technology 19 (2022) 3272–3286, https://doi.org/10.1016/j.jmrt.2022.06.090.

[4] Y.J. Won, Y.J. Kwon, J.S. You, S.S. Park, K.S. Cho, Role of W addition in reducing heat checking and enhancing the mechanical properties of hot work tool steel, J. Mater. Res. Technol. 24 (2023) 3413–3422, https://doi.org/10.1016/j.jmrt.2023.04.008.

[5] A.P. Oliveira, L.H.Q.R. Lima, B.C.A. Felipe, C. Bolfarini, R.T. Coelho, P. Gargarella, Effect of microstructure and defect formation on the bending properties of additive manufactured H13 tool steel, Journal of Materials Research and Technology 15 (2021) 3598–3609, https://doi.org/10.1016/j.jmrt.2021.10.011.

[6] L. Wu, S. Das, W. Gridin, S. Leuders, M. Kahlert, M. Vollmer, T. Niendorf, Hot Work Tool Steel Processed by Laser Powder Bed Fusion: A Review on Most Relevant Influencing Factors, Adv Eng Mater 23 (2021) 2100049, https://doi.org/10.1002/adem.202100049.

[7] B. Podgornik, I. Belič, V. Leskovšek, M. Godec, Tool Steel Heat Treatment Optimization Using Neural Network Modeling, Metall Mater Trans A 47 (2016) 5650–5659, https://doi.org/10.1007/s11661-016-3723-0.

[8] A. Eser, C. Broeckmann, C. Simsir, Multiscale modeling of tempering of AISI H13 hot-work tool steel – Part 1: Prediction of microstructure evolution and coupling with mechanical properties, Computational Materials Science 113 (2016) 280–291, https://doi.org/10.1016/j.commatsci.2015.11.020.

[9] J.Y. Li, Y.L. Chen, J.H. Huo, Mechanism of improvement on strength and toughness of H13 die steel by nitrogen, Materials Science and Engineering: A 640 (2015) 16–23, https://doi.org/10.1016/j.msea.2015.05.006.

[10] S. Kheirandish, A. Noorian, Effect of niobium on microstructure of cast AISI H13 hot work tool steel, J. Iron Steel Res. Int. 15 (2008) 61–66, https://doi.org/10.1016/S1006-706X(08)60145-4.

[11] R. Casati, M. Coduri, N. Lecis, C. Andrianopoli, M. Vedani, Microstructure and mechanical behavior of hot-work tool steels processed by Selective Laser Melting, Materials Characterization 137 (2018) 50–57, https://doi.org/10.1016/j.matchar.2018.01.015.

[12] R. Besler, M. Bauer, K.P. Furlan, A.N. Klein, R. Janssen, Effect of Processing Route on the Microstructure and Mechanical Properties of Hot Work Tool Steel, Mat. Res 20 (2017) 1518–1524, https://doi.org/10.1590/1980-5373-mr-2016-0726.

[13] S. Mayer, C. Scheu, H. Leitner, I. Siller, H. Clemens, Correlation between heat treatment, microstructure and mechanical properties of a hot-work tool steel, International Journal of Materials Research 100 (2009) 86–91, https://doi.org/10.3139/146.101782.

[14] B. Skela, M. Sedlaček, F. Kafexhiu, B. Podgornik, Wear behaviour and correlations to the microstructural characteristics of heat treated hot work tool steel, Wear 426–427 (2019) 1118–1128, https://doi.org/10.1016/j.wear.2018.12.032.

[15] V. Leskovšek, B. Šuštaršič, G. Jutriša, The influence of austenitizing and tempering temperature on the hardness and fracture toughness of hot-worked H11 tool steel, Journal of Materials Processing Technology 178 (2006) 328–334, https://doi.org/10.1016/j.jmatprotec.2006.04.016.

[16] G. Telasang, J.Dutta Majumdar, G. Padmanabham, I. Manna, Structure–property correlation in laser surface treated AISI H13 tool steel for improved mechanical properties, Materials Science and Engineering: A 599 (2014) 255–267, https://doi.org/10.1016/j.msea.2014.01.083.

[17] J. Wróbel, A. Kulawik, A. Bokota, The Numerical Analysis of the Hardening Phenomena of the Hot-work Tool Steel, Procedia Engineering 177 (2017) 33–40, https://doi.org/10.1016/j.proeng.2017.02.179.

[18] F. Kara, N. Bulan, M.A. Akgün, U.K. Köklü, Multi-Objective Optimization of Process Parameters in Milling of 17-4 PH Stainless Steel using Taguchi-based Gray Relational Analysis, Engineered Science 26 (2023) 961, https://doi.org/10.30919/es961.

[19] I.V. Manoj, H. Soni, S. Narendranath, P.M. Mashinini, F. Kara, Examination of Machining Parameters and Prediction of Cutting Velocity and Surface Roughness Using RSM and ANN Using WEDM of Altemp HX, Advances in Materials Science and Engineering 2022 (2022) 1–9, https://doi.org/10.1155/2022/5192981.

[20] Z.L. Wang, T. Ogawa, Y. Adachi, Properties-to-microstructure-to-processing Inverse Analysis for Steels via Machine Learning, ISIJ Int 59 (2019) 1691–1694, https://doi.org/10.2355/isijinternational.ISIJINT-2019-089.

[21] M. Danish, M.K. Gupta, S.A. Irfan, S.M. Ghazali, M.F. Rathore, G.M. Krolczyk, A. Alsaady, Machine learning models for prediction and classification of tool wear in sustainable milling of additively manufactured 316 stainless steel, Results in Engineering 22 (2024) 102015, https://doi.org/10.1016/j.rineng.2024.102015.

[22] T.X. Yang, P. Dou, Prediction of hardness or yield strength for ODS steels based on machine learning, Materials Characterization 211 (2024) 113886, https://doi.org/10.1016/j.matchar.2024.113886.

[23] A. Cetin, G. Atali, C. Erden, S.S. Ozkan, Assessing the performance of state-of-the-art machine learning algorithms for predicting electro-erosion wear in cryogenic treated electrodes of mold steels, Advanced Engineering Informatics 61 (2024) 102468, https://doi.org/10.1016/j.aei.2024.102468.

[24] J. Xiong, T. Zhang, S. Shi, Machine learning of mechanical properties of steels, Sci. China Technol. Sci. 63 (2020) 1247–1255, https://doi.org/10.1007/s11431-020-1599-5.

[25] Y. Gui, K. Aoyagi, H. Bian, A. Chiba, Machine-Learning-Assisted Development of Carbon Steel With Superior Strength and Ductility Manufactured by Electron Beam Powder Bed Fusion, Metall Mater Trans A 55 (2024) 320–334, https://doi.org/10.1007/s11661-023-07251-1.

[26] F. Kazemi, A. Özyüksel Çiftçioğlu, T. Shafighfard, N. Asgarkhani, R. Jankowski, RAGN-R: A multi-subject ensemble machine-learning method for estimating mechanical properties of advanced structural materials, Computers & Structures 308 (2025) 107657, https://doi.org/10.1016/j.compstruc.2025.107657.

[27] A. Özyüksel Çiftçioğlu, F. Kazemi, T. Shafighfard, Grey wolf optimizer integrated within boosting algorithm: Application in mechanical properties prediction of ultra high-performance concrete including carbon nanotubes, Applied Materials Today 42 (2025) 102601, https://doi.org/10.1016/j.apmt.2025.102601.

[28] T. Shafighfard, F. Kazemi, F. Bagherzadeh, M. Mieloszyk, D. Yoo, Chained machine learning model for predicting load capacity and ductility of steel fiber–reinforced concrete beams, Computer Aided Civil Eng 39 (2024) 3573–3594, https://doi.org/10.1111/mice.13164.

[29] T. Shafighfard, F. Kazemi, N. Asgarkhani, D.Y. Yoo, Machine-learning methods for estimating compressive strength of high-performance alkali-activated concrete, Engineering Applications of Artificial Intelligence 136 (2024) 109053, https://doi.org/10.1016/j.engappai.2024.109053.

[30] S. Oh, H. Ki, Deep learning model for predicting hardness distribution in laser heat treatment of AISI H13 tool steel, Applied Thermal Engineering 153 (2019) 583–595, https://doi.org/10.1016/j.applthermaleng.2019.01.050.

[31] N. Pillai, R. Karthikeyan, Prediction of ttt curves of cold working tool steels using support vector machine model, IOP Conf. Ser.: Mater. Sci. Eng. 346 (2018) 012067, https://doi.org/10.1088/1757-899X/346/1/012067.

[32] C.L. Mambuscay, C. Ortega-Portilla, J.F. Piamba, M.G. Forero, Predictive Modeling of Vickers Hardness Using Machine Learning Techniques on D2 Steel with Various Treatments, Materials 17 (2024) 2235, https://doi.org/10.3390/ma17102235.

[33] B. Podgornik, B. Žužek, V. Leskovšek, Experimental Evaluation of Tool Steel Fracture Toughness Using Circumferentially Notched and Precracked Tension Bar Specimen, Materials Performance and Characterization 3 (2014) 87–103, https://doi.org/10.1520/MPC20130045.

[34] International Organization for Standardization, ISO 6508-1:2016: Metallic materials — Rockwell hardness test — Part 1: Test method, ISO, Geneva, 2016.

[35] Shen Wei, Zhao Tingshi, Gao Daxing, Liu Dunkang, Li Poliang, Qui Xiaoyun, Fracture toughness measurement by cylindrical specimen with ring-shaped crack, Engineering Fracture Mechanics 16 (1982) 69–82, https://doi.org/10.1016/0013-7944(82)90036-4.

[36] E. Nas, S. Akincioğlu, Optimization of Cryogenic Treated Nickel-Based Superalloy in Terms of Electro-Erosion Processing Performance, Academic Platform Journal of Engineering and Science 7 (2019) 1, https://doi.org/10.21541/apjes.412042. –1.

[37] J. Hao, T.K. Ho, Machine Learning Made Easy: A Review of *Scikit-learn* Package in Python Programming Language, Journal of Educational and Behavioral Statistics 44 (2019) 348–361, https://doi.org/10.3102/1076998619832248.

[38] X. Su, X. Yan, C. Tsai, Linear regression, WIREs Computational Stats 4 (2012) 275–294, https://doi.org/10.1002/wics.1198.

[39] B. De Ville, Decision trees, WIREs Computational Stats 5 (2013) 448–455, https://doi.org/10.1002/wics.1278.

[40] A. Navada, A.N. Ansari, S. Patil, B.A. Sonkamble, Overview of use of decision tree algorithms in machine learning, in: 2011 IEEE Control and System Graduate Research Colloquium, IEEE, Shah Alam, Malaysia, 2011, pp. 37–42, https://doi.org/10.1109/ICSGRC.2011.5991826.

[41] V. Rodriguez-Galiano, M. Sanchez-Castillo, M. Chica-Olmo, M. Chica-Rivas, Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines, Ore Geology Reviews 71 (2015) 804–818, https://doi.org/10.1016/j.oregeorev.2015.01.001.

[42] G. N, P. Jain, A. Choudhury, P. Dutta, K. Kalita, P. Barsocchi, Random Forest Regression-Based Machine Learning Model for Accurate Estimation of Fluid Flow in Curved Pipes, Processes 9 (2021) 2095, https://doi.org/10.3390/pr9112095.

[43] O.I. Abiodun, A. Jantan, A.E. Omolara, K.V. Dada, N.A. Mohamed, H. Arshad, State-of-the-art in artificial neural network applications: A survey, Heliyon 4 (2018) e00938, https://doi.org/10.1016/j.heliyon.2018.e00938.

[44] H. Ramchoun, M. Amine, J. Idrissi, Y. Ghanou, M. Ettaouil, Multilayer Perceptron: Architecture Optimization and Training, IJIMAI 4 (2016) 26, https://doi.org/10.9781/ijimai.2016.415.

[45] S. Kahrobaee, S. Ghanei, M. Kashefi, Using an Artificial Neural Network for Nondestructive Evaluation of the Heat Treating Processes for D2 Tool Steels, J. of Materi Eng and Perform 28 (2019) 3001–3011, https://doi.org/10.1007/s11665-019-04057-4.

[46] H. Taud, J.F. Mas, Multilayer Perceptron (MLP), in: Geomatic Approaches for Modeling Land Change Scenarios, Springer International Publishing, Cham, 2018, pp. 451–455, https://doi.org/10.1007/978-3-319-60801-3_27.

[47] V.K. Ayyadevara, Gradient Boosting Machine. Pro Machine Learning Algorithms, Apress, Berkeley, CA, 2018, pp. 117–134, https://doi.org/10.1007/978-1-4842-3564-5_6.

[48] U. Singh, M. Rizwan, M. Alaraj, I. Alsaidan, A Machine Learning-Based Gradient Boosting Regression Approach for Wind Power Production Forecasting: A Step towards Smart Grid Environments, Energies 14 (2021) 5196, https://doi.org/10.3390/en14165196.

[49] T. Chen, C. Guestrin, XGBoost: A Scalable Tree Boosting System, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and

Data Mining, ACM, San Francisco California USA, 2016, pp. 785–794, https://doi.org/10.1145/2939672.2939785.

[50] W. Zhang, C. Wu, H. Zhong, Y. Li, L. Wang, Prediction of undrained shear strength using extreme gradient boosting and random forest based on Bayesian optimization, Geoscience Frontiers 12 (2021) 469–477, https://doi.org/10.1016/j.gsf.2020.03.007.

[51] D. Chakraborty, J. Mondal, H.B. Barua, A. Bhattacharjee, Computational Solar Energy- Ensemble Learning Methods for Prediction of Solar Power Generation based on Meteorological Parameters in Eastern India, (2023). https://doi.org/10.48550/ARXIV.2301.10159.

[52] A.V. Dorogush, V. Ershov, A. Gulin, CatBoost: gradient boosting with categorical features support, (2018). https://doi.org/10.48550/ARXIV.1810.11363.

[53] H.T. Thai, Machine learning for structural engineering: A state-of-the-art review, Structures 38 (2022) 448–491, https://doi.org/10.1016/j.istruc.2022.02.003.

[54] L. Wen, Y. Li, W. Zhao, W. Cao, H. Zhang, Predicting the deformation behaviour of concrete face rockfill dams by combining support vector machine and AdaBoost ensemble algorithm, Computers and Geotechnics 161 (2023) 105611, https://doi.org/10.1016/j.compgeo.2023.105611.

[55] A. Sumayli, Development of advanced machine learning models for optimization of methyl ester biofuel production from papaya oil: Gaussian process regression (GPR), multilayer perceptron (MLP), and K-nearest neighbor (KNN) regression models, Arabian Journal of Chemistry 16 (2023) 104833, https://doi.org/10.1016/j.arabjc.2023.104833.

[56] X. Li, H. Chen, L. Xu, Q. Mo, X. Du, G. Tang, Multi-model fusion stacking ensemble learning method for the prediction of berberine by FT-NIR spectroscopy, Infrared Physics & Technology 137 (2024) 105169, https://doi.org/10.1016/j.infrared.2024.105169.

[57] T. Shafighfard, F. Bagherzadeh, R.A. Rizi, D.Y. Yoo, Data-driven compressive strength prediction of steel fiber reinforced concrete (SFRC) subjected to elevated temperatures using stacked machine learning algorithms, Journal of Materials Research and Technology 21 (2022) 3777–3794, https://doi.org/10.1016/j.jmrt.2022.10.153.

[58] L. Qin, D. Lu, H. Zheng, C. Wang, W. Dong, A new stacking model method to solve an inverse flow and heat coupling problem for aero-engine turbine blades, Case Studies in Thermal Engineering 56 (2024) 104209, https://doi.org/10.1016/j.csite.2024.104209.

[59] M. Feurer, F. Hutter, Hyperparameter Optimization, in: Automated Machine Learning, Springer International Publishing, Cham, 2019, pp. 3–33, https://doi.org/10.1007/978-3-030-05318-5_1.

[60] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, A. Talwalkar, Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization, Journal of Machine Learning Research 18 (2018) 1–52.

[61] T.T. Cai, H. Namkoong, S. Yadlowsky, Diagnosing Model Performance Under Distribution Shift, (2023). https://doi.org/10.48550/ARXIV.2303.02011.

[62] L. Prokhorenkova, G. Gusev, A. Vorobev, A.V. Dorogush, A. Gulin, CatBoost: unbiased boosting with categorical features, in: Neural Information Processing Systems, Montréal, Canada, 2018, in: https://proceedings.neurips.cc/paper_files/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf.

[63] J.W. Lee, C. Park, B. Do Lee, J. Park, N.H. Goo, K.S. Sohn, A machine-learning-based alloy design platform that enables both forward and inverse predictions for thermo-mechanically controlled processed (TMCP) steel alloys, Sci Rep 11 (2021) 11012, https://doi.org/10.1038/s41598-021-90237-z.

[64] W.L. Martinez, Graphical user interfaces, WIREs Computational Stats 3 (2011) 119–133, https://doi.org/10.1002/wics.150.