

Research Article

Applying Bayesian Optimization for Machine Learning Models in Predicting the Surface Roughness in Single-Point Diamond Turning Polycarbonate

Van-Hai Nguyen ^{1,2}, Tien-Thinh Le ^{1,2}, Hoanh-Son Truong,³ Minh Vuong Le,⁴
Van-Luc Ngo,^{1,2} Anh Tuan Nguyen ⁵, and Huu Quang Nguyen ⁵

¹Faculty of Mechanical Engineering and Mechatronics, PHENIKAA University, Yen Nghia, Ha Dong, Hanoi 12116, Vietnam

²PHENIKAA Research and Technology Institute (PRATI), A&A Green Phoenix Group JSC, No. 167 Hoang Ngan, Trung Hoa, Cau Giay, Hanoi 11313, Vietnam

³School of Mechanical Engineering, Hanoi University of Science and Technology, Dai Co Viet Street 1, Hanoi, Vietnam

⁴Faculty of Engineering, Vietnam National University of Agriculture, Gia Lam, Hanoi 100000, Vietnam

⁵Faculty of Mechanical Engineering, University of Economics and Technology for Industries, 456 Minh Khai Str., Hai Ba Trung District, Hanoi 100000, Vietnam

Correspondence should be addressed to Van-Hai Nguyen; hai.nguyenvan1@phenikaa-uni.edu.vn and Tien-Thinh Le; thinh.letien@phenikaa-uni.edu.vn

Received 4 May 2021; Revised 31 May 2021; Accepted 11 June 2021; Published 18 June 2021

Academic Editor: Samuel Yousefi

Copyright © 2021 Van-Hai Nguyen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper deals with the prediction of surface roughness in manufacturing polycarbonate (PC) by applying Bayesian optimization for machine learning models. The input variables of ultraprecision turning—namely, feed rate, depth of cut, spindle speed, and vibration of the X-, Y-, and Z-axis—are the main factors affecting surface quality. In this research, six machine learning- (ML-) based models—artificial neural network (ANN), Cat Boost Regression (CAT), Support Vector Machine (SVR), Gradient Boosting Regression (GBR), Decision Tree Regression (DTR), and Extreme Gradient Boosting Regression (XGB)—were applied to predict the surface roughness (Ra). The predictive performance of the baseline models was quantitatively assessed through error metrics: root means square error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2). The overall results indicate that the XGB and CAT models predict Ra with the greatest accuracy. In improving baseline models such as XGB and CAT, the Bayesian optimization (BO) is next used to determine their best hyperparameters, and the results indicate that XGB is the best model according to the evaluation metrics. Results have shown that the performance of the models has been improved significantly with BO. For example, the values of RMSE and MAE of XGB have decreased from 0.0076 to 0.0047 and from 0.0063 to 0.0027, respectively, for the training dataset. Using the testing dataset, the values of RMSE and MAE of XGB have decreased from 0.4033 to 0.2512 and from 0.2845 to 0.2225, respectively. Moreover, the vibrations of the X, Y, and Z axes and feed rate are the most significant feature in predicting the results, which is in high accordance with the literature. We find that, in a specified value domain, the vibration of the axes has a greater influence on the surface quality than does the cutting condition.

1. Introduction

The elements used in optical applications are mostly made of glass, crystalline materials, polymers, or plastic materials. Properties such as degree of transparency, refractive index, and spectrum are the deciding factors for the choice of optical materials [1]. In recent years, polymer optics have

become increasingly widely used. They offer advantages over traditional glass materials in different applications, due to their lightweight, low cost, high impact resistance, and flexibility in complex geometric design, despite their lower optical quality in comparison with glass materials [2]. Recently, various methods have been developed to mass-manufacture polymers for optics application such as

compression molding of the polymer followed by multistep grinding to achieve the desired surface finish, with the workpiece being beveled to prevent surface damage from sharp edges, and high-pressure injection into a mold of the desired geometry. After the mold is filled, the liquid polymer is allowed to cool and solidify. The workpiece is then polished to the desired surface finish [3, 4]. In addition, a manufacturing process method improves flexibility and cost-effectiveness in the production of small batches in diamond turning. This is often a significant gain compared to glass optics, facilitating a great deal of freedom in optical style. A variety of polymers are used for optical applications, such as polystyrene, PMMA, polyurethane, and PC. In terms of mechanical properties, PC offers a range of benefits over its rivals, especially for lighting applications [5].

Recently, there has been a dramatic increase in the breadth of applications for ultraprecision machining using single-crystal diamond tools [6]—particularly in the fields of manufacturing optical or magnetic parts, such as magnetic discs, polygonal mirrors, copier drums, and different lens shapes. As a result, the technique has successfully been expanded to machining various soft materials, creating mirror-like surfaces with submicrometer geometrical precision using ultraprecision machining and a single-crystal diamond tool [7]. Previous research indicates that surface quality from diamond turning is influenced by many factors such as machine vibration, chatter, tool wear, tool geometry, built-up edge, chip striking, and lubrication. Indeed, the surface roughness of the diamond turning machine parts plays a crucial role [6, 7]. There are several difficulties when studying the surface roughness of diamond turning parts. One of the main obstacles is that machining is a nonlinear process, governed by several parameters such as tool geometry, the interaction between parts, cutting parameters, machine vibration, and material properties.

Determination of surface quality is an important step in the manufacturing of any machine. Designers of modern production systems and machines are always aiming to better control surface quality, using better computational means with new algorithms. This has led researchers to seek to better understand the prediction of quality [8]. Various mathematical models have been proposed in the literature for predicting surface roughness. For instance, in a series of works, Krolczyk et al. [9, 10] have constructed second-order polynomial prediction functions for predicting the surface roughness and tool life in the dry machining of duplex stainless steel. In such a mathematical model, the influence of different parameters, namely, cutting speed, feed, and depth of cut, has been revealed based on the Student's t -test (comparison of two mean values of populations with Gaussian distributions and homogeneous variances). Krolczyk et al. [10] have exposed that the feed rate was the main influencing factor on the surface roughness. In another study, Kuntoğlu et al. [11] have also employed response surface methodology to seek the optimum cutting conditions, analysis of vibration, and surface roughness under different cutting speeds, feed rates, and cutting edge angles in turning of AISI 5140 steel. It is interesting to notice that the model developed in [11] can predict both surface

roughness and vibration during the turning of AISI 5140 within an accepted range of error of 10%. They also found that the feed rate was the most affecting parameter in increasing the surface roughness, which is in accordance with Krolczyk et al. [10] in turning steel. For machining of alloys, Gupta et al. [12] have constructed an optimization procedure to estimate the machining responses of nickel-based superalloy. The optimization problem has been conducted using a combination of response surface methodology, particle swarm optimization, and teaching learning-based optimization techniques. Different parameters have been considered including cutting speed of machining, feed rate, and cutting tool angle, whereas the machining responses were cutting force input, the potential of tool wear, surface roughness, and the length of tool-chip contact.

Recent research has suggested that machine learning may accurately predict surface roughness in the turning process. Eser et al. [13] have estimated the surface roughness of AA6061 alloy in milling using artificial neural networks and response surface methodology. It should be noticed that in such a study, the impact of the cutting parameters on the prediction has been characterized by using variance analysis [13]. In another work, Elangovan et al. [8] built a Multiple Linear Regression (MLR) model to predict surface roughness on the basis of input parameters: feed rate, depth of cut, spindle speed, flank wear, and vibration signal. The artificial neural network (ANN) method has been applied to predict the roughness for different cutting parameters (cutting speed, depth of cut, and feed rate) [14–17]. Özgören et al. [18] have employed the ANN technique to predict the power and torque values obtained from a beta-type Stirling engine. The best ANN's architectures have been determined such as 5-13-9-1 and 5-13-7-1, respectively, by using Levenberg–Marquardt learning algorithm [18]. In recent years, besides ANN modeling, some new ML methods have been introduced: Support Vector Regression (SVR), Gradient Boosting Regression (GBR), Linear Regression (LR), and Random Forest Regression (RFR) [19–21]. Pimenov et al. [22] have tested different machine learning models such as random forest, standard multilayer perceptrons, regression trees, and radial-based functions, for the prediction of surface roughness deviations from face milling machining processes. It is worth noticing that, in [22], the final machine learning prediction model has been developed in an automatic real-time manner for the machining processes. In addition, comparative research into ML models has been carried out to find the best in output prediction. The results for the SVR, polynomial regression, and ANN models indicated that ANN performs best in predicting lifetime but also worst in predicting cutting force and Ra [21]. This shows that not only does the regression of various machine learning models yield different results, but also a specific model has a certain effect on one type of output.

Therefore, this research sets out to investigate the application of SVR, Cat Boost Regressor (CAT), XG Boosting Regressor (XGB), Decision Tree Regressor (DTR), GBR, and ANN on surface roughness prediction. The paper is organized as follows. Section 2.1 introduces the database used in this study, while Section 2.2 outlines the research

methodology. Section 2 gives a brief overview of all the machine learning models used in this work. Finally, Section 3 provides the results and discussion thereof.

2. Materials and Methods

2.1. Data Collection and Analysis. In this study, the results of surface roughness measurements for 35 experiment runs are harvested from the available literature (Bolat [23], published in open access mode). In Bolat [23], the PC sheet was cut into a workpiece whose diameter was 30 mm and thickness was 10 mm. All workpieces were numbered as shown in Figure 1(a). Then, the workpiece was mounted inside a fixture, and the material of the fixture was not affected by temperature changes (shown in Figure 1(b)). During the machining process, the mixture between Kerosene and air was used as cutting lubrication.

The monocrystalline diamond tool S95843 was mounted on a tool holder (Figure 1(c)). According to the tool numbering for monocrystalline of C0.5mLG, the cutting tool has properties such that the type of radius is controlled waviness tool, noise radius is 0.5 mm, the top rank angle is 0° , and front clearance angle is 10° . To implement the experiments, the Precitech Freeform 700 U four-axis diamond turning machine tool was used for 35 experiment runs with feed rate, spindle speed, and depth of cut being 1–12 $\mu\text{m}/\text{rev}$, 1000–2250 rpm, and 3–50 μm , respectively, as shown in Table 1.

The surface roughness prediction dataset consists of two subdatasets. These subdatasets were combined to make the prediction data. Table 1 shows all these 35 measurement results, with seven attributes. These attributes contain information about six independent variables—feed rate, cut depth, spindle speed, and respective vibrations of X-, Y-, and Z-axis—and a dependent variable: surface roughness.

To measure the roughness of the finishing workpiece, the white light interferometer method was used by the Zygo NewView 5000 device with a vertical resolution of 0.1–0.2 nm. The surface roughness of the finished workpiece was measured in terms of three different positions. The measured values of surface roughness shown in Table 1 are root mean squared of three measured results obtained from the interferometer measurements.

2.2. Methodology. The data used in this study have been extracted using exploratory data analysis (EDA). The resulting dataset has been standardized by the $\log(1+x)$ function to ensure a consistent scale and distribution for all variables. The input data are divided into training and testing datasets, respectively, representing 78% and 22%, corresponding to 27 trials for training and 8 trials for testing, respectively. The training of the models consisted of six regression methods: multilayer perceptron neural network (MLP-NN), SVR, CAT, XGB, DTR, and GBR. The testing dataset was used to validate the model. To assess the performance of the proposed model, various error metrics—RMSE, MAE, and R^2 —were employed.

Figure 2 presents the flowchart of the model used in this paper: dataset extraction, feature selection, and different combinations of the dataset made to predict surface roughness. The figure also shows how to fine-tune the model parameters by using the Bayesian optimization algorithm to seek the best parameter for each model and then to determine the best model to predict Ra. Finally, the results and observations are discussed in Section 3.

3. Machine Learning Algorithms

3.1. Artificial Neural Network (ANN). ANNs are complicated computational models inspired by biological neural networks, which are capable of regression, classification, and pattern recognition. There are different ANN-type algorithms, such as backpropagation neural networks [24], probabilistic neural networks [25], convolutional neural networks [26], time-recurrent neural networks [27], and long short-term memory networks [28]. With straightforward and original characteristics, the multilayer perceptron (MLP) model has been chosen in this study. MLP consists of three main layers, fully connected: the input layer, hidden layer, and output layer [29]. Thanks to its properties, MLP has been used to predict tool wear flank and surface roughness [30–32].

The main advantages of the ANN model are the capability to work with any type of input data (complete or incomplete) [33]; the information is stored on the entire network instead of a database and the capability of parallel computing, which helps to reduce the computational time. On the other hand, some drawbacks of this approach need to be mentioned such as the hardware dependence, which requires a lot of computer resources when the input data is large. In addition, the behavior and the duration of the network are hard to control, which requires a lot of trials.

3.2. Support Vector Machine (SVM). Support Vector Machine (SVM) theory was developed by Cortes and Vapnik [34], and a version of an SVM for regression was launched in 1997 [35]. Support Vector Machine consists of different branches in which Support Vector Regression (SVR) is an important application. Many ML algorithms follow the principle of empirical error minimization, while SVR follows the principle of structural risk minimization across a restricted range of learning patterns, so it can obtain better generalization [36]. SVR is a distinguished analytical tool. It is applicable because it uses linear improvement techniques to seek out optimal solutions to nonlinear prediction issues in relation to higher-dimensional features. Therefore, it has been widely used for forecasting in the fields of finance, agriculture, hydrology, the environment, etc., and especially, in mechanical machining [21, 44, 45]. Consequently, SVR is an appropriate model to predict Ra in ultraprecision machining PC for applications in optics.

In terms of advantages, SVM can work really well with high-dimensional input space and is relatively memory efficient. In terms of drawbacks, unlike ANN, SVM is not



FIGURE 1: Experiment setup reproduced with permission from Bolat [23]: (a) numbered PC workpieces, (b) fixture for the placement of PC workpieces, (c) mounted monocristalline diamond on the cutting tool (front view), and (d) mounted assembly fixture and workpiece on the Precitech Freeform 700 U four-axis machine tool.

suitable for large datasets; it does not perform well with any type of data (for example, data set with more noise).

3.3. Cat Boost Regression (CAT). Most popular implementations of gradient boosting use decision trees as base predictors. It is convenient to use decision trees for numerical features, but, in practice, many datasets embody categorical features that are vital for prediction. CAT is a novel gradient boosting technology, developed by Yandex. It is associated with improved implementation within the gradient boosting tree algorithmic framework. This framework relies on a symmetrical decision tree algorithmic rule with few parameters, support for categorical variables, and high accuracy [46]. CAT improves the accuracy of the algorithm and its generalizability [47]. It has been successfully applied in many fields such as weather forecasting, media popularity prediction, evapotranspiration, and biomass [48, 49]. It is for this reason that the model is applied here to predict the performance of ultraprecision machining.

The advantages of gradient boosting approaches are their predictive accuracy compared to other machine learning models. These types of approaches have a lot of flexibility,

which means that they can be optimized using different loss functions or several hyperparameter tuning options that make the function fit flexible. They can work directly with the input data, which means that no data preprocessing is required. In terms of drawbacks, gradient boosting approaches are usually computationally expensive. The minimization of errors in these approaches can cause overfitting, and the influence of parameters is quite heavy on the behavior of the approach.

3.4. Decision Tree Regression (DTR). Decision trees (for classification and regression) are classic ML algorithms. As a group, their learning ability is not outstanding, but they are well known for their generalizability and feature filtering. When used for regression tasks, they are called regression trees [50]. As the number of iterations increases, the model continues to learn. The training is stopped when triggered by hyperparameters such as the number of selected features, the maximum depth of the tree, and the minimum sample size of branches.

Compared to other algorithms that require data preprocessing, the decision tree algorithm requires less effort in

TABLE 1: Surface roughness measurements for experiment (reproduced with permission from Bolat [23]).

Trial number	Feed rate ($\mu\text{m}/\text{rev}$)	Depth of cut (μm)	Spindle speed (rpm)	VX (mgRMS)	VY (mgRMS)	VZ (mgRMS)	Ra (nm)	Type
1	5	40	2000	28.07	45.97	41.69	17.90	Training dataset
2	9	22.5	1000	27.90	52.76	37.75	15.0	
3	5	22.5	2000	27.86	44.58	38.87	16.50	
4	9	22.5	2000	29.90	61.70	46.74	33.70	
5	9	5	1500	26.97	43.16	37.54	19.70	
6	5	5	2000	25.35	49.91	35.04	9.40	
7	1	40	1000	28.74	44.8	39.13	10.10	
8	1	5	2000	25.17	45.21	32.67	2.70	
9	1	40	2000	26.54	56.23	35.81	9.50	
10	5	5	1500	27.09	52.08	36.90	12.0	
11	1	22.5	2000	26.45	52.67	36.93	6.40	
12	1	40	1500	28.75	52.60	35.59	9.20	
13	5	40	1000	32.77	66.69	38.66	12.80	
14	9	5	1000	26.06	54.75	35.74	11.50	
15	1	22.5	1000	25.88	48.86	33.23	8.30	
16	1	5	1000	28.14	54.58	37.88	4.70	
17	5	40	1500	26.01	50.85	40.10	13.60	
18	9	40	1000	25.64	48.62	33.21	25.90	
19	1	22.5	1500	25.25	44.57	33.09	4.30	
20	9	5	2000	24.65	50.64	34.67	22.10	
21	2	3	1250	6.90	28.61	59.82	47.17	
22	7	22	1000	18.10	28.63	59.83	46.42	
23	2	6	2250	6.30	28.76	60.88	49.09	
24	7	30	1750	23.0	37.26	78.91	54.13	
25	12	40	1250	52.10	28.59	59.90	46.48	
26	2	50	1000	5.80	28.92	60.0	46.70	
27	7	20	2250	14.40	28.91	59.77	46.93	
28	5	22.5	1000	26.72	52.41	37.35	12.20	Testing dataset
29	1	5	1500	25.53	46.66	33.04	3.70	
30	5	5	1000	28.91	58.81	46.34	10.20	
31	5	22.5	1500	25.15	46.91	33.72	8.50	
32	9	40	2000	36.22	74.97	45.09	46.80	
33	9	40	1500	25.61	48.27	33.20	21.60	
34	9	22.5	1500	28.301	59.02	46.32	26.40	
35	12	10	1750	47.70	28.16	58.97	47.48	

this process. A decision tree does not require normalization or scaling of data. The decision tree algorithm can also work with incomplete data. In terms of disadvantages, a small change in data requires a lot of changes in the structure of the decision tree, which can eventually cause instability. The computational time of the decision tree is often expensive, especially when training the model.

3.5. Gradient Boosting Regression (GBR). Gradient Boosted Trees (GBTs) are a set of DTs, whose results are a combination of predictions of base models. DT-based ensembles like GBT have often been used in regression and classification problems, as they perform well [51]. GBT is an iterative algorithm, which means that each tree can take account of the error in the previous one. The final result of the GBT is the mean of the predicted results from all trees.

3.6. Extreme Gradient Boosting Regression (XGB). XGB is the most popular ML algorithm, developed in 2015. Regardless of the data type, it is well known to provide better solutions

than other ML algorithms, because of its rapidity, efficiency, and scalability [52, 53]. It has been the focus of research in various fields [54–56]. In particular, in mechanical machining [52, 57, 58], XGB is a good choice to predict tool wear and surface roughness.

XGB is used for supervising learning problems, where we use the training data (with multiple features) x_i to predict a target variable y_i . Before we learn about trees specifically, let us first review the basic elements in supervised learning. Below are the analytical formulas for the regression math on which XGB is based [52, 59].

Let $D = \{(x_i, y_i)\}$ be a dataset composed of n samples and m features $|D| = n, x_i \in R^m, y_i \in R$. A number of k -additive functions $f_k(x)$ are used in tree ensemble models to evaluate the function $\varnothing(x_i)$. This function can be expressed as [52]

$$\hat{y}_i = \varnothing(x_i) = \sum_{k=1}^K f_k(x_i), \quad f_k \in F, \quad (1)$$

where k represents the number of trees, x_i is the i^{th} iteration of the training process, and f_k denotes the decision rules of

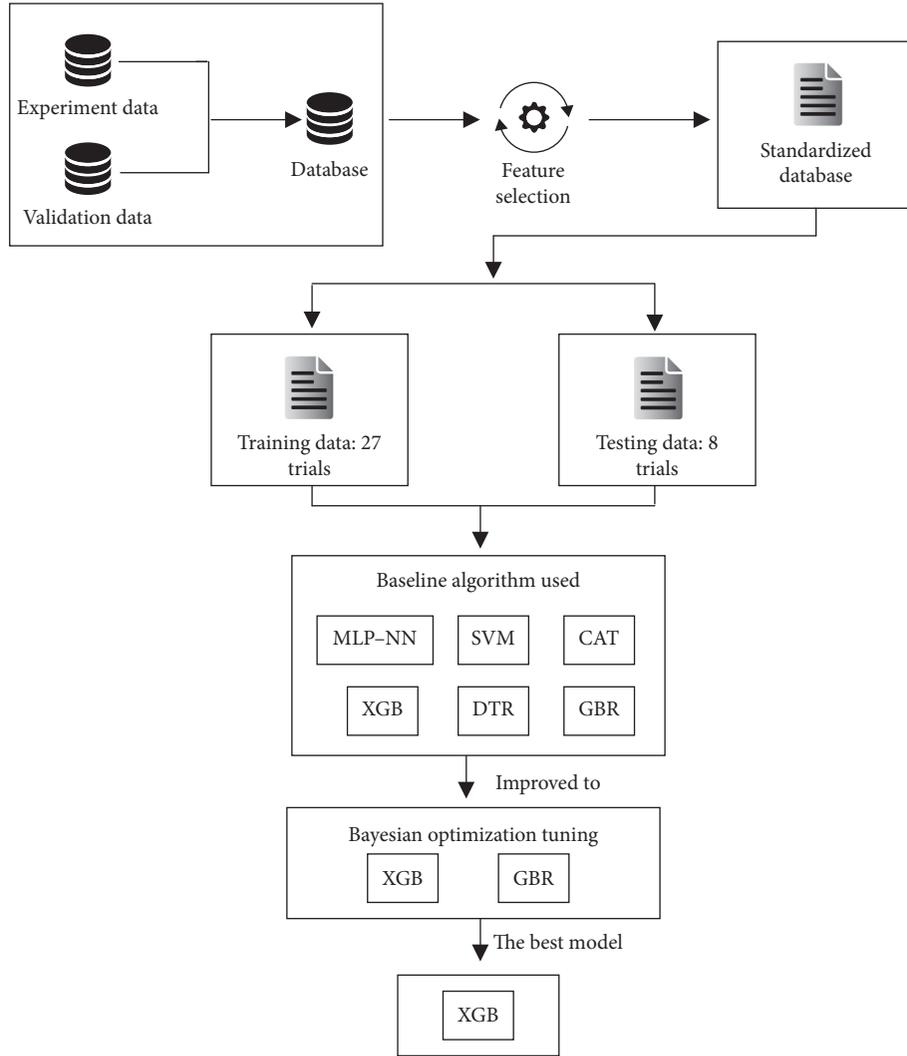


FIGURE 2: Overall methodology employed in this study.

the tree and weight of leaf score. The regression space F can be expressed as follows [52]:

$$F = \{f_x = w_{q(x)}\}, \quad q: R^m \longrightarrow T, \quad w \in R^T, \quad (2)$$

with q , T , and w representing the tree structure, the number of leaf nodes, and the corresponding weight, respectively. The errors of the model can be minimized using a regularized objective function, as shown as follows [52]:

$$L(\emptyset) = \sum_i \psi(y_i, F_K(x_i)) + \sum_i \Omega(f_k), \quad (3)$$

with ψ being a differentiable convex loss function, in terms of regression type, which can be the mean squared error function, while Ω is the regularization term that penalizes the complexity of the model to avoid overfitting, defined as [52]

$$\Omega(f_k) = \gamma T + 0.5\lambda w^2. \quad (4)$$

γ denotes the complexity, and λ is a constant coefficient. Since XGB is an additive algorithm, the prediction $F_K(x_i)$ of

the i^{th} instance at the k^{th} iteration can be expressed as follows [52]:

$$L(\emptyset) = \sum_i \psi(y_i, F_{K-1}(x_i) + f_k(x_i)) + \sum_i \Omega(f_k). \quad (5)$$

The main objective of XGB is to determine an additive function f_k that minimizes the objective function using the gradient descent optimization algorithm.

3.7. Bayesian Algorithm Optimization. In XGB and CAT, the main goal of hyperparameter optimization (i.e., tuning) is to minimize the objective function defined in equation (5). There are two popular hyperparameter optimization methods: random search (RS) and Bayesian optimization (BO). In random search, the hyperparameters are randomly chosen from the predefined search domain, and the searching is independent of the previous boosting result [60, 61]. The main advantage of RS is that it can be applied to high-dimensional problems. Bayesian optimization can be considered a probability approach, using probability

theories to optimize the hyperparameters [62]. In this work, we have chosen Bayesian optimization because of its performance, which has been demonstrated in previous studies in the literature [63, 64]. For simplicity's sake, in the rest of the paper, we use XGB_opt and CAT_opt to denote the XGB and CAT models built by using BO, respectively.

Before GBR and XGB training, we need to initialize the model's hyperparameters. However, the selection of network hyperparameters, based on experience or RS such as GridSearchCV and RandomizedSearchCV, currently requires a large number of attempts. On the other hand, to optimize the model's performance and reduce computational time, the hyperparameter screening process must be optimized—this is the main purpose of using BO. The Bayesian optimization framework utilizes historical data to optimize the search domain and constantly predict the posterior piece of information [65]. In particular, suppose that we have a functional relation between the hyperparameters and loss function:

$$p^* = \operatorname{argmin}_{p \in P} \operatorname{loss}(p), \quad (6)$$

where p is the set of all hyperparameters in which p is the set of hyperparameter combinations, p^* is the optimal parameter combination obtained from the final optimization, and $\operatorname{loss}(p)$ is the objective function.

In the proposed model, the hyperparameters are maximum tree depth (D), number of nodes in each tree (γ), number of trees (K), learning rate (η), regularization parameter (λ), and number of samples (N), as introduced in equations (1)–(4). The loss function is defined by the RMSE as [65]

$$\operatorname{loss}(p_j) = \sqrt{\frac{\sum_{i=1}^N (\hat{y}_i(p_j) - y_i)^2}{N}}, \quad (7)$$

where p_j is the j^{th} hyperparameter combination, y is the true value, and $\hat{y}_i(p_j)$ is the model output results obtained using the j^{th} hyperparameter combination p_j .

The next step of BO is to construct the dataset $D = \{(x_i, y_i)\}$, where x_i is the i^{th} set of hyperparameters and y_i is the corresponding error of the model output [65]:

$$y_i = \operatorname{loss}(p_i). \quad (8)$$

The posterior probability $P(y|x, D)$ is denoted as D . A Gaussian distribution is applied to the alternative model M whose mean and variance are denoted as μ and K , respectively. The specific functional expression M is obtained by fitting the dataset D [65]:

$$p(\operatorname{loss}) = G(\operatorname{loss}; \mu, K). \quad (9)$$

Based on M , the next observation is calculated using the acquisition function $\alpha(p)$ [65]:

$$p^* = \operatorname{arg max} \alpha(P, p(y|x)). \quad (10)$$

In the Bayesian decision theory, the capture function works by calculating the expected loss corresponding to the hyperparameter space p loss. In each iteration, the dataset D

is updated by receiving the parameters and losses from the previous one. The main characteristic of BO is model construction based on historical data to optimize the hyperparameters for each model [66].

3.8. Performance Assessment Criteria. Three statistical metrics have been used in this study to assess the performance of the proposed AI model in predicting surface roughness: root means square error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2). These metrics are defined as follows [67–70]:

$$\begin{aligned} \operatorname{RMSE} &= \sqrt{\frac{\sum_{i=1}^N (\hat{y}_i - y_i)^2}{N}}, \\ \operatorname{MAE} &= \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i|, \\ R^2 &= 1 - \frac{\sum_{i=1}^N (\hat{y}_i - y_i)^2}{\sum_{i=1}^N (\hat{y}_i - \bar{y})^2}, \end{aligned} \quad (11)$$

where y_i and \hat{y}_i are the measured and predicted values, respectively, and N is the total number of predicted data points. Higher values of R^2 represent the better performance of the model. On the other hand, the better performance of the model is reflected by lower values of RMSE and MAE.

When contrasting the values of metrics, we will prioritize the evaluation criteria selected in RMSE, because this is a more suitable method than MAE when the model error follows a normal distribution. Moreover, RMSE has a distinct advantage over MAE in that RMSE avoids using absolute value, which is highly undesirable in many mathematical calculations [71]. Therefore, if comparing the prediction accuracy of various regression models, then RMSE is a better choice as it is simple to calculate and differentiable. Moreover, a higher value of R^2 is considered desirable.

4. Results and Discussion

4.1. Prediction Accuracy of Various Baseline Models. The analyzed performance of MLP-NN, SVR, CAT, XGB, DTR, and GBT baseline regression models in terms of Ra prediction for diamond ultraturning is reported in this section. Table 2 shows the results of the various models acting on the training and testing datasets, sorted in ascending order of RMSE for the testing dataset. It can be seen that the predicted Ra varies considerably from one model to another. Using the testing dataset, XGB exhibits the best performance, in terms of all error metrics—this model yields the highest value of R^2 and the smallest values of RMSE and MAE. With the training dataset, XGB also exhibits similar performance, with the highest R^2 . The DTR model performs best in MAE with the training dataset and best in RMSE with the testing dataset. On the contrary, DTR exhibits the worst performance out of all models, in terms of both the training and testing datasets.

TABLE 2: Accuracy statistics of ML regression models using training and testing datasets.

Model	Training dataset			Testing dataset		
	RMSE	MAE	R^2	RMSE	MAE	R^2
XGB	0.0076	0.0063	0.9999	0.4033	0.2845	0.7227
CAT	0.1594	0.1224	0.9565	0.4704	0.3503	0.6229
GBT	0.0018	0.0015	0.9995	0.4812	0.3440	0.5981
MLP-NN	0.3249	0.2663	0.8194	0.4829	0.41115	0.6025
SVR	0.3964	0.3249	0.7313	0.5032	0.4334	0.5684
DTR	0.1056	0.0720	0.9809	0.5712	0.3537	0.4927

Notes: root mean square error, RMSE; mean absolute error, MAE; coefficient of determination, R^2 ; multilayer perceptron neural network, MLP-NN; Support Vector Regression, SVR; Cat Boost Regression, CAT; Decision Tree Regression, DTR; Gradient Boosting Regression, GBR; Extreme Gradient Boosting Regression, XGB.

In this work, a comparison between different activation functions for the MLP-NN model was performed with initial parameters, layers (32, 16), optimizer (“Adam”), and activation (“relu”, “identity”, “sigmoid”, “tanh”, and “logistic”). Table 3 shows the results of using different activation functions acting on the assessment criteria using training and testing datasets, respectively. It should be noticed that parametric study was only conducted on activation functions in the present study.

As shown in Table 3, the “identity” activation function exhibits the best performance, in terms of all error metrics—this model yields the highest value of R^2 and the smallest values of RMSE and MAE.

Figure 3 illustrates the performance of the six models on metrics on the training dataset. Each color of the bars corresponds to one metric. The ranking of the models is marked above each bar. As can be seen from Figure 3, there is a slight difference between the rankings in relation to the training and testing datasets. With the training dataset, the accuracy rankings with respect to the metrics of RMSE are GBT, XGB, DTR, CAT, MLP-NN, and SVR, while with metrics of MAE, the order is GBT, XGB, DTR, CAT, MLP-NN, and SVR. The rankings with metrics of R^2 are XGB, GBT, DTR, CAT, MLP-NN, and SVR.

However, Figure 4 shows that there is a small difference in terms of performance ranking when using the testing dataset. Furthermore, the performance ranking varies as a function of error metrics. The accuracy ranking with RMSE metrics in increasing order is XGB, CAT, GBT, MLP-NN, SVR, and DTR, and that of MAE in increasing order is XGB, GBT, CAT, DTR, MLP-NN, and SVR, while ranking results with R^2 metrics are XGB, CAT, MLP-NN, GBT, SVR, and DTR. Considering the performance using both the training and testing datasets, it can be concluded that XGB yielded the best performance whereas DTR exhibited the worst performance out of these models.

The line and scatter plots of the measured Ra and the predicted values found by the six ML baseline models with the training and testing dataset are presented in Figures 5 and 6 for each trial, respectively. The different colors of the scattered points in the figures represent the values predicted by different baseline models. In the training and testing

TABLE 3: Comparison MLP-NN models under different activation functions.

Activation functions	Training dataset			Testing dataset		
	RMSE	MAE	R^2	RMSE	MAE	R^2
“relu”	0.7349	0.5711	0.0765	0.6913	0.5649	0.1855
“identity”	0.3249	0.2663	0.8194	0.4829	0.4112	0.6025
“sigmoid”	0.7457	0.6447	0.0490	0.7453	0.6257	0.0533
“tanh”	0.8092	0.6965	-0.1198	0.8301	0.7007	-0.1744
“logistic”	0.7656	0.6417	-0.0023	0.7621	0.6678	0.0101

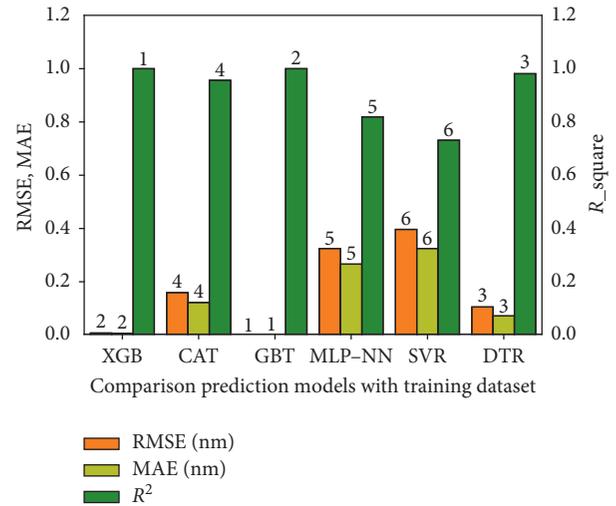


FIGURE 3: Regression of six baseline models’ performance in predicting surface roughness (Ra)—results from 27 trials.

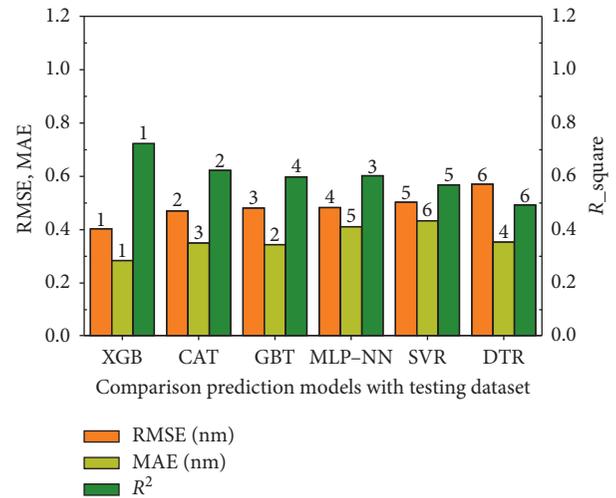


FIGURE 4: Regression of six baseline models’ performance in predicting surface roughness (Ra)—results from 8 trials.

datasets, the values predicted by the XGB model are the closest to the original values of Ra. The value predicted by the CAT model does not closely correspond to the original measured values of Ra.

However, as discussed above, RMSE is the preferred criterion for selecting better models, in relation to the testing

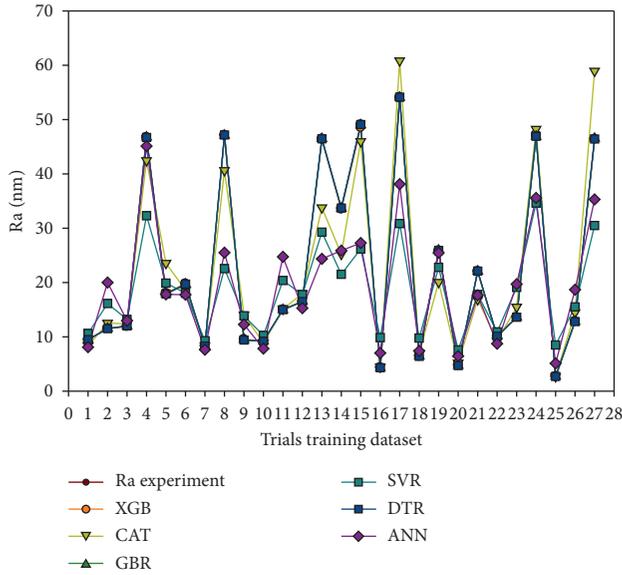


FIGURE 5: Comparison of Ra experiments and prediction of Ra by six deep learning models on the training dataset.

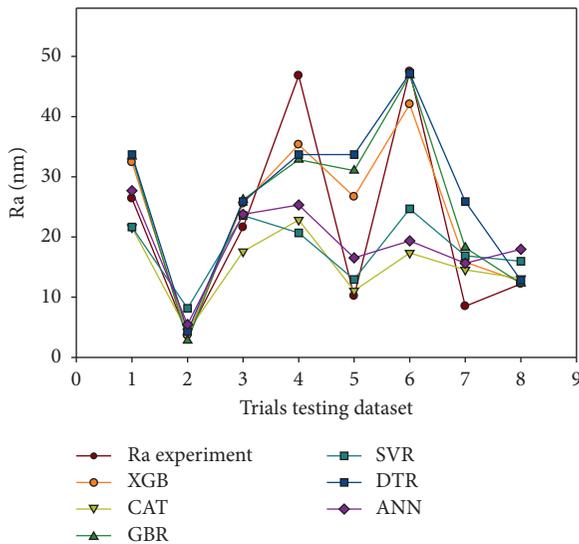


FIGURE 6: Comparison of Ra experiment and prediction of Ra by six deep learning models on the testing dataset.

dataset. Therefore, to continue to improve the prediction accuracy for Ra, we choose 2 models: XGB and CAT, for the next procedure, in keeping with the approach illustrated in Figure 2.

4.2. Description of the Optimization Problems. The XGB and CAT methods require tuning hyperparameters to prevent overfitting and improve model performance. Table 4 presents the hyperparameters of the XGB and CAT models. As mentioned above, XGB and CAT belong to regression with a large number of hyperparameters. The value of the parameter is crucial and so must be carefully selected. However, to date, only heuristic methods have been put forward.

As discussed in Section 2, the overfitting in XGB and CAT is negated by optimizing their hyperparameters using the Bayesian optimization approach. To save computational time, in this paper, we focus only on the hyperparameters that have a significant effect on the model performance, as found in previous studies.

The two optimization problems (one for XGB and one for CAT) of this study are described in Table 4. For XGB, the decision variables are learning_rate, max_depth, subsample, colsample_bytree, reg_alpha, max_leaves, gamma, and min_child_weight. On the other hand, for CAT, the decision variables are learning_rate, depth, bagging_temperature, and num_leaves. The inferior and superior bounds of these variables are also given in Table 4. In this work, the objective of the optimization problems is to minimize the value of RMSE between predicted and experimental data points.

4.3. Hyperparameter Tuning with Bayesian Optimization.

The settings of the Bayesian optimization algorithm search domain are derived from historical data and also from initial tests [72–74]. In terms of other hyperparameters such as “n_estimators” (number of boosted trees), “min_child_samples” (minimum number of data points needed in a leaf), and “subsample_for_bin” (number of samples for constructing bins), default settings in Python have been applied [54]. Table 5 presents the evolution of the optimization procedure for XGB, for instance, whereas Table 6 indicates the best value found for each hyperparameter of the two methods XGB and CAT, respectively.

As discussed above, the objective of this study is to find the best model for predicting surface roughness. Therefore, XGB and CAT have been selected for performance comparison, with the Bayesian optimization algorithm being applied to each model. Table 7 demonstrates the comparison models under the metrics of RMSE, MAE, and R^2 found by XGB and CAT, based on their hyperparameters in the best in the BO and the default in the baseline models. By comparing, we found the effect of hyperparameters on the metrics of the models. The correlation exhibits a promising RMSE and MAE in XGB_opt, achieving the lowest values shown in Figures 7 and 8. Moreover, the metric of R^2 significantly increases from 0.9999 to 1 in performance on the training dataset, and for the testing dataset, it increases from 0.7227 to 0.8924.

The line and scatter plots of the measured Ra and the values predicted by the optimized models XGB_opt and CAT_opt on the training and testing datasets are presented in Figures 9 and 10 for each trial, respectively. The different colors of the scattered points in the figures represent the values predicted by different baseline models. In the training and testing dataset, the predictions of the XGB_opt model are the closest to the original measured values of Ra. The value predicted by the CAT model does not closely correspond to the original values of Ra measured in both datasets.

However, as discussed above, RMSE is the preferred criterion for selecting better models in relation to the training dataset. Therefore, to continue to improve the prediction accuracy for Ra, we choose 2 models: XGB and

TABLE 4: Description of the optimization problems in this study.

Parameter	For XGB	For CAT
Decision variables	learning_rate max_depth subsample colsample_bytree reg_alpha max_leaves gamma min_child_weight	learning_rate depth bagging_temperature num_leaves
Objective	Minimize(test-RMSE)	Minimize(test-RMSE)
Bounds of decision variables	$0.001 \leq \text{learning_rate} \leq 0.3$ $5 \leq \text{max_depth} \leq 30$ $0.5 \leq \text{subsample} \leq 1$ $0.3 \leq \text{colsample_bytree} \leq 1$ $0.005 \leq \text{reg_alpha} \leq 0.02$ $0 \leq \text{max_leaves} \leq 0.02$ $0 \leq \text{gamma} \leq 1$ $1 \leq \text{min_child_weight} \leq 10$	$0.001 \leq \text{learning_rate} \leq 0.5$ $5 \leq \text{depth} \leq 8$ $3 \leq \text{bagging_temperature} \leq 10$ $30 \leq \text{num_leaves} \leq 150$

TABLE 5: Evolution of optimization procedure for XGB (first 50 iterations only).

Iter	Target	Colsample_bytree	Gamma	Learning_rate	Max_depth	Min_child_weight	Reg_alpha	Subsample
1	-7.056	0.8801	0.8997	0.2827	20.34	4.861	0.006904	0.5405
2	-6.516	0.5613	0.4843	0.09498	6.732	5.114	0.007405	0.9069
3	-16.69	0.5848	0.2136	0.1965	19.36	9.066	0.01777	0.6896
4	-8.5	0.9966	0.8161	0.1771	5.423	7.812	0.008559	0.9149
5	-8.822	0.7103	0.598	0.1231	6.284	6.038	0.007801	0.9096
6	-5.852	0.5	0.3518	0.06219	7.254	4.037	0.006944	0.9037
7	-9.343	0.5	0.5106	0.01885	5.633	3.34	0.005851	0.8316
8	-6.15	0.5	0.2702	0.1017	8.293	4.807	0.007843	0.9567
9	-7.601	1	1	0.3	7.727	4.384	0.02	0.5
10	-6.5	1	1	0.3	20.87	2.587	0.005	0.5
11	-6.26	1	1	0.3	22.83	3.751	0.005	0.5
12	-25.55	0.5	0	0.001	23.13	1.845	0.02	1
13	-7.601	1	1	0.3	22.05	4.527	0.005	0.5
14	-6.26	1	1	0.3	19.62	3.248	0.005	0.5
15	-7.601	1	1	0.3	23.95	4.842	0.005	0.5
16	-25.48	0.5	0	0.001	8.813	3.053	0.005	1
17	-5.912	0.5	0.1875	0.2799	7.554	4.955	0.008345	1
18	-25.44	0.5	0.8634	0.001	8.342	5.686	0.005	1
19	-6.418	0.5345	0.2921	0.2898	8.177	4.709	0.01122	0.9844
20	-11.41	0.6047	0.03536	0.01905	7.271	4.588	0.01701	0.6337
21	-5.658	0.5	0.73	0.3	7.259	4.412	0.005	1
22	-6.839	0.9758	0.9844	0.2983	20.45	3.433	0.005328	0.5055
23	-25.48	0.531	0.8686	0.001	6.918	3.876	0.005	1
24	-6.541	0.5	0.4384	0.2496	7.453	4.4	0.00543	0.9837
25	-16.64	0.8722	0.1358	0.05604	13.76	9.028	0.01675	0.5171
26	-5.996	0.6539	0.2812	0.1042	7.178	4.032	0.01673	0.919
27	-6.889	0.7965	0.4768	0.2854	7.075	5.049	0.01465	0.9937
28	-5.911	0.8249	0.2209	0.1936	24.47	1.008	0.01477	0.6331
29	-16.66	0.6999	0.2054	0.2231	16.34	7.16	0.01825	0.5339
30	-6.5	1	1	0.3	20.18	2.931	0.005	0.5
31	-6.559	0.5	0.7239	0.3	7.301	4.792	0.005	0.9979
32	-6.974	0.5	0.08312	0.23	7.39	3.94	0.005782	0.9513
33	-6.219	0.7449	0.6745	0.2195	19.98	3.314	0.01041	0.6499
34	-6.614	0.8682	0.3464	0.03563	14.74	2.282	0.006153	0.8187
35	-6.252	0.8297	0.9535	0.2535	14.33	3.153	0.007105	0.6695
36	-11.64	0.6522	0.5504	0.02992	13.11	6.753	0.01481	0.7805
37	-11.8	0.8327	0.5598	0.02797	15.28	5.888	0.009776	0.6274
38	-6.562	0.7739	0.2751	0.293	17.17	4.494	0.01663	0.8818

TABLE 5: Continued.

Iter	Target	Colsample_bytree	Gamma	Learning_rate	Max_depth	Min_child_weight	Reg_alpha	Subsample
39	-6.339	0.8632	0.9508	0.1429	21.26	1.244	0.01272	0.9479
40	-5.643	0.8481	0.6627	0.1492	14.53	2.736	0.006655	0.741
41	-5.792	0.5397	0.2942	0.09671	7.252	4.123	0.01874	0.8521
42	-5.838	0.8593	0.7312	0.1051	14.02	2.605	0.005175	0.7117
43	-6.016	0.6624	0.2504	0.1153	8.384	3.601	0.00867	0.7995
44	-6.313	0.8624	0.2579	0.2642	13.42	1.179	0.01796	0.9567
45	-5.629	0.7182	0.3777	0.122	8.18	3.918	0.009016	0.7433
46	-5.981	0.7067	0.9652	0.3	14.33	2.576	0.02	0.9897
47	-9.125	0.8218	0.602	0.2969	18.08	6.542	0.0195	0.777
48	-6.014	0.9996	0.9764	0.3	14.38	2.526	0.02	0.5146
49	-16.74	0.7799	0.6707	0.007666	25.28	1.548	0.007938	0.7785
50	-6.792	0.5	0.7475	0.3	14.31	2.629	0.02	0.5203

TABLE 6: Optimized hyperparameters for XGBoost and CAT.

Algorithm	Library	Hyperparameters tuned	Domain	Optimized value
XGB	XGBoost	$n_estimators$	100	100
		$learning_rate$	(0.001, 0.3)	0.1220
		max_depth	(5, 30)	9
		$subsample$	(0.5, 1)	0.7433
		$colsample_bytree$	(0.3, 1)	0.7182
		reg_alpha	(0.005, 0.02)	0.0090
		max_leaves	(0, 0.02)	0
		$gamma$	(0, 1)	0.3778
CAT	catboost	min_child_weight	(1, 10)	3.9175
		$iterations$	(1000, 2000)	1000
		$depth$	(5, 8)	6.1114
		$learning_rate$	(0.001, 0.5)	0.4669
		$bagging_temperature$	(3, 10)	6.8176
		num_leaves	(30, 150)	138

TABLE 7: Statistical accuracy of ML model regression performing 27 trials on the training dataset and 8 trials on the testing dataset.

Model	Training dataset			Testing dataset		
	RMSE	MAE	R^2	RMSE	MAE	R^2
XGB	0.0076	0.0063	0.9999	0.4033	0.2845	0.7227
XGB bayes_opt	0.0047	0.0027	1	0.2512	0.2225	0.8924
CAT	0.1594	0.1224	0.9565	0.4704	0.3503	0.6229
CAT bayes_opt	0.0792	0.0579	0.9893	0.3940	0.2870	0.7355

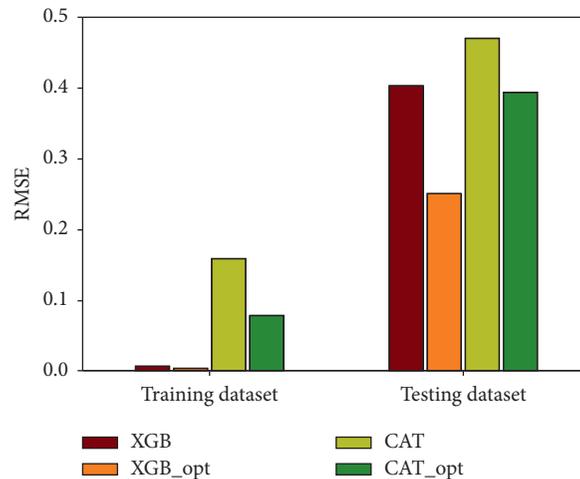


FIGURE 7: Comparison as a function of RMSE.

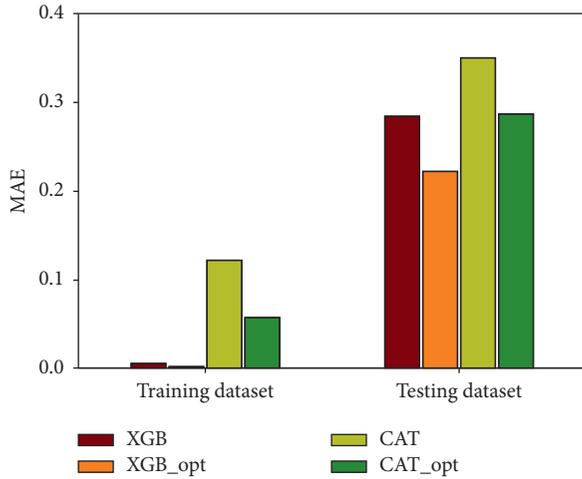


FIGURE 8: Comparison as a function of MAE.

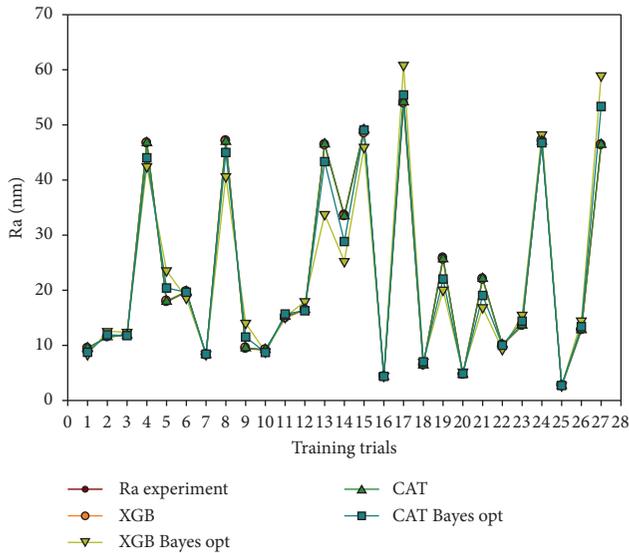


FIGURE 9: Comparison of Ra experiments and prediction of Ra by baseline and their Bayesian optimization of XGB and CAT machine learning models on the training dataset.

CAT, for the next procedure, in keeping with the approach illustrated in Figure 2.

5. Discussion

With the rise of modern technology, in the production process of PC for optical applications by single-point diamond turning, the equipment maintains a stable operating state, so the production process is also carried out stably. Therefore, for given strip specifications, most turning processes will obtain relatively stable datasets without wide variability. The key point of prediction models is to improve prediction performance. The main goal of this paper is to determine the optimal predictive model in relation to surface roughness, by comparing the performances of different

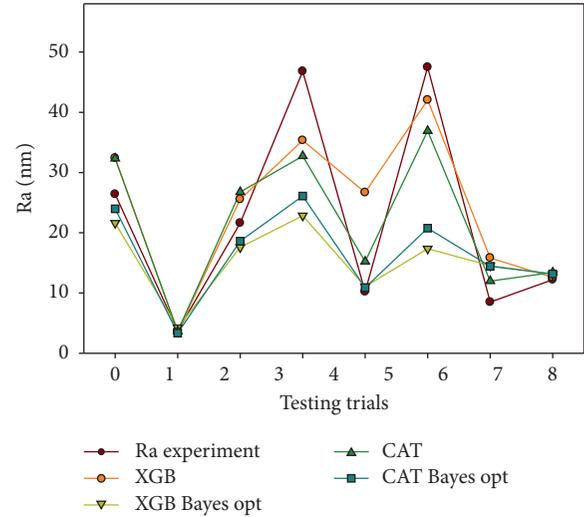


FIGURE 10: Comparison of Ra experiments and prediction of Ra by baseline and their Bayesian optimization of XGB and CAT machine learning models on the testing dataset.

models. In addition, in order to preserve the principal characteristics of the considered models, in this paper, we have not coupled those models with other optimization algorithms. Then, we performed hyperparameter tuning by using BO to find the best model.

By executing the XGB, we can determine which features contribute most heavily to the result. In short, the importance of each feature can be found and plotted. The feature importance graphs for each dataset are plotted in Figure 11. From Figure 11, we can clearly see that Z-axis vibration is the most important feature which has contributed to the prediction of the results, followed by the Y-axis vibration and X-axis vibration. This result is in accordance with Kara and Bayraktar [75], in which experiments showed that the surface roughness increased while increasing vibration of the machining tool (i.e., increase of the cutting speed). For instance, in Kara and Bayraktar [75], the cutting speed exhibited 42.14% of the influence on the surface roughness using the analysis of variance. However, it is worth noticing that the contrary is not confirmed (low value of vibration does not imply good surface quality) Bolat [23]. Also shown in Figure 11, the feed rate is a crucial variable highly affecting the surface roughness of the material. Kara and Bayraktar [75] and Krolczyk et al. [10] have also found similar results in their works. It has been obtained in these studies such that a low value of feed rate provides a small value of surface roughness and vice versa. An important point should be noticed such that there is an influence of feed rate on the surface roughness and energy consumption. Such consideration has not been conducted in the present study. Moreover, another limitation of the present study is that the lack of confirmation tests. Such procedures should be carried in order to assess the performance of the prediction model. Finally, analysis of variance should be processed in further studies to reveal how the input variables influence each other and the surface roughness. Such information is crucial to control the turning process.

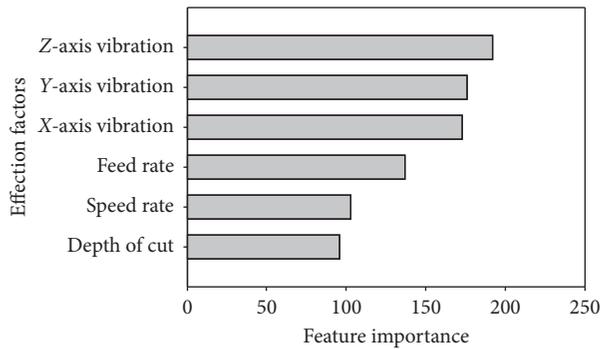


FIGURE 11: Importance of each contribution feature to the result.

6. Conclusions

In this work, six machine learning models—namely, ANN, SVR, CAT, DRT, GBT, and XGB—have been applied to predict surface roughness in the SPDT process. We have compared the performances of the six models using experimental data. Three quality assessment metrics (RMSE, MAE, and R^2) have been used to evaluate the performance of each model. Subsequently, hyperparameter optimization using BO has been applied. The results are summarized as follows:

- (1) The result comparison has shown that there is little difference in terms of ranking when using testing RMSE. For example, the values of RMSE for XGB, CAT, GBT, GBT, MLP-NN, SVR, and DTR are 0.4033, 0.4704, 0.4812, 0.4829, 0.5032, and 0.5712, respectively. However, it has been concluded that overall, XGB, and CAT perform better, followed by GBR, SVR, and DTR, respectively.
- (2) The results in relation to hyperparameters using BO stability show inconsistency in the evaluation metrics. XGB_opt performs best with RMSE, MAE, and R^2 in both the training and testing datasets. It can be seen that using BO has improved significantly the performance of the 2 considered models, especially CAT. Indeed, the value of R^2 when using CAT has increased from 0.9565 to 0.9893 for the training dataset and from 0.6229 to 0.7355 for the testing dataset. The value of MAE when using CAT has decreased from 0.1224 to 0.0579 for the training dataset and from 0.3503 to 0.2870 for the testing dataset. The value of RMSE has decreased from 0.1594 to 0.0792 for the training dataset and from 0.4704 to 0.3940 for the testing dataset.
- (3) Comprehensively considering the prediction accuracy of the six models, XGB appeared to be the best model for predicting surface roughness. Indeed, this model has shown strong performance in terms of different qualitative estimators. For example, XGB exhibited the highest values in terms of R^2 for both training and testing datasets. In terms of other estimators such as RMSE and MAE, XGB also

produced one of the best values. In addition, XGB also performs best when combined with Bayesian optimization algorithm.

- (4) We can clearly see that machine vibration is the most important feature in contributing to the prediction of the results—especially, vibration on the Z-axis, followed by the Y-axis vibration and X-axis vibration. The least important feature is the depth of cut.
- (5) Overall, out of six initially proposed models, we have succeeded to filter out the 2 best models based on different qualitative estimators such as RMSE, MAE, and R^2 for predicting the surface roughness in turning polycarbonate for optical application. In addition, we have improved the performance of these two best models using the Bayesian optimization algorithm.

Certainly, further studies should be conducted to enhance the results of the present paper. From an experimental point of view, confirmation tests should be carried out in order to test the performance of the proposed method in real conditions. Besides, working parameters such as feed rate, cut depth, spindle speed, and cutting speed should be related to the surface roughness by an explicit equation, providing an easy and direct application by engineers/researchers in practice. This means that the proposed machine learning model should be employed to derive such an equation in further research studies. From a computational point of view, uncertainty quantification should also be performed, in order to propagate the variability of the experimental database. Besides, the number of data points for training and testing the models was not large, which may yield some unexpected effect in the results. For such a reason, more data points should be collected in further studies, or experimental tests should be conducted by the researchers' group.

Nomenclature

PC:	Polycarbonate
ML:	Machine learning
BO:	Bayesian optimization
ANN:	Artificial neural network
MLP:	Multilayer perceptron
CAT:	Cat Boost Regression
CAT_opt:	CAT built with BO
SVR:	Support Vector Regression
GBT:	Gradient boosting trees
GBR:	Gradient Boosting Regression
DTR:	Decision Tree Regression
XGB:	Extreme Gradient Boosting Regression
XGB_opt:	XGB built with BO
EDA:	Exploratory data analysis
RMSE:	Root mean square error
MAE:	Mean absolute error
R^2 :	Correlation coefficient
VX:	Vibration of X-axis (mgrms)
VY:	Vibration of Y-axis (mgrms)
VZ:	Vibration of Z-axis (mgrms).

Data Availability

The Excel data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] V. Sudarsan, "Optical materials," in *Functional Materials: Preparation, Processing and Applications*, pp. 285–322, Elsevier, London, UK, 2012.
- [2] M. Roeder, T. Guenther, and A. Zimmermann, "Review on fabrication technologies for optical mold inserts," *Micro-machines*, vol. 10, no. 4, p. 233, 2019.
- [3] Y. Jaluria, *Advanced Materials Processing and Manufacturing*, Springer, Berlin, Germany, 2018.
- [4] C. Gerhard, *Optics Manufacturing: Components and Systems*, CRC Press, Boca Raton, FL, USA, 2017.
- [5] H. de Brouwer, J. van den Bogerd, and J. Hoover, "Color stability of polycarbonate for optical applications," *European Polymer Journal*, vol. 71, pp. 558–566, 2015.
- [6] C. He and W. Zong, "Influencing factors and theoretical models for the surface topography in diamond turning process: a review," *Micromachines*, vol. 10, no. 5, p. 288, 2019.
- [7] V. Saini, D. Sharma, S. Kalla, and T. Chouhan, "Optimisation of process parameter in ultra-precision diamond turning of polycarbonate material," in *Proceedings of the International Conference on Manufacturing Excellence MANFEX*, Noida, India, 2012.
- [8] M. Elangovan, N. R. Sakthivel, S. Saravanamurugan, B. B. Nair, and V. Sugumaran, "Machine learning approach to the prediction of surface roughness using statistical features of vibration signal acquired in turning," *Procedia Computer Science*, vol. 50, pp. 282–288, 2015.
- [9] G. Krolczyk, M. Gajek, and S. Legutko, "Predicting the tool life in the dry machining of duplex stainless steel," *Eksploatacja I Niezawodnosc—Maintenance and Reliability*, vol. 15, no. 1, pp. 62–65, 2013.
- [10] G. Krolczyk, S. Legutko, and M. Gajek, "Predicting the surface roughness in the dry machining of duplex stainless steel (DSS)," *Metalurgija-Sisak Then Zagreb*, vol. 52, pp. 259–262, 2013.
- [11] M. Kuntoğlu, A. Aslan, D. Y. Pimenov, K. Giasin, T. Mikolajczyk, and S. Sharma, "Modeling of cutting parameters and tool geometry for multi-criteria optimization of surface roughness and vibration via response surface methodology in turning of AISI 5140 steel," *Materials (Basel, Switzerland)*, vol. 13, p. 4242, 2020.
- [12] M. K. Gupta, M. Mia, C. I. Pruncu et al., "Parametric optimization and process capability analysis for machining of nickel-based superalloy," *The International Journal of Advanced Manufacturing Technology*, vol. 102, no. 9–12, pp. 3995–4009, 2019.
- [13] A. Eser, E. Aşkar Ayyıldız, M. Ayyıldız, and F. Kara, "Artificial intelligence-based surface roughness estimation modelling for milling of AA6061 alloy," *Advances in Materials Science and Engineering*, vol. 2021, Article ID 5576600, 10 pages, 2021.
- [14] M. S. Alajmi and A. M. Almeshal, "Prediction and optimization of surface roughness in a turning process using the ANFIS-QPSO method," *Materials*, vol. 13, no. 13, p. 2986, 2020.
- [15] A. T. Abbas, M. Alata, A. E. Ragab, M. M. El Rayes, and E. A. El Danaf, "Prediction model of cutting parameters for turning high strength steel grade-H: comparative study of regression model versus ANFIS," *Advances in Materials Science and Engineering*, vol. 2017, Article ID 2759020, 12 pages, 2017.
- [16] S. K. Pal and D. Chakraborty, "Surface roughness prediction in turning using artificial neural network," *Neural Computing and Applications*, vol. 14, no. 4, pp. 319–324, 2005.
- [17] M. Nalbant, H. Gokkaya, and İ. Toktaş, "Comparison of regression and artificial neural network models for surface roughness prediction with the cutting parameters in CNC turning," *Modelling and Simulation in Engineering*, vol. 2007, Article ID 092717, 14 pages, 2007.
- [18] Y. Ö. Özgören, S. Çetinkaya, S. Sarıdemir, A. Çiçek, and F. Kara, "Artificial neural network based modelling of performance of a beta-type Stirling engine," *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, vol. 227, no. 3, pp. 166–177, 2013.
- [19] N. K. Vuong, Y. Xue, S. Liu, Y. Zhou, and M. Wu, "Predicting surface roughness and flank wear in turning processes," in *Proceedings of the 2020 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pp. 1–6, IEEE, Detroit, MI, USA, 2020.
- [20] T. Zhou, L. He, J. Wu, F. Du, and Z. Zou, "Prediction of surface roughness of 304 stainless steel and multi-objective optimization of cutting parameters based on GA-GBRT," *Applied Sciences*, vol. 9, no. 18, p. 3684, 2019.
- [21] Z. Jurkovic, G. Cukor, M. Brezocnik, and T. Brajkovic, "A comparison of machine learning methods for cutting parameters prediction in high speed turning process," *Journal of Intelligent Manufacturing*, vol. 29, no. 8, pp. 1683–1693, 2018.
- [22] D. Y. Pimenov, A. Bustillo, and T. Mikolajczyk, "Artificial intelligence for automatic prediction of required surface roughness by monitoring wear on face mill teeth," *Journal of Intelligent Manufacturing*, vol. 29, no. 5, pp. 1045–1061, 2018.
- [23] M. Bolat, *Machining of Polycarbonate for Optical Applications*, Middle East Technical University, Ankara, Turkey, 2013.
- [24] H. Thanh Duong, H. Chi Phan, T.-T. Le, and N. Duc Bui, "Optimization design of rectangular concrete-filled steel tube short columns with balancing composite motion optimization and data-driven model," *Structures*, vol. 28, pp. 757–765, 2020.
- [25] F. Morin and Y. Bengio, "Hierarchical probabilistic neural network language model," in *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pp. 246–252, Bridgetown, Barbados, January 2005.
- [26] T.-T. Le, V.-H. Nguyen, and M. V. Le, "Development of deep learning model for the recognition of cracks on concrete surfaces," *Applied Computational Intelligence and Soft Computing*, vol. 2021, Article ID 8858545, 10 pages, 2021.
- [27] A. C. Tsoi and A. Back, "Discrete time recurrent neural network architectures: a unifying review," *Neurocomputing*, vol. 15, no. 3–4, pp. 183–223, 1997.
- [28] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," 2016, <http://arxiv.org/abs/11601.06733>.
- [29] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 4, pp. 809–821, 2015.

- [30] T.-T. Le, "Prediction of tensile strength of polymer carbon nanotube composites using practical machine learning method," *Journal of Composite Materials*, vol. 55, no. 6, 2020.
- [31] T.-T. Le, "Practical hybrid machine learning approach for estimation of ultimate load of elliptical concrete-filled steel tubular columns under axial loading," *Advances in Civil Engineering*, vol. 2020, Article ID 8832522, 19 pages, 2020.
- [32] T.-T. Le, "Practical machine learning-based prediction model for axial capacity of square CFST columns," *Mechanics of Advanced Materials and Structures*, pp. 1–16, 2020.
- [33] N. X. Ho and T.-T. Le, "Effects of variability in experimental database on machine-learning-based prediction of ultimate load of circular concrete-filled steel tubes," *Measurement*, vol. 176, Article ID 109198, 2021.
- [34] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [35] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Advances in Neural Information Processing Systems*, vol. 9, pp. 155–161, 1997.
- [36] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.
- [37] H. Yang, L. Chan, and I. King, "Support vector machine regression for volatile stock market prediction," in *Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning*, pp. 391–396, Springer, Guimaraes, Portugal, November 2002.
- [38] D. Gupta, M. Pratama, Z. Ma, J. Li, and M. Prasad, "Financial time series forecasting using twin support vector regression," *PLoS One*, vol. 14, Article ID e0211402, 2019.
- [39] B. M. Henrique, V. A. Sobreiro, and H. Kimura, "Stock price prediction using support vector regression on daily and up to the minute prices," *The Journal of Finance and Data Science*, vol. 4, no. 3, pp. 183–201, 2018.
- [40] V. Anandhi and R. Manicka Chezian, "Support vector regression to forecast the demand and supply of pulpwood," *International Journal of Future Computer and Communication*, vol. 2, pp. 266–269, 2013.
- [41] S. Kumar, V. Kumar, and R. K. Sharma, "Rice yield forecasting using support vector machine," *International Journal of Recent Technology and Engineering*, vol. 8, no. 4, pp. 2588–2593, 2019.
- [42] B. B. Sahoo, R. Jha, A. Singh, and D. Kumar, "Application of support vector regression for modeling low flow time series," *KSCE Journal of Civil Engineering*, vol. 23, no. 2, pp. 923–934, 2019.
- [43] A. Suárez Sánchez, P. J. García Nieto, P. Riesgo Fernández, J. J. del Coz Díaz, and F. J. Iglesias-Rodríguez, "Application of an SVM-based regression model to the air quality study at local scale in the Avilés urban area (Spain)," *Mathematical and Computer Modelling*, vol. 54, no. 5–6, pp. 1453–1466, 2011.
- [44] S. Saravanamurugan, S. Thiyagu, N. R. Sakthivel, and B. B. Nair, "Chatter prediction in boring process using machine learning technique," *International Journal of Manufacturing Research*, vol. 12, no. 4, pp. 405–422, 2017.
- [45] T.-T. Le and M. V. Le, "Development of user-friendly kernel-based Gaussian process regression model for prediction of load-bearing capacity of square concrete-filled steel tubular members," *Materials and Structures*, vol. 54, no. 2, p. 59, 2021.
- [46] P. W. Khan, Y.-C. Byun, S.-J. Lee, D.-H. Kang, J.-Y. Kang, and H.-S. Park, "Machine learning-based approach to predict energy consumption of renewable and nonrenewable power sources," *Energies*, vol. 13, no. 18, p. 4870, 2020.
- [47] A. V. Dorogush, V. Ershov, and A. Gulin, "CatBoost: gradient boosting with categorical features support," 2018, <http://arxiv.org/abs/1810.11363>.
- [48] G. Huang, L. Wu, X. Ma et al., "Evaluation of CatBoost method for prediction of reference evapotranspiration in humid regions," *Journal of Hydrology*, vol. 574, pp. 1029–1041, 2019.
- [49] Y. Zhang, J. Ma, S. Liang, X. Li, and M. Li, "An evaluation of eight machine learning regression algorithms for forest aboveground biomass estimation from multiple satellite data products," *Remote Sensing*, vol. 12, no. 24, p. 4015, 2020.
- [50] O. Z. Maimon and L. Rokach, *Data Mining with Decision Trees: Theory and Applications*, World Scientific, Singapore, 2014.
- [51] H. C. Phan, T.-T. Le, N. D. Bui, H. T. Duong, and T. D. Pham, "An empirical model for bending capacity of defected pipe combined with axial load," *International Journal of Pressure Vessels and Piping*, vol. 191, Article ID 104368, 2021.
- [52] M. S. Alajmi and A. M. Almeshal, "Predicting the tool wear of a drilling process using novel machine learning XGBoost-SDA," *Materials*, vol. 13, no. 21, p. 4952, 2020.
- [53] S. Ramraj, N. Uzir, R. Sunil, and S. Banerjee, "Experimenting XGBoost algorithm for prediction and classification of different datasets," *International Journal of Control Theory and Applications*, vol. 9, pp. 651–662, 2016.
- [54] T. Chen and C. Guestrin, "Xgboost: a scalable tree boosting system," in *Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, San Francisco, CA, USA, August 2016.
- [55] S. Cerna, C. Guyeux, G. Royer, C. Chevallier, and G. Plumerel, "Predicting fire brigades operational breakdowns: a real case study," *Mathematics*, vol. 8, no. 8, p. 1383, 2020.
- [56] K. D. Kankanamge, Y. R. Witharanage, C. S. Withanage, M. Hansini, D. Lakmal, and U. Thayasivam, "Taxi trip travel time prediction with isolated XGBoost regression," in *Proceedings of the 2019 Moratuwa Engineering Research Conference (MERCon)*, pp. 54–59, IEEE, Moratuwa, Sri Lanka, 2019.
- [57] Y. Li, X. Meng, Z. Zhang, and G. Song, "A remaining useful life prediction method considering the dimension optimization and the iterative speed," *IEEE Access*, vol. 7, pp. 180383–180394, 2019.
- [58] R. S. Peres, J. Barata, P. Leitao, and G. Garcia, "Multistage quality control using machine learning in the automotive industry," *IEEE Access*, vol. 7, pp. 79908–79916, 2019.
- [59] L. Munkhdalai, T. Munkhdalai, O.-E. Namsrai, J. Lee, and K. Ryu, "An empirical comparison of machine-learning methods on bank client credit assessments," *Sustainability*, vol. 11, no. 3, p. 699, 2019.
- [60] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281–285, 2012.
- [61] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS 2011)*, Granada, Spain, 2011.
- [62] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox, "Hyperopt: a python library for model selection and hyperparameter optimization," *Computational Science & Discovery*, vol. 8, no. 1, 2015.
- [63] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Auto-WEKA: combined selection and hyperparameter

- optimization of classification algorithms,” in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 847–855, Chicago, IL, USA, 2013.
- [64] F. Hutter, H. H. Hoos, and K. Leyton-Brown, “Sequential model-based optimization for general algorithm configuration,” in *Proceedings of the International Conference on Learning and Intelligent Optimization*, pp. 507–523, Springer, Athens, Greece, 2011.
- [65] H. Abbasimehr and R. Paki, “Prediction of COVID-19 confirmed cases combining deep learning methods and Bayesian optimization, Chaos,” *Solitons & Fractals*, vol. 142, Article ID 110511, 2020.
- [66] K. Zhang, L. Zheng, Z. Liu, and N. Jia, “A deep learning based multitask model for network-wide traffic speed prediction,” *Neurocomputing*, vol. 396, pp. 438–450, 2020.
- [67] T.-T. Le and H. C. Phan, “Prediction of ultimate load of rectangular CFST columns using interpretable machine learning method,” *Advances in Civil Engineering*, vol. 2020, Article ID 8855069, 16 pages, 2020.
- [68] T.-T. Le and M. V. Le, “Nanoscale effect investigation for effective bulk modulus of particulate polymer nanocomposites using micromechanical framework,” *Advances in Materials Science and Engineering*, vol. 2021, Article ID 1563845, 13 pages, 2021.
- [69] T.-T. Le, “Probabilistic modeling of surface effects in nano-reinforced materials,” *Computational Materials Science*, vol. 186, Article ID 109987, 2021.
- [70] T.-T. Le, “Probabilistic investigation of the effect of stochastic imperfect interfaces in nanocomposites,” *Mechanics of Materials*, vol. 151, Article ID 103608, 2020.
- [71] T. Chai and R. R. Draxler, “Root mean square error (RMSE) or mean absolute error (MAE)?—arguments against avoiding RMSE in the literature,” *Geoscientific Model Development*, vol. 7, no. 3, pp. 1247–1250, 2014.
- [72] W. Y. Loh, “Classification and regression trees,” *WIREs Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 14–23, 2011.
- [73] J. Elith, J. R. Leathwick, and T. Hastie, “A working guide to boosted regression trees,” *Journal of Animal Ecology*, vol. 77, no. 4, pp. 802–813, 2008.
- [74] G. G. Moisen, E. A. Freeman, J. A. Blackard, T. S. Frescino, N. E. Zimmermann, and T. C. Edwards Jr., “Predicting tree species presence and basal area in Utah: a comparison of stochastic gradient boosting, generalized additive models, and tree-based methods,” *Ecological Modelling*, vol. 199, no. 2, pp. 176–187, 2006.
- [75] F. Kara and F. Bayraktar, “Investigation of the effect on surface roughness of cryogenic process applied to cutting tool,” *International Journal of Analytical, Experimental and Finite Element Analysis*, vol. 7, no. 2, pp. 19–27, 2020.