INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

IJARIIT

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 9, Issue 5 - V9I5-1153)
Available online at: https://www.ijariit.com

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

Lalitkishore. N
lalitkishore.20me@kct.ac.in
Kumaraguru College of Technology,
Coimbatore, Tamil Nadu

Shriraam Manoharan
shriraam.manoharan@gmail.com
University of Texas -Alumini
Arlington, USA

ABSTRACT

In this paper, an ensemble learning method, in the form of a Categorical boost (Catboost) algorithm is adopted as an effective predictive tool for envisaging values of average surface roughness and material removal rate during CNC turning operation of C45 steel workpiece with a tungsten carbide cutting tool. In order to develop the related models, a grid with 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 root mean squared percentage error. It can be observed that Catboost can be efficiently applied as a predictive tool with excellent accuracy in machining processes.

Keywords: *Catboost, LSTM, Material removal rate, Root Mean Square Error, Root Mean Squared Percentage Error (RMSPE)*

1. INTRODUCTION

At some point throughout the manufacturing process, CNC machine tools are recommended for the accurate machining of all metal components. The turning operations machining parameters depend on the machine tool, the material, the tool life, and the operator's effectiveness. Only theoretical investigations benefit from the selection of machining variables based on the operator's expertise and manuals. Good surface roughness and short machining times are the goals of the turning process, yet they are incompatible. How these objectives balance out depends on the specific selection of cutting speed, feed, and depth of cut. In order to investigate the complex interactions between the machining parameters, experiments were conducted in a CNC lathe to evaluate performance metrics such surface roughness and machining time. Cemal Cakir et al.proposed a technique for figuring out the amount of machining required for turning operations with the lowest production cost as the goal.[1] Lee et al. established a relationship between the cutting speed, feed, and depth of cut with the surface roughness, cutting force, and tool life using an adaptive modeling method.[2] S Chakraborty et al. presents the key algorithmic techniques behind Catboost, a new gradient boosting toolkit. Their combination leads to Catboost, a new gradient boosting toolkit. Their combination leads to outperforming other publicly available boosting implementations in terms of quality on a variety of datasets.[3]. To get improved surface quality and surface integrity that is comparable to that produced by grinding, employ modest feed rates, fine depths of cut, and suitable cutting instruments under dry circumstances.[4] The multi-response optimization is crucial in industrial applications. It is superior to single-response technology optimization because all components are affected simultaneously by all input factors. To optimize the turning process parameters on a CNC lathe with surface roughness, cutting forces, and MRR as multi-performance characteristics, the Taguchi approach with GRA (Grey Relational Analysis) is utilized. It has been utilized successfully to create high-quality products at minimal cost in the fields of automotive, aerospace, etc [5] Goel et al. established an effective way for improving the Taguchi with GRA-based HSLA steel slab milling process for multi-function features.[6] Siddiquee et al used the Taguchi approach to conduct the tests on AISI 321 steel in order to optimize the deep drilling process parameters.[7] CatBoost employs a more successful method. It is based on the ordering principle, which is the paper's main concept, and is motivated by online learning algorithms that obtain training samples in a sequential sequence over time.[8,11] Gradient boosting is a powerful machine-learning technique which produces cutting-edge outcomes in a range of real-world activities. It has long been the go-to technique for learning issues involving diverse characteristics, noisy data, and intricate connections, such as online search, recommendation systems, weather forecasting, and many more.[9,10]

2. EXPERIMENTAL SETUP

A 2-axis CNC lathe with a spindle rated at 7 Kw, 2800 rpm was used for the experiments. A tungsten carbide cutting tool is used to perform the turning operation on a workpiece made of C45 steel. Various speeds between 00 and 2000 rpm with an increment of 200 rpm and feed rates of 0.1, 0.2, 0.3, and 0.4 mm/rev were used in the experiments, with a constant depth of cut of 1.5 mm. Material removal rate, surface roughness, and tool life were the response variables in an experiment with a three-level, two-factor factorial and three center points. The process variables and the experimental setup are given in Table 1. The machining parameters taken into account during the design of the experiment are shown in Table 2, along with the associated material removal rate, tool life, and surface roughness that were achieved for the input values.

Table 1 Process Parameters and Experimental Conditions

	Level /Factors	High	Medium	Low
A	Cutting speed (m/min)	5233	3663	1570
B	Feed rate (mm/rev)	0.1-0.4		
C	Depth of (mm)	1.5		

Table 2 Experimental design and Results

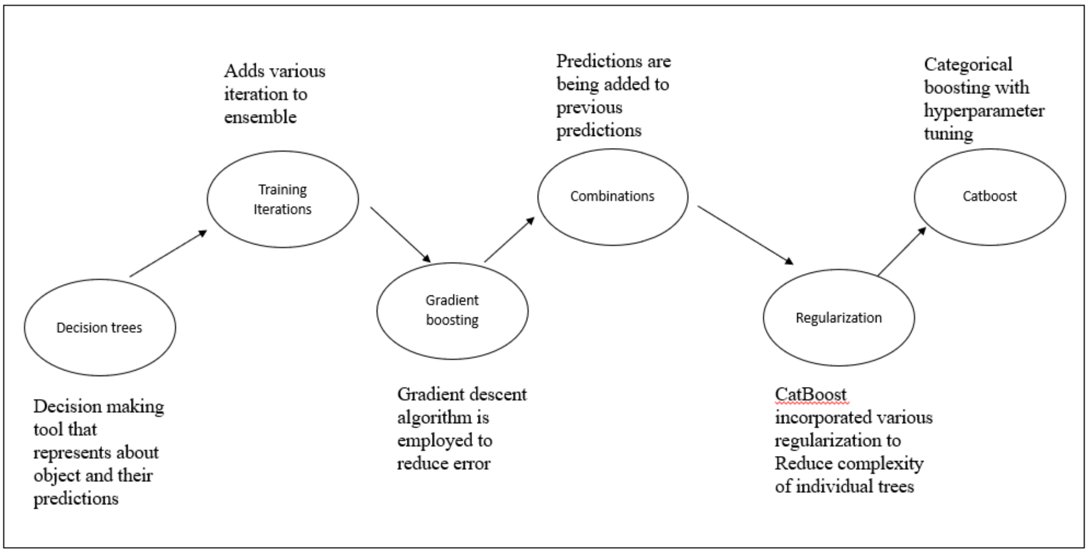
Expt.No	Cutting speed, (m/min)	Feed rate, (mm/rev)	Tool life (min)	Total material removed(cm^3)	Surface Roughness (μm)
1	1570	0.1	14.41	3393.555	0.0125
2	1570	0.2	14.4	6782.4	0.05
3	1570	0.3	14.46	10215.99	0.1125
4	1570	0.4	14.46	13621.32	0.2
5	2093	0.1	14.45	4537.3	0.0125
6	2093	0.2	14.44	9068.32	0.05
7	2093	0.3	14.46	13621.32	0.1125
8	2093	0.4	14.45	18149.2	0.2
9	2617	0.1	14.46	5675.55	0.0125
10	2617	0.2	14.46	11351.1	0.05
11	2617	0.2	14.55	11421.75	0.05
12	2617	0.2	14.6	11461	0.05
13	3140	0.2	14.44	13602.48	0.05
14	3140	0.2	14.33	13498.86	0.05
15	3140	0.2	14.66	13809.72	0.05
16	3140	0.2	14.56	13715.52	0.05
17	3663	0.2	14.56	16001.44	0.05
18	3663	0.2	14.63	16078.37	0.05
19	3663	0.2	14.44	15869.56	0.05
20	3663	0.2	14.56	16001.44	0.05
21	4187	0.2	15.56	19543.36	0.05

22	4187	0.2	16.56	20799.36	0.05
23	4187	0.2	17.56	22055.36	0.05
24	4187	0.2	18.56	23311.36	0.05
25	4710	0.2	19.56	27638.28	0.05
26	4710	0.2	20.56	29051.28	0.05
27	4710	0.2	21.56	30464.28	0.05
28	4710	0.2	22.56	31877.28	0.05
29	5233	0.2	23.56	36989.2	0.05
30	5233	0.2	24.56	38559.2	0.05
31	5233	0.2	25.56	40129.2	0.05
32	5233	0.2	26.56	41699.2	0.05

3. CATBOOST AS THE PREDICTIVE TOOL

When it comes to structured data, CatBoost, an advanced gradient boosting technique, is ideally suited to address a number of machine learning and predictive modeling difficulties. Statistical techniques are used in machine learning, a branch of artificial intelligence, to teach robots how to imitate human behavior. As a result, the objective of machine learning algorithms is to more accurately generalize existing problems. The designers must develop a variety of learners, some of which may frequently become quite unreliable because to the randomness in the data. CatBoost's gradient boosting ensemble generates decision trees. Every time the CatBoost algorithm boosts, a new decision tree is created, and these trees are then integrated to create a powerful predictive model. The decision tree used in Catboost is represented in Fig 1.

Fig 1. Evolution of CatBoost algorithm

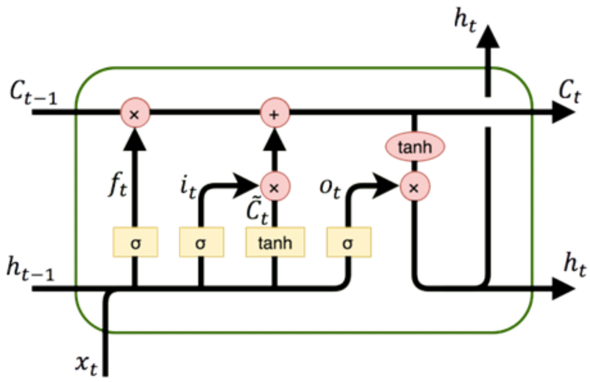


The study employs a two-step regression methodology to achieve precise predictions for crucial machining parameters—tool life, total material removed, and surface roughness.

The equation of the two-step regression is $\hat{y}_1 = \beta_0 + \beta_1X_1 + \beta_2X_2 + ... + \beta_pX_p$, where $\hat{y}_1$ is the predicted value of the dependent variable in the first stage, β_0 is the intercept, β_1 to β_p are the regression coefficients for the independent variables X_1 to X_p. In the first step, Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks are used to predict tool life, capturing complex temporal relationships within the data. The LSTM model we implemented in your code is used for predicting "Tool Life" based on the input features. The LSTM model can be described mathematically as follows: Input Sequence: Let X_t be the input feature vector at time step *t*. You have multiple input features in your code so X_t is a vector of features at each time step *t*. LSTM Cell Operations each LSTM cell processes the input X_t and maintains its internal hidden state and cell state. At each time step *t*, the LSTM cell performs the following operations (i) Forget Gate: It decides what information from the previous cell state should be thrown away or kept. Let f_t represent the forget gate value at time *t.* (ii) Input Gate: It updates the cell state with new information. Let i_t represent the input gate value at time *t.* (iii)Candidate Cell State: It calculates a new candidate cell state $\tilde{C}_t$ that could be added to the cell state. This is based on the current input X_t and the previous hidden state. (iv) Cell State Update: It updates the cell state $\tilde{C}_t$ using the forget gate, input gate, and the candidate cell state. (v) Output Gate: It decides what the next hidden state should be. Let o_t represent the output gate value at time *t.*
In Output Layer: After processing the entire sequence of inputs, you have a final hidden state h_t from the last LSTM cell. You then pass this hidden state through a Dense layer with a single unit (your output layer) to obtain the predicted "Tool Life" value. So,

mathematically, the LSTM operations at each time step *t* can be summarized as:

$$f_t = \sigma\left(W_f.[h_{t-1}, X_t] + b_f\right)$$
$$i_t = \sigma\left(W_{iW_f}.[h_{t-1}, X_t] + b_i\right)$$
$$\tilde{C}_t = \tanh(W_c.[h_{t-1}, X_t] + b_c)$$
$$C_t = f_t.C_{t-1} + i_t.\tilde{C}$$
$$o_t = \sigma(W_o.[h_{t-1}, X_t] + b_o)$$
$$h_t = o_t.\tanh(C_t)$$



]
Fig 2. Long Short-Term Memory (LSTM) neural networks [12

Where W_f, W_f, W_c, W_o are weight matrices for the forget gate, input gate, candidate cell state, and output gate, respectively.b_f, b_i, b_c *and* b_o are bias terms for the respective gates.σ is the sigmoid activation function.

$tanh$ is the hyperbolic tangent activation function.h_{t-1}, X_t represents the concatenation of the previous hidden state h_{t-1} and the current input X_t. The final predicted "Tool Life" value is obtained by passing through h_t the output Dense layer. This mathematical representation captures the operations performed by the LSTM model in your code to predict "Tool Life" based on the input features. The LSTM layer in code has 50 units, which means there are 50 LSTM cells in this layer.

In the second stage of two-step regression, the predicted values from the first stage are used as independent variables in a new regression model to predict the dependent variable, then equation is $\hat{y}_2 = \hat{y}_1 + \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p$, where $\hat{y}_2$ is the predicted value of the dependent variable in the second stage.

LSTM-predicted tool life values are incorporated into a CatBoost regression model in the second step, which leverages Bayesian hyperparameter optimization and L2 regularization for fine-tuning, leading to significantly improved prediction accuracy. CatBoost is an ensemble method based on gradient boosting with decision trees. While the mathematical equations for CatBoost are complex due to the ensemble nature of the algorithm and the interaction with categorical features, A simplified representation of CatBoost combines decision trees and updates predictions.

The core equation for CatBoost can be summarized as follows:

$$\hat{y}_i = \sum_{t=1}^{N} f_t(X_i)$$

Where: $\hat{y}_i$ represents the predicted target value for sample *i*. *N* is the total number of trees in the ensemble. $f_t(X_i)$ represents the prediction of the *t*-th decision tree for the input features X_i. CatBoost builds an ensemble of decision trees, typically shallow trees with limited depth. These trees are constructed sequentially, and each tree aims to correct the errors of the previous ones. Each decision tree in the ensemble, denoted by $f_t(X_i)$, predicts the target value for a given input X_i.

For a dataset with *N* samples and a single decision tree, the update of the prediction $f_t(X_i)$ after the *t*-th tree is

$$f_t(X_i) = f_{t-1}(X_i) + \sum_{T}^{N} \eta.\omega_i \; h_t(X_i)$$

Where $f_t(X_i)$ is the updated prediction after the *t*-th tree. $f_{t-1}(X_i)$ is the previous prediction (output of the previous trees). η is the learning rate (shrinkage). ω_i is the contribution of the *i*-th sample to the gradient of the loss function. $h_t(X_i)$ is the prediction of the *t*-th tree for the *i*-th sample.

The contribution ω_i is calculated based on how the *t*-th tree affects the gradient of the loss function. It depends on the loss function used (e.g., mean squared error for regression, cross-entropy for classification) and can be more complex in practice. These individual tree predictions are typically a real number (for regression tasks) or a probability (for classification tasks). The final prediction $\hat{y}_i$ for a sample *i* is obtained by combining the predictions of all the decision trees in the ensemble. This combination can involve simple averaging or weighted averaging, depending on the problem and hyperparameters.

The results demonstrate that this approach consistently yields Root Mean Squared Errors (RMSE) below 1 for all parameters, showcasing the effectiveness of combining deep learning and gradient boosting techniques with advanced hyperparameter tuning in machining predictions, offering valuable insights and applications in the manufacturing industry. A broad variety of hyperparameters are available in CatBoost that may be adjusted to optimize the gradient boosting models. The performance of the model may be influenced and many training process variables can be controlled. Booster parameters and learning task parameters are the two primary types of parameters, and they are briefly detailed below.

Booster parameters:

iterations (or n- estimators): This parameter sets the number of boosting iterations or the number of trees in the ensemble. Increasing this value may improve the model's performance, but be cautious of overfitting.

***learning_rate (or eta*)**:** Learning rate controls the step size at each iteration while moving towards a minimum of the loss function. Lower values make the learning process more robust but require more iterations.

depth (or max_depth): This parameter specifies the maximum depth of each tree in the ensemble. Deeper trees can capture more complex patterns but may lead to overfitting.

l2_leaf_reg: This is the L2 regularization term on the weights of the leaf nodes. It helps control overfitting by penalizing large weights.

***Subsample*:** Subsample controls the fraction of data used for training each tree. A value less than 1.0 introduces randomness and can help prevent overfitting.

***colsample_bylevel and colsample_bynode*:** These parameters control the fraction of features (columns) used at each level or node of the tree. They add more randomness to the model and can improve generalization.

loss_function: Specifies the loss function used for training. It can be set to different loss functions depending on your regression or classification task.

Learning Parameters:

For Classification Tasks:

***loss_function (default: 'Logloss'*)**:** This parameter specifies the loss function to be used for classification. Common choices include 'Logloss' (logarithmic loss, suitable for binary and multiclass classification) and 'CrossEntropy' (alternative name for 'Logloss').

eval_metric Determines the metric used for evaluating the model's performance during training and early stopping. Common choices include 'Logloss' for binary classification, 'MultiClass' for multiclass classification, and 'AUC' (Area Under the ROC Curve).

custom_metric (default: None): Allows you to define custom evaluation metrics. You can pass a list of custom metric functions to this parameter.

***class_weights (default: None*)**:** If you have imbalanced classes, you can use this parameter to assign different weights to different classes. It helps the model give more importance to minority classes.

For Regression Tasks:

loss_function (default: 'RMSE'): This parameter specifies the loss function for 2. regression tasks. Common choices include 'RMSE' (Root Mean Squared Error) and 'MAE' (Mean Absolute Error).

***eval_metric (default: 'RMSE'*)**:** Determines the metric used for evaluating the model's performance during training and early stopping. Common choices include 'RMSE,' 'MAE,' and 'R2' (Coefficient of Determination).

A statistical learning algorithm's test error is made up of two elements: bias and variance. The bias in a model is the inaccuracy brought about by distinct model assumptions being oversimplified. The difference between the average forecast made by the generated model and the actual value that it is attempting to predict may be used to describe it. A heavily biased model oversimplifies the model and pays little attention to the training set of data. In both training and test data, it always results in larger mistakes. The inaccuracy brought on by the training data's unpredictability is known as variance. High variance models closely scrutinize the training data without making generalizations. In light of this, these models perform admirably on training data but may exhibit significant error rates on test data.

4. CATBOOST AS THE VALIDATORY TOOL

In order to validate the accuracy of the Catboost algorithm for this CNC Turning algorithm statistical error estimators are considered. i.e. Root Mean Squar Error(RMSE), Root Mean Squared Percent Error (RMSPE).

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

$$RMSPE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[\frac{A_i - P_i}{A_i}\right]^2} \times 100$$

where Ai and Pi are the actual and predicted responses respectively. n is the observation in the dataset. y and y_i is the predicted and actual values respectively. Errors are calculated using the above-mentioned formula. Table 3 represents the statistical values of the Material removal Rate , Surface Roughness and tool life.

Based on the above-mentioned formulations, the corresponding statistical metrics are calculated, as shown in Table 4. Among them, lower values of RMSPE, RMSE 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 CatBoost algorithm in almost accurately envisaging the response values of the said CNC turning operation.

TABLE 3 EXPRIMENTAL DATA

S NO	TOOL LIFE		MATERIAL REMOVAL RATE		SURFACE ROUGHNESS	
	Predicted Value	Actual Value	Predicted Value	Actual Value	Predicted Value	Actual Value
1	14.4031	14.41	3393.587907	3393.555	0.01258758	0.0125
2	14.3984	14.4	6782.412637	6782.4	0.04999325	0.05
3	14.5026	14.46	10215.9427	10215.99	0.11237214	0.1125
4	14.5011	14.46	13621.36681	13621.32	0.19993738	0.2
5	14.3964	14.45	4537.280921	4537.3	0.01246115	0.0125
6	14.4040	14.44	9068.287756	9068.32	0.05000493	0.05
7	14.4956	14.46	13621.33405	13621.32	0.1126115	0.1125
8	14.5032	14.45	18149.16589	18149.2	0.19990491	0.2
9	14.4991	14.46	5675.571559	5675.55	0.01258865	0.0125
10	14.5014	14.46	11351.17618	11351.1	0.05022354	0.05

TABLE 4 STATISTICAL METRICS OF Ra AND MRR

STATISTICAL METRIC	Ra	MRR
RMSE	0.058465	0.000116
RMSPE	0.000436	0.284633

*Ra indicates Surface Roughness
*MRR is Total Material Removed

5. CONCLUSION

The selection of the most appropriate machine learning technique in the form of an effective predictive tool is crucial in order to accurately envisage response values in any of the machining processes. In this paper, Catboost algorithm is employed for predicting three responses, i.e., Material Removal Rate and Ra of a CNC turning process with the cutting speed, depth of cut, feed rate as the input parameters. Optimal values of various hyper parameters are considered for implementing Catboost. Excellent values of all the considered statistical metrics prove the efficacy and higher accuracy of Catboost algorithm for the CNC turning process.

6. REFERENCE

[1]. Asokan P, Baskar N, Babu K, Prabhaharan G, Saravanan R (2005) Optimization of surface grinding operation using particle swarm optimization technique. Journal of Manufacturing Science and Engineering, 127:885–892.

[2]. Cakir MC, Gurarda A (1998) Optimization and graphical representation of machining conditions in multi-pass turning operations. Computer Integrated Manufacturing Systems, 11:157–170.

[3]. Chakraborty, Shankar, and Shibaprasad Bhattacharya. "Application of XGBoost algorithm as a predictive tool in a CNC turning process." *Reports in Mechanical Engineering* 2, no. 1 (2021): 190-201.

[4]. Meyer, R., Köhler, J., and Denkena, B. Influence of the Tool Corner Radius on the Tool Wear and Process Forces during Hard Turning. The International Journal of Advanced Manufacturing Technology. 2012, 58: 933-940.

[5]. Singh O P, Kumar G and Kumar M 2019 Role of Taguchi and grey relational method in optimization of machining parameters of different materials: a review Acta Electronica Malaysia (AEM) 3(1) pp19-22

[6]. Goel P, Khan Z A, Siddiquee A N, Kamaruddin S and Gupta, R.K 2012 Influence of slab milling process parameters on surface integrity of HSLA: a multi-performance characteristics optimization The International Journal of Advanced

Manufacturing Technology 61(9-12) pp 859-871.

[7]. Siddiquee A N, Khan Z A, Goel P, Kumar M, Agarwal G and Khan N Z 2014 Optimization of deep drilling process parameters of AISI 321 steel using Taguchi method Procedia Materials Science 6 pp 1217-1225

[8]. L. Bottou and Y. L. Cun. Large scale online learning. In Advances in neural information processing systems, pages 217–224, 2004

[9]. R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms.In Proceedings of the 23rd international conference on Machine learning, pages 161–168. ACM,2006.

[10]. B. P. Roe, H.-J. Yang, J. Zhu, Y. Liu, I. Stancu, and G. McGregor. Boosted decision trees as an alternative to artificial neural networks for particle identification. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 543(2):577–584, 2005.

[11]. Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, Andrey Gulin. “CatBoost: unbiased boosting with categorical features.” arXiv:1706.09516.

[12]. Mohsen Soori, Behrooz Arezoo, Roza Dastres. “Machine learning and artificial intelligence in CNC machine tools, A review”.

