Reports in Mechanical Engineering
Vol. 2, No. 1, 2021, pp. 190-201.
ISSN: 2683-5894, DOI: https://doi.org/10.31181/rme2001021901b

Application of XGBoost algorithm as a predictive tool in a CNC turning process

Shibaprasad Bhattacharya[1] and Shankar Chakraborty[2]

[1]Department of Production Engineering, Jadavpur University, India, e-mail: shibaprasadb@yahoo.com
[2]Department of Production Engineering, Jadavpur University, India, e-mail: s_chakraborty00@yahoo.co.in

Article Info

Article history:

Received Jun 12, 2021
Revised August 29, 2020
Accepted September 6, 2021

Keywords:

CNC turning,
XGBoost,
Prediction,
Response.

ABSTRACT

In this paper, an ensemble learning method, in the form of extreme gradient boosting (XGBoost) algorithm is adopted as an effective predictive tool for envisaging values of average surface roughness and material removal rate during CNC turning operation of high strength steel grade-H material. In order to develop the related models, a grid with 24600 combinations of different hyperparameters is created and tested for all the possible hyperparametric combinations of the model. The configurations having the optimal values of the considered hyperparameters and yielding the lowest training error are finally employed for predicting the response values in the CNC turning process. The performance of the developed models is finally validated with the help of five statistical error estimators, i.e. mean absolute percentage error, root mean squared percentage error, root mean squared logarithmic error, correlation coefficient and root relative squared error. Based on the favorable values of all the statistical metrics, it can be observed that XGBoost can be efficiently applied as a predictive tool with excellent accuracy in machining processes.

Corresponding Author:

Shankar Chakraborty,
Department of Production Engineering, Jadavpur University, India.
Email: s_chakraborty00@yahoo.co.in

1. Introduction

In the field of manufacturing, machining is the process of removing unwanted material from a given workpiece to provide it the required shape and geometry, while meeting the requirements for better surface quality and close dimensional tolerance. Among all the machining operations, turning using a non-rotating single-point cutting tool plays a significant role in removing material from the outer diameter of a rotating cylindrical workpiece while reducing its diameter to a specified size and obtaining a smooth surface after machining (Haynes, 2018). It is also known as subtractive machining process. In computer numerical control (CNC) turning operation, usually a cylindrically shaped material is clamped on a mandrel and rotated, while a cutting tool is fed against it to remove material and generate the desired shape (Lan & Wang, 2009). The turret with additional tool is programmed to perform the desired machining operation while generating the required part geometries and features based on the input drawing. Although CNC machine tools are more cost-intensive and complex in operation as compared to conventional lathes, they outperform their manual counterparts with respect to high production rate, flexibility, precision, customization, minimum human error etc. Their capability to generate perfect copies of design with minimum human intervention makes them a popular choice in present-day manufacturing industries (Lan & Wang, 2009). Initially high investment for a CNC lathe would later be compensated by high volume of production of almost defect-free end products/components.

Like all other machining processes, the outputs (responses) of a CNC turning operation, like material removal rate (MRR), average surface roughness (Ra), tool wear, tool tip vibration, tool-workpiece interface temperature, energy consumption etc. are significantly influenced by its several input parameters, such as cutting speed (Vc), depth of cut (t), feed rate (f), types of the tool and work material, tool nose radius, workpiece hardness, type of the lubricant, machining environment etc. It is usually a customary practice to evaluate the quality of a final product based on these responses. For this reason, the concerned process designer/machine operator must closely control and better understand the effects and interactions of different turning parameters on the responses. Based on the available experimental data, various statistical and machine learning techniques can be deployed to effectively model the existent interrelationships between the turning parameters and responses (Han & Chi, 2016). The developed models can be employed as predictive tools to envisage the tentative response values for a given set of turning parameters. These models can also be applied to perceive the responses of CNC turning operations using the same set of input parameters. There are several different types of statistical and machine learning techniques employed for this purpose. Table 1 provides a comparative study of the most popular statistical and machine learning techniques with respect to problem type, assumptions, interpretability, accuracy, training speed, amount of parameter tuning and performance with smaller datasets. Due emphasis needs to be provided on these features while choosing an appropriate tool for prediction purposes.

Table 1. Comparative analysis of various statistical and machine learning techniques

Technique	Problem type	Assumptions	Interpretability	Predictive accuracy	Training speed	Parameter tuning	Performance with small data
Linear regression	Regression	Normality	Yes	Lower	Fast	None	Good
Logistic regression	Classification	Normality	Yes	Lower	Fast	None	Good
Linear discriminant analysis	Classification	Normality	Yes	Lower	Fast	None	Good
k-nearest neighbour	Classification	None	Yes	Lower	Fast	Minimal	Poor
Support vector machine	Both	None	Yes	Lower	Fast	Minimal	Poor
Naïve Bayes	Classification	None	Somewhat	Lower	Fast	Minimal	Good
Decision trees	Both	None	Somewhat	Lower	Fast	Minimal	Poor
Random forests	Both	None	Somewhat	Higher	Slow	Moderate	Poor
Neural networks	Both	None	No	Higher	Slow	Large	Poor

Extreme gradient boosting (XGBoost) supersedes the other machine learning algorithms in regard of predictive accuracy, training speed, normality assumption of the input variables, interpretability, requirement of minimal tuning parameters etc. For its effective application as classification and regression tool, the existent relationship between the input and output variables needs not to be always linear. These advantageous features of XGBoost algorithm thus make it a suitable choice among the research community as a regressor to predict output variables based on a set of input variables. Its interpretability is comparable with that of random forests, but its predictive accuracy is higher than random forests when trained properly with the corresponding hyperparameters.

XGBoost has already found wide-ranging applications in diverse domains of manufacturing. Kiangala and Wang (2021) employed both XGBoost and random forest to develop an adaptive and effective customization framework in a real-time Industry 4.0 environment, leading to synchronization of a single customer input with better customizable outputs from a manufacturing process. Deng et al. (2019) endeavoured to predict bead geometry for a multi-layer wire and arc additive manufacturing process based on the application of XGBoost algorithm. It was noticed that XGBoost would outperform artificial neural networks even with a small dataset which could otherwise lead to overfitting of the model. In material science, the relationship between steel properties, their compositions and manufacturing parameters is extremely difficult to comprehend. For this purpose, Song et al. (2020) adopted linear regression, support vector machine and XGBoost to determine the mapping functions between tensile strength, plasticity and other influencing factors for steel. The developed mapping functions were later employed as the fitness values of particle swarm optimization technique to propose a steel property optimization model. Finally, the experimental results were analyzed theoretically, demonstrating superiority of XGBoost over the over considered algorithms. Gao et al. (2019) applied XGBoost algorithm as a material removal prediction model for robotic belt grinding operation of Inconel 718 material.

A comparative study of the derived results showed its supremacy over other machine learning algorithms, like support vector machine, radian function model etc.

In quality monitoring, Chen et al. (2019) examined the relationships between welding current, welding speed, energy input and weld bead geometry during metal active gas welding operation. Based on XGBoost, two data driven models were proposed to recognize penetration status and predict bead reinforcement quality. A novel regression model, in the form of random forest-principal component analysis-XGBoost, was developed by Zhang et al. (2021) for on-line prediction of seam tensile strength of Al-Li alloy during laser welding process. Excellent coefficient of determination values proved the efficacy of the proposed XGBoost-based approach over the other tree-based ensemble learning models as predictive tools.

Another investigation on fatigue strength behaviour of steels was conducted by Choi (2019). Six different numerical models were designed for predicting fatigue strength of steel, and with the help of statistical inference analysis, superiority of XGBoost algorithm over the others was validated. While predicting tool wear during a drilling operation, Alajmi et al. (2020) proved the effectiveness of XGBoost algorithm against support vector machine and multilayer perception artificial neural network. Although this algorithm has been successfully employed in different discrete domains of manufacturing, its application as an efficient predictive tool based on real-time machining data is really scarce. Thus, this paper presents the application of XGBoost algorithm to envisage values of MRR (in mm^3/min) and Ra (in μm) during CNC turning operation of high strength grade-H steel material.

2. Experimental data

Taking into consideration Vc, t and f as the input parameters, and MRR and Ra as the responses, Abbas et al. (2017) conducted 5^3 (125) experiments using an EMCO Concept Turn CNC lathe equipped with Sinumeric 840-D controller on high strength grade-H steel materials. During the turning operation, an uncoated tungsten-carbide insert was employed as the cutting tool. The work material, also known as 'gun steel', has found wide applications in manufacturing of gun barrel and muzzle brake as military and civilian products. Before the turning operation, the workpiece was forged into cylindrical form and subsequently annealed to remove the residual stresses. All the turning parameters were varied at five different operating levels, as shown in Table 2. The experimental details and values of the measured responses are provided in Table 3. Among 125 experimental observations (classified into 25 groups), group numbers 3, 10, 14, 16 and 22 are randomly chosen for testing the prediction performance of XGBoost algorithm, whereas, the remaining 20 groups are utilized for training of this algorithm.

Table 2. CNC turning parameters and their levels (Abbas et al., 2017)

Turning parameter	Unit	Level				
		1	2	3	4	5
Cutting speed	m/min	75	100	125	150	175
Depth of cut	mm	0.15	0.3	0.45	0.6	0.75
Feed rate	mm/rev	0.02	0.04	0.08	0.16	0.32

3. XGBoost as a predictive tool

In machine learning, a subfield of artificial intelligence, statistical methods are employed to train machines so that they can mimic human behaviour. Thus, the goal of machine learning algorithms is to better generalize existing problems while providing accurate solutions. To achieve the desired goal, the designers need to train different learners which due to the presence of randomness in the data may often become very weak. Ensemble learning is a prototype of machine learning (Zhou, 2009) where several learners are combined together to act as an effective prediction tool. The two most commonly used ensemble learning methods are bagging and boosting (Oza& Russell, 2001). Bagging aggregation or bootstrapping is a parallel aggregation method, while boosting is considered as a sequential aggregation method. The ensemble learning model is found to be most suitable for machine learning techniques which are usually unstable, like decision trees, ANN etc. (Kittler &Roli, 2003). It helps the learners to be aggregated together so that they can provide different generalization patterns, while minimizing variability in the model to some extent (Brown, 2011). The XGBoost algorithm belongs to the second group of ensemble learning model, and can be employed for both regression and classification purposes (Friedman, 2001; Chen & Guestrin, 2016). The evolution process from a simple decision tree to XGBoost is illustrated in Figure 1.

Table 3. Experimental data (Abbas et al., 2017)

Group	Vc	t	f	Ra	MRR	Group	Vc	t	f	Ra	MRR
	75	0.15	0.020	0.100	225		125	0.60	0.020	0.292	1500
	75	0.15	0.040	0.212	450		125	0.60	0.040	0.558	3000
1	75	0.15	0.080	0.650	900	14	125	0.60	0.080	1.005	6000
	75	0.15	0.160	1.415	1800		125	0.60	0.160	2.028	12000
	75	0.15	0.320	3.121	3600		125	0.60	0.320	4.030	24000
	75	0.30	0.020	0.152	450		125	0.75	0.020	0.441	1875
	75	0.30	0.040	0.368	900		125	0.75	0.040	0.670	3750
2	75	0.30	0.080	0.738	1800	15	125	0.75	0.080	1.118	7500
	75	0.30	0.160	1.645	3600		125	0.75	0.160	2.028	15000
	75	0.30	0.320	3.312	7200		125	0.75	0.320	4.119	30000
	75	0.45	0.020	0.182	675		150	0.15	0.020	0.123	450
	75	0.45	0.040	0.389	1350		150	0.15	0.040	0.254	900
3	75	0.45	0.080	0.798	2700	16	150	0.15	0.080	0.783	1800
	75	0.45	0.160	1.699	5400		150	0.15	0.160	1.698	3600
	75	0.45	0.320	3.386	10800		150	0.15	0.320	3.745	7200
	75	0.60	0.020	0.261	900		150	0.30	0.020	0.184	900
	75	0.60	0.040	0.499	1800		150	0.30	0.040	0.442	1800
4	75	0.60	0.080	0.898	3600	17	150	0.30	0.080	0.885	3600
	75	0.60	0.160	1.811	7200		150	0.30	0.160	1.974	7200
	75	0.60	0.320	3.599	14400		150	0.30	0.320	3.974	14400
	75	0.75	0.020	0.394	1125		150	0.45	0.020	0.218	1350
	75	0.75	0.040	0.599	2250		150	0.45	0.040	0.467	2700
5	75	0.75	0.080	0.999	4500	18	150	0.45	0.080	0.957	5400
	75	0.75	0.160	1.811	9000		150	0.45	0.160	2.038	10800
	75	0.75	0.320	3.678	18000		150	0.45	0.320	4.063	21600
	100	0.15	0.020	0.106	300		150	0.60	0.020	0.313	1800
	100	0.15	0.040	0.225	600		150	0.60	0.040	0.598	3600
6	100	0.15	0.080	0.688	1200	19	150	0.60	0.080	1.077	7200
	100	0.15	0.160	1.497	2400		150	0.60	0.160	2.173	14400
	100	0.15	0.320	3.311	4800		150	0.60	0.320	4.318	28800
	100	0.30	0.020	0.162	600		150	0.75	0.020	0.472	2250
	100	0.30	0.040	0.389	1200		150	0.75	0.040	0.718	4500
7	100	0.30	0.080	0.782	2400	20	150	0.75	0.080	1.198	9000
	100	0.30	0.160	1.743	4800		150	0.75	0.160	2.173	18000
	100	0.30	0.320	3.504	9600		150	0.75	0.320	4.413	36000
	100	0.45	0.020	0.193	900		175	0.15	0.020	0.128	525
	100	0.45	0.040	0.412	1800		175	0.15	0.040	0.272	1050
	100	0.45	0.080	0.845	3600		175	0.15	0.080	0.832	2100
8	100	0.45	0.160	1.798	7200	21	175	0.15	0.160	1.812	4200
	100	0.45	0.320	3.585	14400		175	0.15	0.320	3.994	8400
	100	0.60	0.020	0.276	1200		175	0.30	0.020	0.195	1050
	100	0.60	0.040	0.527	2400		175	0.30	0.040	0.471	2100
9	100	0.60	0.080	0.950	4800	22	175	0.30	0.080	0.945	4200
	100	0.60	0.160	1.916	9600		175	0.30	0.160	2.106	8400
	100	0.60	0.320	3.807	19200		175	0.30	0.320	4.239	16800
	100	0.75	0.020	0.416	1500		175	0.45	0.020	0.232	1575
	100	0.75	0.040	0.633	3000		175	0.45	0.040	0.497	3150
10	100	0.75	0.080	1.057	6000	23	175	0.45	0.080	1.021	6300
	100	0.75	0.160	1.916	12000		175	0.45	0.160	2.174	12600
	100	0.75	0.320	3.891	24000		175	0.45	0.320	4.334	25200
	125	0.15	0.020	0.112	375		175	0.60	0.020	0.334	2100
	125	0.15	0.040	0.237	750		175	0.60	0.040	0.638	4200
11	125	0.15	0.080	0.728	1500	24	175	0.60	0.080	1.149	8400
	125	0.15	0.160	1.585	3000		175	0.60	0.160	2.318	16800
	125	0.15	0.320	3.495	6000		175	0.60	0.320	4.606	33600
	125	0.30	0.020	0.170	750		175	0.75	0.020	0.504	2625
	125	0.30	0.040	0.412	1500		175	0.75	0.040	0.767	5250
12	125	0.30	0.080	0.826	3000	25	175	0.75	0.080	1.278	10500
	125	0.30	0.160	1.843	6000		175	0.75	0.160	2.318	21000
	125	0.30	0.320	3.709	12000		175	0.75	0.320	4.707	42000

Table 3. Continued

	125	0.45	0.020	0.203	1125
	125	0.45	0.040	0.435	2250
13	125	0.45	0.080	0.893	4500
	125	0.45	0.160	1.902	9000
	125	0.45	0.320	3.792	18000

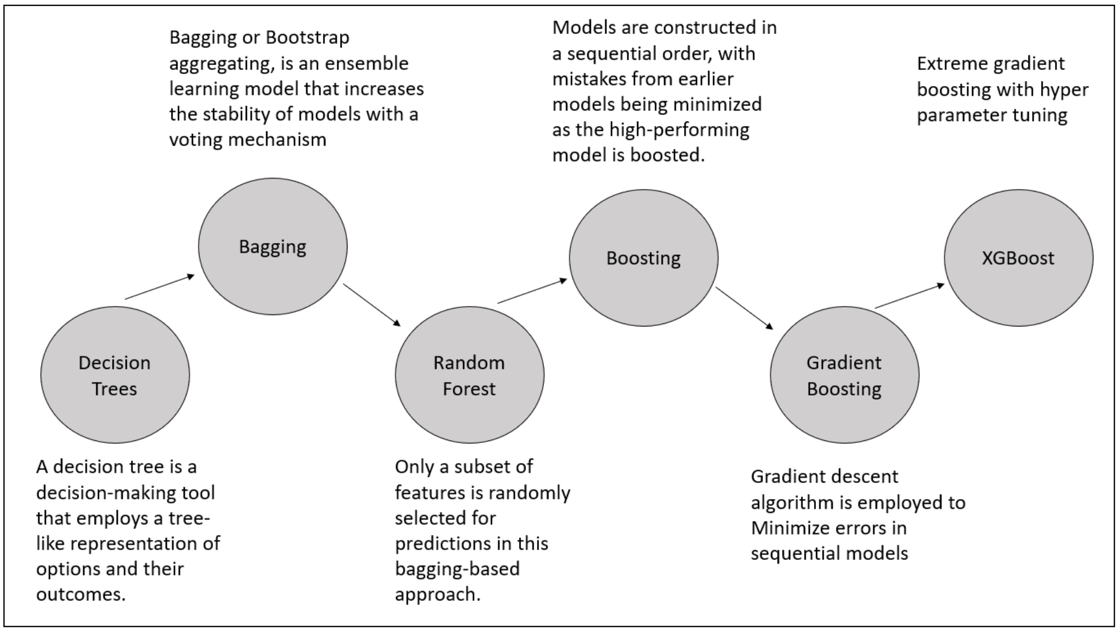


Figure 1. Evolution of XGBoost from a simple decision tree

In decision trees, a universe of objects is defined with a collection of attributes (Quinlan, 1986). When these attributes (discrete or continuous, categorical or numerical) are taken together, they characterize the objects belonging to two or more sets of mutually exclusive classes. As these decision trees suffer from the problem of instability, ensemble learning models are applied to combine them while providing more stability as well as predictive accuracy. In bagging, these decision trees are coupled together and the final solution is derived after taking into account the output from each of the decision trees. Regression-based problems can also be solved using bagging technique where all the decision trees are combined and their average value is considered as the final solution. Another tree-based bagging process is random forest where different tuning parameters, like tree depth, number of trees in each split etc. are utilized to reduce variability in the model.

On the other hand, boosting is an ensemble learning technique where models are constructed in sequential order with a goal to minimize the learning error in each sequential step. One of the most widely used and popular boosting techniques is Adaboost where weak learners are strengthened to have better results (Schapire, 2013). The boosting techniques have further been modified with the arrival of gradient boosting models where gradient descent algorithms are deployed to minimize errors in the sequential models more quickly and accurately. Catboost is one such boosting technique where gradient boosting mechanism is employed (Dorogush et al., 2018).

The predictive power of gradient boosting techniques has been rapidly changed with the arrival of XGBoost algorithm with introduction of more additive optimization of the functional space. In XGBoost, besides the usual regularization of gradient boosting techniques, two additional features, i.e. shrinkage and column subsampling have been incorporated to prevent overfitting of data and increase robustness of the model (Friedman, 2002). Shrinkage, which is analogous to learning rate in stochastic optimization, scales the newly added weights by a factor after each step of tree boosting. It minimizes the influence of each tree, leaving space for future trees to further improve the model. On the other hand, column subsampling greatly reduces the computational time of this algorithm (Friedman & Popescu, 2003). In order to fit the data into an XGBoost model, appropriate parameter values need to be selected. In this paper, to predict responses of the said CNC turning process using XGBoost algorithm, open-sourced software R is used (Wickham et al., 2019). In

XGBoost algorithm, there are mainly two types of parameters, i.e. booster parameters and learning task parameters which are briefly discussed as below (Chen et al., 2015):

Booster parameters

a) *min_child_weight*: It refers to the minimum 'sum of weights' of an observation and is quite similar to the minimum child leaf of gradient boosting machine. The default value is 1 and its higher value may lead to data overfitting.
b) *max_leaf_nodes*: Maximum number of terminal nodes or leaves in a decision tree is denoted by this parameter. For binary trees, a depth of 'n' would produce a maximum of 2^n number of leaves.
c) *nrounds*: It is the number of trees that the model should have. Its value can be set based on intuition or with the help of fine hyperparametric tuning.
d) *gamma*:Gamma specifies the minimum loss reduction required to perform the split while making the algorithm more conservative. The corresponding default value is 0. If there is a positive reduction in the loss function, then only a node is split.
e) *max_delta_step*: In maximum delta step, the weight of each tree is estimated. Its zero value represents no restriction in weight estimation. Setting it to a positive value makes the upgrade step more conservative. Although specifying its value is not mandatory, but it may help in optimal binary regression when the classes are extremely unbalanced.
f) *eta*: Having a default value of 0.3, it basically controls the learning rate. It scales the contribution of each tree by a given factor. When its value is low, the model would become more robust to overfitting, but at the same time, the convergence time would increase.
g) *early_stopping_rounds*: This parameter denotes the number of rounds the model should be run before stopping without having any improvement. The improvement is measured with respect to root mean squared value. Although, the value of early stopping round depends on the designer, it is usually set as 3 or 5.
h) *max_depth*: It represents the greatest depth to which a tree can grow. If its value is set too low, the model would not be able to learn important features, resulting in underfitting. On the other hand, in case of its higher value, the model would learn relationships that are particularly specific to that dataset, resulting in overfitting. Its value is set to 6 as default.
i) *subsample*: It denotes the score of an observation that would be a random sample of each tree. Its typical value is 1. A higher value of subsample would make the algorithm more conservative and prevent overfitting, but its lower value would lead to underfitting.

Learning task parameters

a) *objective*:It defines the objective function of XGBoost model which may be either regression or classification.
b) *eval_metric*: It is employed to validate predictive performance of a model. Among different types of evaluation metrics, like root mean squared error (*RMSE*), mean absolute error (*MAE*), *logloss*, *error*, *merror*, *mlogloss* and *auc*, *RMSE* and *error* are respectively considered as the default metrics for regression and classification.
c) *seed*: This is a random number used when reproducibility is needed.

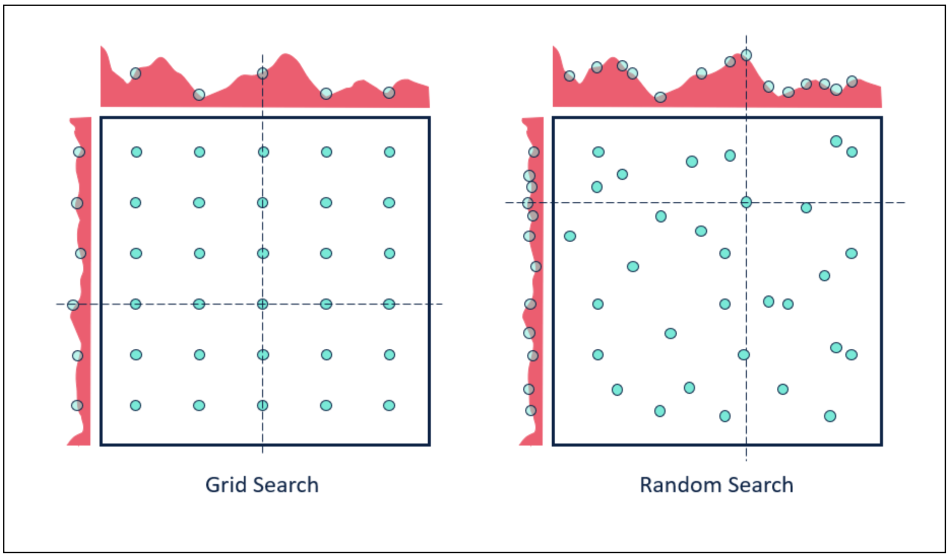


Figure 2. Grid search and random search

The test error of a statistical learning algorithm has two components, i.e. bias and variance. Bias is the error induced in the model due to simplification of different model assumptions. It can be stated as the difference between average prediction of the developed model and actual value that it is attempting to predict. A highly biased model pays less attention to the training data and oversimplifies the model. It always leads to higher errors in training and test data. Variance is the error induced by randomness of the training data. High variance models pay close attention to the training data without generalizing the data. Therefore, these models perform very well on training data, but may have a high error rate on test data. The trade-off between bias and variance is determined by the complexity of the model and amount of training data (James et al., 2013). Optimal selection of hyperparameters in a model helps to avoid both overfitting and underfitting of data. These hyperparameters are learned from the data and are tuned properly to achieve best fit of the data. As searching values of these hyperparameters is a tedious and time-consuming process, grid search and random search techniques are applied to resolve this problem, as shown in Figure 2.

In grid search, a possible set of values is considered for testing and the model is run on all these values, followed by subsequent evaluation of different statistical metrics, like *MAE*, *RMSE* etc. to check predictive accuracy of the considered model. The combination of hyperparameters yielding lowest error value would be selected. In random search, instead of providing the model with all possible values of the hyperparameters, their statistical distributions are considered. The model would then randomly sample values from those distributions and employ them for training. The subsequent steps of random search are quite similar to grid search. Random search is comparatively quicker because only a subset of features is considered for hyperparameter tuning. But, as grid search covers all possible combinations of the hyperparameters, it can yield more accurate model. In this paper, grid search technique is employed for hyperparameter optimization. Among all the huperparameters, *nrounds*, *eta* and *max_depth* are considered here for tuning as they are noticed to be more sensitive influencing the derived solutions (James et al., 2013). The other hyperparameters are set at their default values. The corresponding grid is now developed with the following configuration of the considered hyperparameters, as shown in Table 4.

Table 4. Grid search configuration

Hyperparameter	Lowest value	Highest value	Step size	Number of combinations
nrounds	1	1000	10	100
eta	0.2	1	0.02	41
max_depth	1	6	1	6

It can be noticed from Table 4 that a total of $100 \times 41 \times 6 = 24600$ possible combinations of the hyperparameters is generated from this configuration. Now, the model is run 24600 times twice, each for MRR and Ra, to determine the optimal values of the considered hyperparameters based on the minimum *RMSE* values. The optimal values of those hyperparameters for MRR and Ra are provided in Table 5.

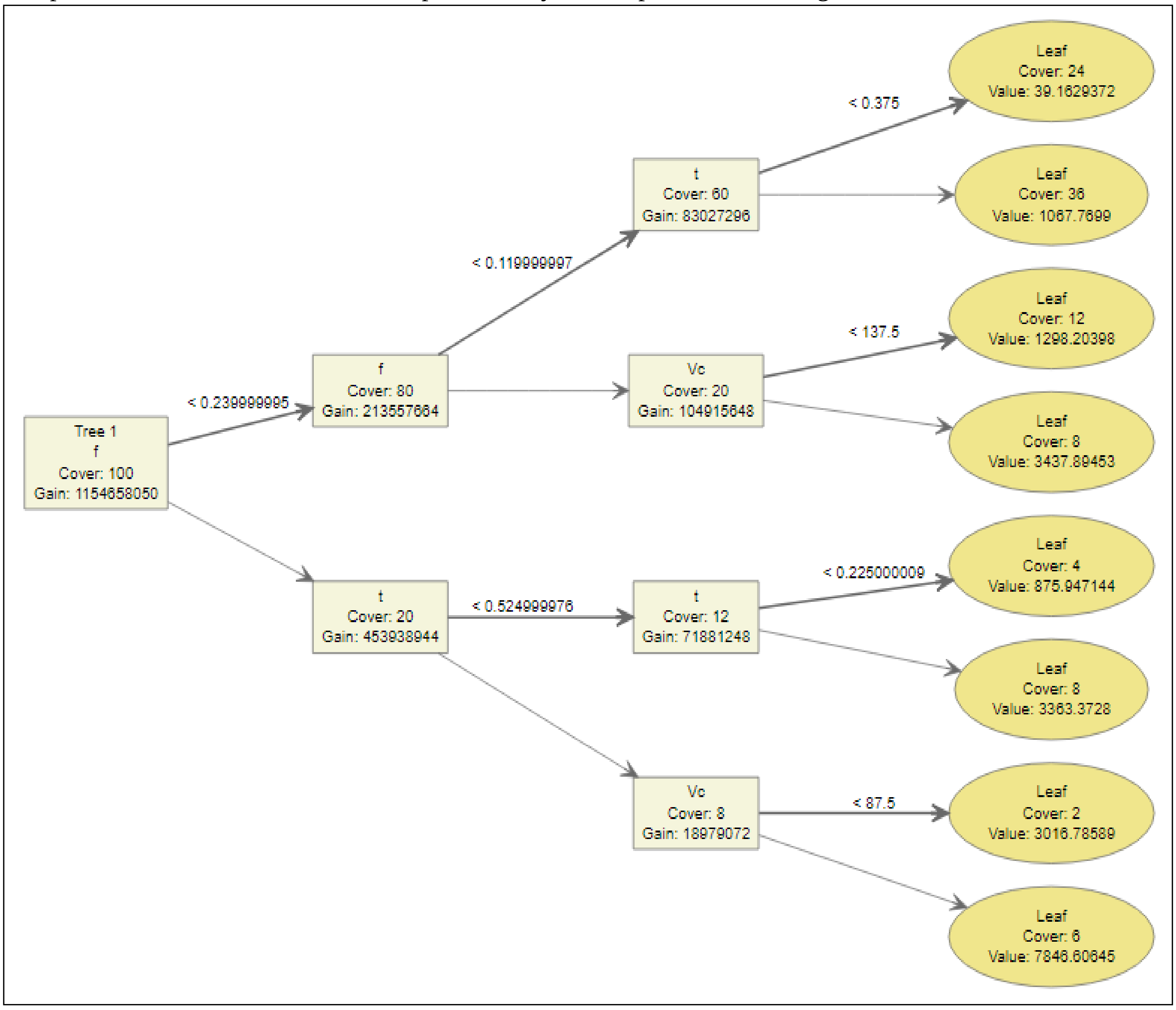
Table 5. Optimal values of the hyperparameters

Hyperparameter	MRR	Ra
nrounds	61	71
Eta	0.64	0.2
max_depth	3	6

A sample decision tree obtained using XGBoost algorithm for MRR is depicted in Figure 3. Similar type of sample decision tree is also developed for Ra, but not included in this paper due to paucity of space and illegibility. Between these two sample decision trees, the decision tree for Ra is longer and more complex for its higher value of*max_depth* (6 as compared to 3 for MRR). The gain value in Figure 3 refers to the relative contribution of a feature to the model and can be mathematically expressed as follows:

Gain = Loss (Parent instance) – (Loss (Upper branch) + Loss (Lower branch)) (1)

The loss value would decrease if the model is successfully trained, resulting in decrement of the gain value in the subsequent branch. There are also two other metrics for XGBoost algorithm, i.e. cover and frequency. The cover metric signifies the relative number of observations related to a particular feature, whereas, frequency represents the percentage of occurrence of a particular feature in the tree. The values of gain, cover and frequency are provided in Table 6 for both the responses of the CNC turning process. On the other hand, Table 7 presents values of MRR and Ra as predicted by the adopted XGBoost algorithm.



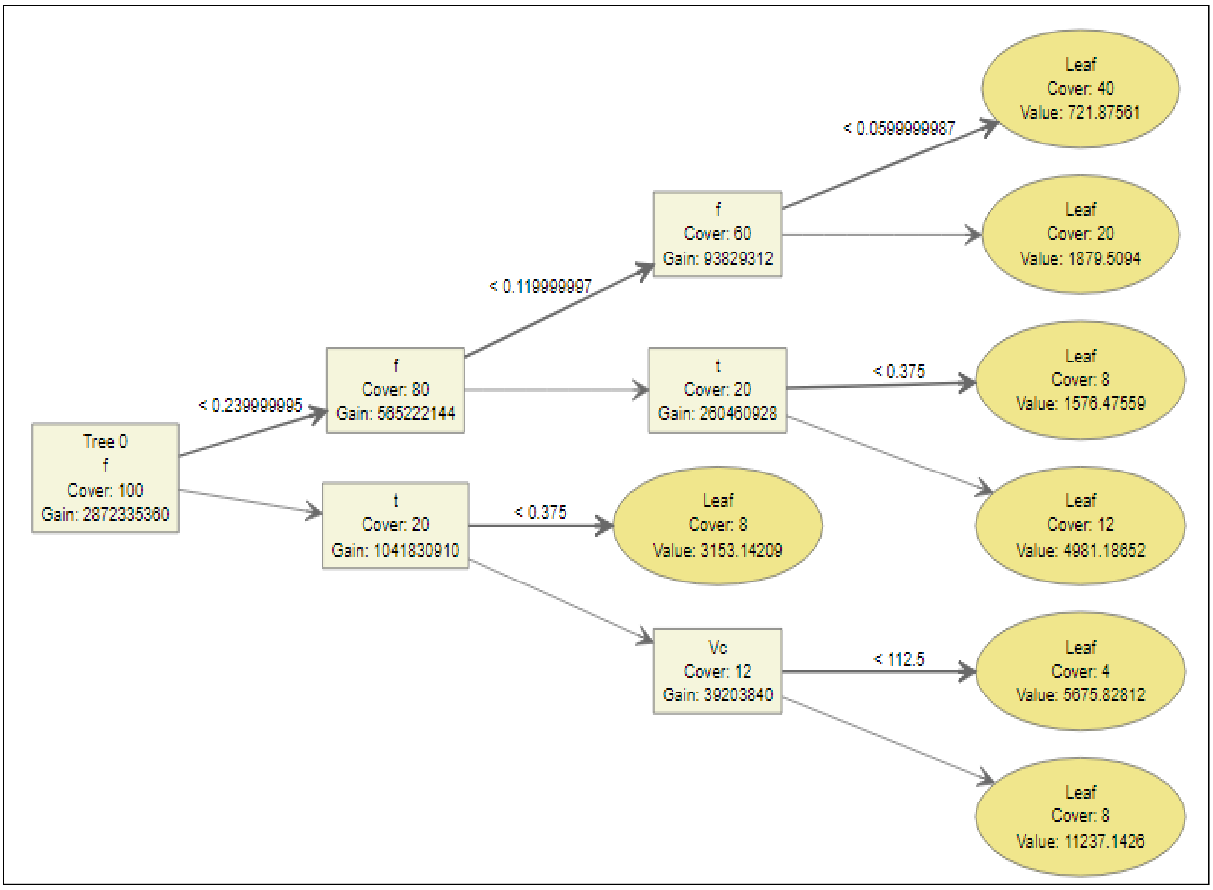


Figure 3. Sample trees for MRR

Table 6. Values of gain, cover and frequency for the responses

Feature	MRR			Ra		
	Gain	Cover	Frequency	Gain	Cover	Frequency
f	0.658103	0.359911843	0.35111513	0.983251955	0.428608633	0.269946809
t	0.309131	0.321772416	0.306811332	0.009602583	0.293931897	0.402925532
Vc	0.032766	0.318315741	0.342073538	0.007145462	0.27745947	0.32712766

Table 7. Predicted values of Ra and MRR

Parameter			Ra		MRR	
Vc	t	f	Actual	Predicted	Actual	Predicted
75	0.45	0.020	0.182	0.180269	675	774.8272
75	0.45	0.040	0.389	0.400073	1350	1363.472
75	0.45	0.080	0.798	0.810117	2700	3457.42
75	0.45	0.160	1.699	1.752465	5400	6078.112
75	0.45	0.320	3.386	3.482308	10800	13170.44
100	0.75	0.020	0.416	0.413105	1500	1761.1
100	0.75	0.040	0.633	0.618138	3000	2833.533
100	0.75	0.080	1.057	1.041479	6000	5781.07
100	0.75	0.160	1.916	1.921121	12000	11869.81
100	0.75	0.320	3.891	3.887996	24000	23521.51
125	0.60	0.020	0.292	0.304922	1500	1303.875
125	0.60	0.040	0.558	0.578894	3000	2808.375

Parameter			Ra		MRR	
Vc	t	f	Actual	Predicted	Actual	Predicted
125	0.60	0.080	1.005	1.022298	6000	5387.497
125	0.60	0.160	2.028	2.023321	12000	11237.71
125	0.60	0.320	4.030	4.040278	24000	24654.71
150	0.15	0.020	0.123	0.124682	450	700.1729
150	0.15	0.040	0.254	0.272165	900	1184.37
150	0.15	0.080	0.783	0.82308	1800	2367.317
150	0.15	0.160	1.698	1.79762	3600	4056.622
150	0.15	0.320	3.745	3.901673	7200	8011.993
175	0.30	0.020	0.195	0.192573	1050	1157.703
175	0.30	0.040	0.471	0.454538	2100	2391.617
175	0.30	0.080	0.945	0.928982	4200	3688.161
175	0.30	0.160	2.106	2.093483	8400	7564.548
175	0.30	0.320	4.239	4.184501	16800	14883.51

In order to validate the prediction accuracy of the XGBoost algorithm for this CNC turning process, five different statistical error estimators, i.e. mean absolute percentage error (*MAPE*), root mean squared percentage error (*RMSPE*), root mean squared logarithmic error (*RMSLE*), correlation coefficient (*R*) and root relative squared error (*RRSE*) are considered here. The mathematical expressions of these metrics are provided as below (Bhattacharya et al., 2021):

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{A_i - P_i}{A_i}\right| \times 100 \tag{2}$$

$$RMSPE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{A_i - P_i}{A_i}\right)^2} \times 100 \tag{3}$$

$$RMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(P_i+1) - \log(A_i+1))^2} \tag{4}$$

$$R = \frac{\sum_{i=1}^{n}(A_i - \overline{A})(P - \overline{P})}{\sqrt{\sum_{i=1}^{n}(A - \overline{A})^2 (P - \overline{P})^2}} \tag{5}$$

$$RRSE = \sqrt{\frac{\sum_{i=1}^{n}(P_i - A_i)^2}{\sum_{i=1}^{n}(A_i - \overline{A})^2}} \tag{6}$$

where A_i and P_i are the actual and predicted responses respectively, $\overline{A}$ and $\overline{P}$ are the means of all the actual and predicted responses respectively, and n is the number of observations in the test dataset. Based on the above-mentioned formulations, the corresponding statistical metrics are calculated, as shown in Table 8. Among them, lower values of *MAPE*, *RMSPE*, *RMSLE* and *RRSE* are always preferred, whereas, higher value of *R* is recommended for validating the performance of any of the prediction tools. Excellent values of all the considered statistical metrics strongly prove the efficacy and potentiality of XGBoost algorithm in almost accurately envisaging the response values of the said CNC turning operation.

Table 8. Statistical metrics for Ra and MRR

Statistical metric	Ra	MRR
MAPE	2.34	13.88
RMSPE	3	18.33
RMSLE	0.014	0.16
R	0.99	0.99
RRSE	0.035	0.115

4. Conclusions

To accurately envisage response values in any of the machining processes, selection of the most appropriate machine learning technique in the form of an effective predictive tool plays an important role. In this paper, XGBoost algorithm is employed for predicting two responses, i.e. MRR and Ra of a CNC turning process with cutting speed, depth of cut and feed rate as the input parameters. Optimal values of different hyperparameters for implementation of this algorithm are selected with the help of grid search method. Excellent values of all the considered statistical metrics prove the efficacy and higher prediction accuracy of XGBoost algorithm for the said machining process. The potentiality of this model can be further extended while considering other categorical variables depicting varying operational conditions of a CNC turning process, like operator’s skill level, types of the work material and cutting tool, coolant type etc. In the similar direction, the appropriateness of XGBoost algorithm can be further validated with the help of experimental datasets from CNC milling and different non-traditional machining processes.

References

Abbas, A. T., Alata, M., Ragab, A. E., El Rayes, M. M. & El Danaf, E. A. (2017). 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, Article ID 2759020, 12 pages, https://doi.org/10.1155/2017/2759020.

Alajmi, M. S., & Almeshal, A. M. (2020).Predicting the tool wear of a drilling process using novel machine learning XGBoost-SDA. Materials, 13(21), https://doi.org/10.3390/ma13214952.

Bhattacharya, S., Das, P.P., Chatterjee, P. & Chakraborty, S. (2021). Prediction of reponses in a sustainable dry turning operation: A comparative analysis. Mathematical Problems in Engineering, Article ID 9967970, 15 pages, https://doi.org/10.1155/2021/9967970.

Brown, G. (2011). Ensemble learning.In: Sammut, C., Webb, G.I. (eds) Encyclopaedia of Machine Learning.Springer, Boston, MA.

Chen, K., Chen, H., Liu, L. Chen, S. (2019). Prediction of weld bead geometry of MAG welding based on XGBoost algorithm. International Journal of Advanced Manufacturing Technology, 101(9), 2283-2295.

Chen, T. & Guestrin, C. (2016).XGBoost. In: Proceedings of 22nd International Conference on Knowledge Discovery and Data Mining, California, USA, 785-794.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y. & Cho, H. (2015). Xgboost: Extreme gradient boosting. R Package Version 0.4-2, 1(4), 1-4.

Choi, D. K. (2019). Data-driven materials modeling with XGBoost algorithm and statistical inference analysis for prediction of fatigue strength of steels.International Journal of Precision Engineering and Manufacturing, 20(1), 129-138.

Deng, J., Xu, Y., Zuo, Z., Hou, Z. & Chen, S. (2019). Bead geometry prediction for multi-layer and multi-bead wire and arc additive manufacturing based on XGBoost. In: Transactions on Intelligent Welding Manufacturing, Springer, Singapore, 125-135.

Dorogush, A. V., Ershov, V. &Gulin, A. (2018). CatBoost: Gradient boosting with categorical features support. In: Proceedings of 32nd Conference on Neural Information Processing Systems, Montréal, Canada, 252-257.

Friedman, J. (2001). Greedy function approximation.A gradient boosting machine.Annals of Statistics, 29(5), 1189-1232.

Friedman, J. H. & Popescu, B. E. (2003). Importance sampled learning ensembles. Journal of Machine Learning Research, 94305, 1-32.

Friedman, J. H. (2002). Stochastic gradient boosting. Computational Statistics & Data Analysis, 38(4), 367-378.

Gao, K., Chen, H., Zhang, X., Ren, X., Chen, J. Chen, X. (2019). A novel material removal prediction method based on acoustic sensing and ensemble XGBoost learning algorithm for robotic belt grinding of Inconel 718. International Journal of Advanced Manufacturing Technology, 105(1), 217-232.

Han, J. H. & Chi, S. Y. (2016).Consideration of manufacturing data to apply machine learning methods for predictive manufacturing. In: Proceedings of 8thInternational Conference on Ubiquitous and Future Networks, Austria, 109-113.

Haynes, G. (2018). Lathe Machine Basics: Introduction to Turning Operations. Cyber Press.

James, G., Witten, D., Hastie, T. &Tibshirani, R. (2013).An Introduction to Statistical Learning with Applications in R. Springer, New York, USA.

Kiangala, S. K. & Wang, Z. (2021). An effective adaptive customization framework for small manufacturing plants using extreme gradient boosting-XGBoost and random forest ensemble learning algorithms in an Industry 4.0 environment. Machine Learning with Applications, 4, 100024.

Kittler, J. & Roli, F. (2003). Multiple classifier systems.In: Lecture Notes in Computer Science, Kittler, J. & Roli, F. (eds.) Springer.

Lan, T. S. & Wang, M. Y. (2009). Competitive parameter optimization of multi-quality CNC turning.International Journal of Advanced Manufacturing Technology, 41(7), 820-826.

Oza, N. C. & Russell, S. J. (2001). Online bagging and boosting. In: International Workshop on Artificial Intelligence and Statistics, Florida, USA, 229-236.

Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81-106.

Schapire, R. E. (2013). Explaining adaboost. In: Empirical Inference. Springer, Heidelberg, 37-52.

Song, K., Yan, F., Ding, T., Gao, L. & Lu, S. (2020). A steel property optimization model based on the XGBoost algorithm and improved PSO. Computational Materials Science, 174, 109472.\

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L.D.A., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J. and Kuhn, M. (2019). Welcome to the Tidyverse. Journal of Open Source Software, 4(43), https://doi.org/10.21105/joss.01686.

Zhang, Z., Huang, Y., Qin, R., Ren, W. & Wen, G. (2021). XGBoost-based on-line prediction of seam tensile strength for Al-Li alloy in laser welding: Experiment study and modelling. Journal of Manufacturing Processes, 64, 30-44.

Zhou, Z. H. (2009). Ensemble learning.Encyclopaedia of Biometrics, 1, 270-273.