Predicting Factory Equipment Lifespan Through Manufacturing Data Analysis Using AI

[1] Jae-Hak Lee, [2] Young-Han Jeong and [3] Jung Kyu Park
[1,2] Department of Aeronautical & Mechanical Engineering, Changshin University, Korea.
[3] Department of Computer Engineering, Changshin University, Korea.
[1]jhleepaul@cs.ac.kr, [2]moteru01@hanmail.net, [3]jkpark@cs.ac.kr

Correspondence should be addressed to Jung Kyu Park : jkpark@cs.ac.kr

Article Info
Journal of Machine and Computing (http://anapub.co.ke/journals/jmc/jmc.html)
Doi: https://doi.org/10.53759/7669/jmc202404066
Received 21 November 2023; Revised from 25 December 2023; Accepted 10 June 2024
Available online 05 July 2024.

Abstract – Recently, research on applying artificial intelligence (AI) to various industries, especially manufacturing, is being actively conducted. In the field of smart factory, the purpose is to improve productivity based on data generated in the process of producing or processing products. The tool breakage during metal product processing causes fatal difficulties of predicting tool life. Moreover, if tool life is not predicted, defects may occur product reliability deteriorate, which may adversely affect product performance or economic aspects. In this paper, data related to machining is collected from CNC equipment in real time, and through machine learning and deep learning, which factors affect the wear of cutting tools are identified and the lifespan of cutting tools is predicted. An AI-based solution was applied to the system, productivity improved due to an increase in tool life.

Keywords – CNC Machining, Tool Life, Machine Learning, Deep Learning, Life Prediction.

I. INTRODUCTION

Recently, research on the application of artificial intelligence has been actively conducted in various industries, especially manufacturing [1-4]. Among them, the smart factory field aims to improve productivity based on data generated during the production or processing of products [5-7]. Defects caused by tool breakage that occur during processing metal products are one of the most fatal difficulties. However, it is difficult to predict the tool life that causes problems. In order to ensure smooth progress of the CNC process, we pursue stabilization of product quality and customer satisfaction through improvement of process quality distribution based on optimization of facility maintenance management, but there are limitations to existing predictive maintenance methods. To solve this problem, it is necessary to introduce an AI solution that can set optimal working conditions for equipment and predict problems in advance [8-10]. In addition, the biggest problem is the occurrence of defects due to tool breakage that occurs unexpectedly during the production process. However, there is difficulty in predicting tool life, which causes such problems.

Failure to predict tool life will lead to a decrease in customer reliability if defects occur in processing equipment. Additionally, it is urgent to prepare preparatory measures in advance as negative impacts may occur on the management or sales side. Therefore, it is necessary to collect data such as real-time usage time, feed rate, load rate, and tool life information and use AI techniques to establish a foundation for extending and predicting the life of tools [11-13]. In addition, it is necessary to determine whether dimensional tolerances of processed products occur due to temperature and humidity. After that, continuous monitoring is required to connect the temperature management device. Additionally, it is also necessary to determine the expected time for cutting tool replacement, such as the occurrence of vibration due to differences in the amount of cut delivered to the processed product depending on the wear amount of the cutting tool bite of the processing equipment. The improvement requirements resulting from this are as follows. First, in order to reduce defect rates and improve productivity, we utilize currently collected manufacturing data from equipment to develop a solution that can extend the life of tools and predict tool replacement. Second, as a problem-solving obstacle, data related to machining is collected in real time from CNC equipment, but the factors that affect the life of the tool are revealed. Therefore, through this study, we will conduct verification of the AI algorithm by learning manufacturing condition data such as feed rate, load rate (SPINDLE SPEED, SPINDLE LOAD), and tool life information, by detecting changes in these variables and checking how they affect tool life.

II. REALATED WORKS

Convolutional neural networks have come a long way since LeNet. In particular, based on ImageNet's ILSVRC, an image processing competition for image classification, image segmentation, and object detection, a new convolutional neural network architecture won the competition every year, and the winning model became the standard convolutional neural network model for image classification.

AlexNet

AlexNet[14] is an early CNN model in the deep learning era. It is the winning model of the 2012 image classification competition called ILSVRC. The start of deep learning in earnest can be seen as AlexNet in 2012. It consists of a total of 5 convolutional layers and 3 fully connected layers, and was composed of 2 parallel structures due to limitations in memory use at the time. Unlike before, AlexNet used ReLU as the activation function rather than sigmoid or tanh, and used Local Response Normalization (LRN) to normalize the feature map. Additionally, a dropout layer is included to avoid overfitting. **Fig 1** shows the structure of AlexNet.

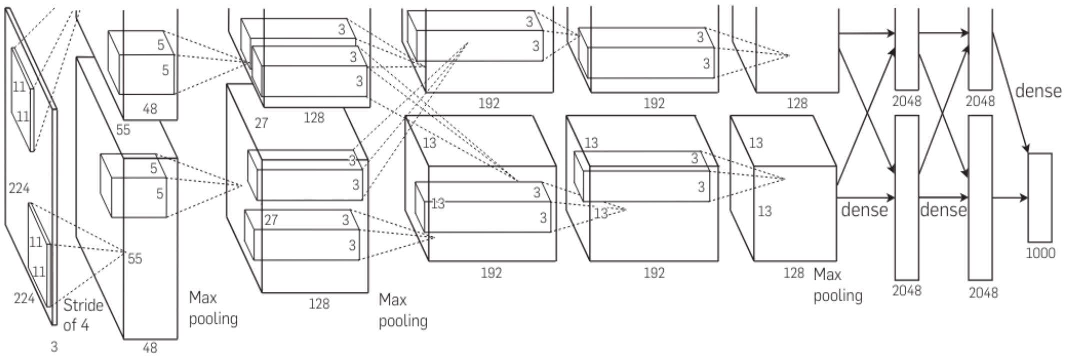


Fig 1. Analysis Using a Microscope.

VGG

VGGNet is a convolutional neural network model developed by the Visual Geometry Group laboratory at Oxford University and is the winning model of ILSVRC 2014 [15]. AlexNet can be said to be the next generation convolutional neural network model, and as the name of the paper suggests, it is a deep learning architecture composed of more convolutional layers. Depending on the type, it is classified into VGG11, VGG13, VGG16, and VGG19, and the number in the model name indicates the number of layers that can be learned in the model.

Table 1 shows several VGG architectures presented in the VGG paper. A represents a convolutional neural network with 11 learnable layers, B with 13, C and D with 16, and E with 19 learnable layers. Here, learnable means a convolutional layer and a fully connected layer with weights to be learned by backpropagation, excluding the pooling layer and softmax. The LRN layer of A-LRN refers to the Local Response Normalization layer, and this layer performs normalization to prevent gradient loss that occurs when the structure of the neural network becomes deeper. In VGG, by presenting the experimental results of A and A-LRN, it was shown that the Local Response Normalization layer is meaningless in improving performance. C and D are composed of the same number of learnable layers of 16, but there is a difference in the filter size of the convolution layer for 1x1 and 3x3 applications.

Table 1. VGG Models

ConvNet Configuration						
Type	A	A-LRN	B	C	D	E
Number of Layers	11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers

ResNet

Normally, the error rate that occurs when humans classify images is considered to be 5%, but ResNet showed an error rate of less than 3.6%, becoming the first learning model to surpass humans. Unlike existing structures, ResNet has connections that skip the convolution layer. This connection is a residual connection, and the input from the previous layer is combined with the result of passing the convolution layer and sent to the next layer. **Fig 2** shows the modules for two residual concatenations. The first module is a general residual concatenation, and the second module adds a 1x1 convolution to reduce the number of parameters to learn like in GoogLeNet.

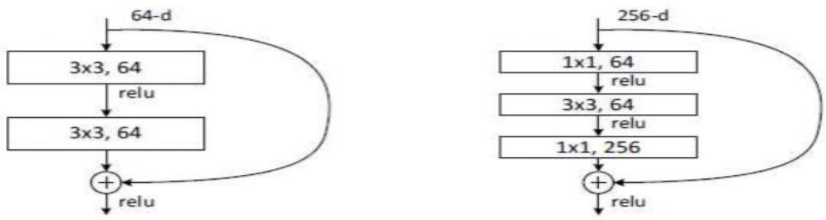


Fig 2. Deeper Bottleneck Architectures.

III. MANUFACTURING DATA ANALYSIS

The method of collecting CNC machining data was to connect the CNC equipment to the Converter program and transmit it to the database in the cloud, collecting data at 100ms intervals. And the data collection period was from November 2, 2022 to December 1, 2022. Learning data was collected, and PoC data was collected for two weeks from December 2. The input variables were composed of X, Y, Z axis, SPINDLE LOAD, SPINDLE SPEED, FEED RATE, and tool count, and the output variable was a data set composed of tool wear rate. At this time, the tool wear rate was photographed through a microscope as shown in **Fig 3**, and in collaboration with field workers, the tool wear size was measured through the microscope every 333, 666, 1000, 1050, and 1100 times. For each tool counting, the tool was taken out of the CNC equipment and tool wear was measured.

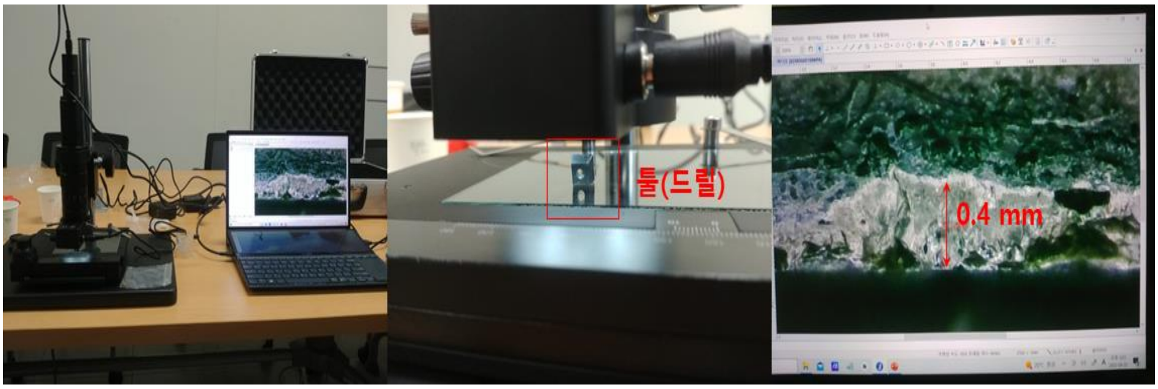


Fig 3. Analysis Using a Microscope.

At this time, in addition to microscopic imaging, the tool wear rate value was completed using the graph in **Fig 4** to complete the tool wear data.

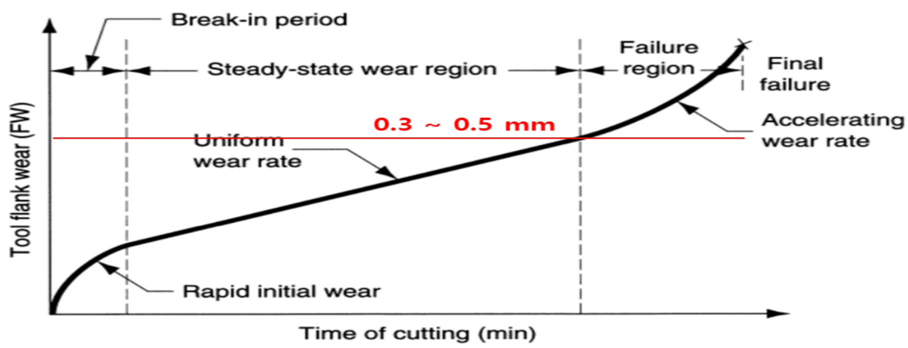
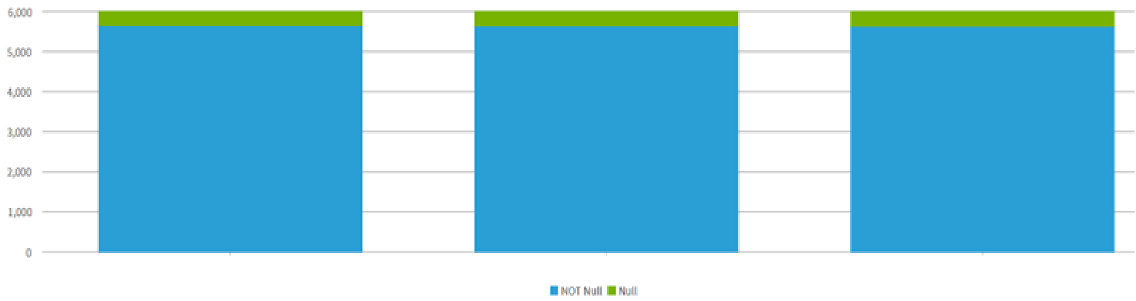
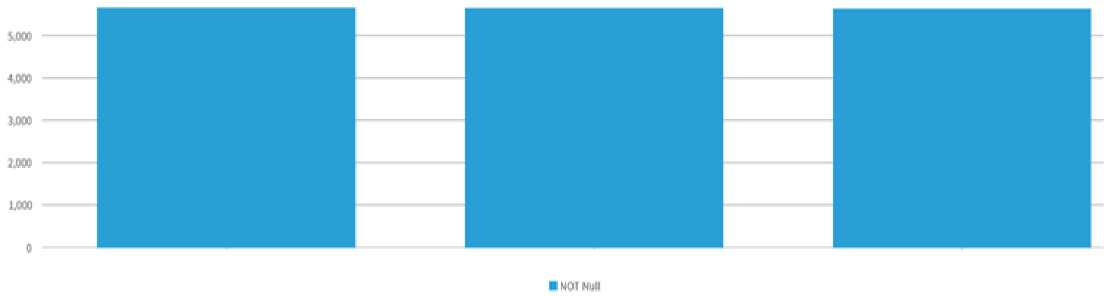


Fig 4. Wear And Replacement Standards According to The Number of Times of Tool Use

Before analyzing the collected CNC machining data, data cleaning and processing were performed. **Fig 5** shows that there were missing values due to network connectivity at the beginning of data collection. All variables with missing values and missing values at the beginning of the connection were removed.



(a) Before Missing Value Removal.



(b) After Missing Value Removal.

Fig 5. Before And After Removing Missing Values.

And since it is natural that tool consumption increases as work time increases, it was judged that a time-based approach could cause overfitting, and unnecessary variables were removed. For example, unnecessary variables such as program number, equipment model name, operation mode, tool life value, and sequence number were removed. After removing unnecessary variables, due to the nature of manufacturing data, the time when product defects occur due to tool damage or wear becomes severe is shorter than the time when normal products are produced, so the problem of data imbalance was solved through the SMOTE method in **Fig 6**.

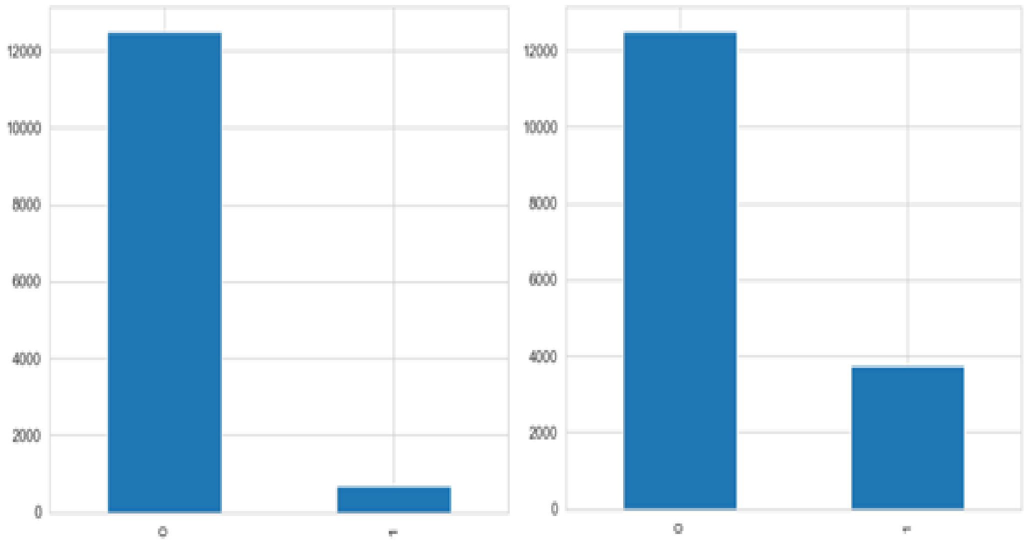


Fig 6. Before SMOTE Application (Left), After Application (Right).

Lastly, to infer tool wear, the backward elimination method among variable selection methods was applied to infer tool wear. The reason why the backward elimination method was chosen was because it has a fast-processing speed. The results of the backward elimination method are as follows **Table 2**.

Table 2. Backward Elimination Results

	If all variables are selected	After removing X, Y, Z axis	Selected Tool count, SPINDLE LOAD
accuracy	95.88	98.5	98.75
precision	52.94	74.47	83.5
recall	51.43	100	91.01
F1-Score	52.17	85.37	87.09

IV. MODEL DESIGN

The model design and verification process were divided into two parts: model design and model verification. First of all, model design involves designing a model using learning data and applying optimal parameters accordingly. Model verification infers the degree of tool wear in PoC data through the built model. At this time, the product is determined to be defective or non-defective depending on the inferred degree of tool wear, and the model is verified until the development of a system that sounds an alarm if a defect occurs. The process of model design and model verification is shown in **Fig 7**.

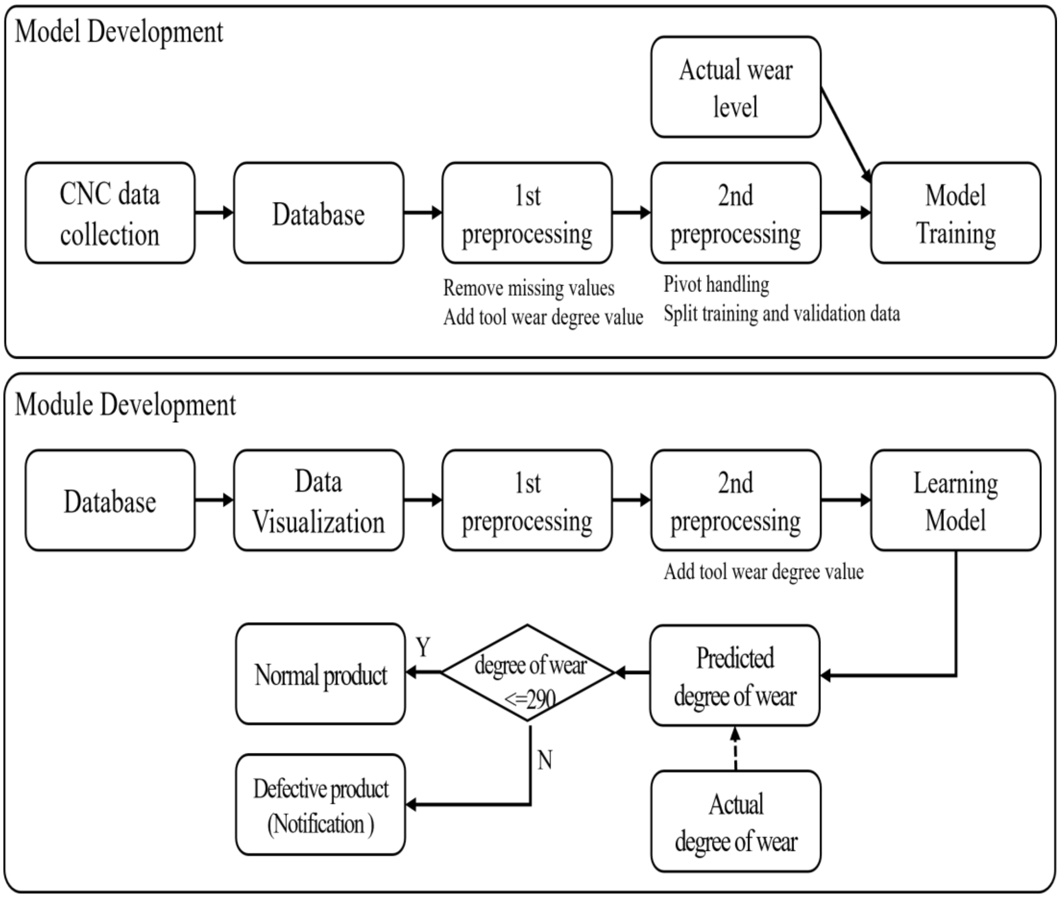


Fig 7. Model Development and Verification Process.

The models to be used for learning are LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit), which have good performance for time series data because CNC data is time series data [16-18]. The LSTM algorithm is a type of short- and long-term memory that complements the shortcomings of existing RNNs. And GRU is a model that improves the structure of the existing LSTM to be simpler. In the case of the existing LSTM, there were three gates: forget gate, input gate, and output gate, but in **Fig 8**, GRU uses only two gates, reset gate and update gate. In addition, the cell state and hidden state are combined to express one hidden state, making it simpler than the existing LSTM.

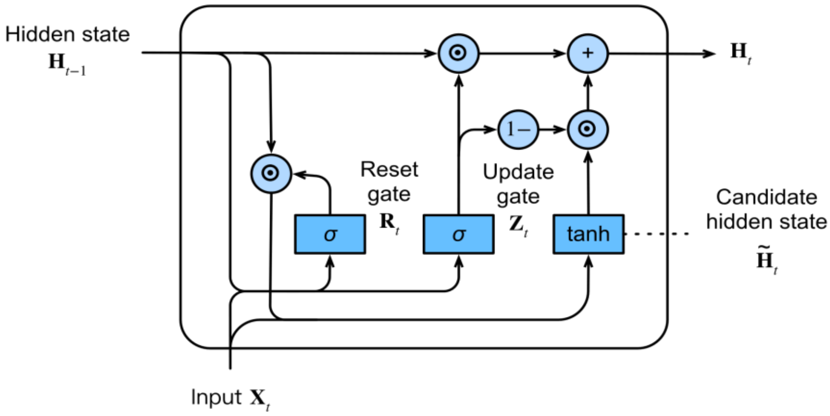


Fig 8. Model Development and Verification Process.

When the degree of tool wear was not properly collected, AutoEncoder, an unsupervised learning model, was selected. The input and output appear to have the same structure, and AutoEncoder was ultimately adopted because the encoder restores the input data well and guarantees minimum performance. The last machine learning selected was LightGBM in **Fig 9**. LightGBM is a Gradient Boosting Model, which is a tree-based learning algorithm. Like GBDT, it is a machine learning method that has good performance in binary classification because it is a method of repeatedly training the difference between correct and incorrect answers like a loss function.

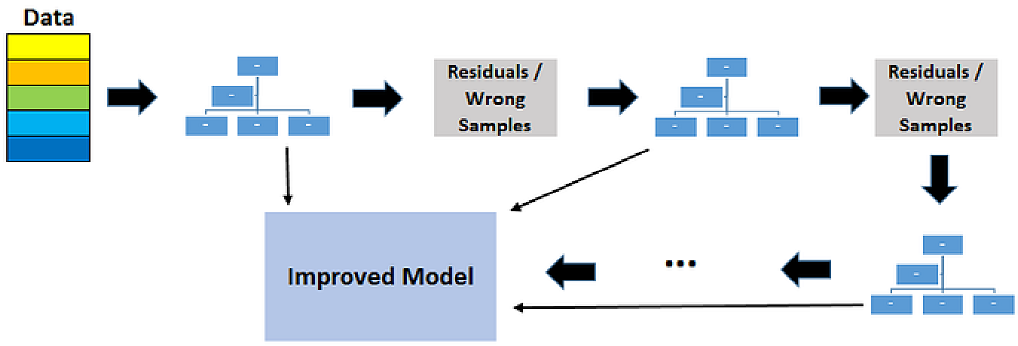


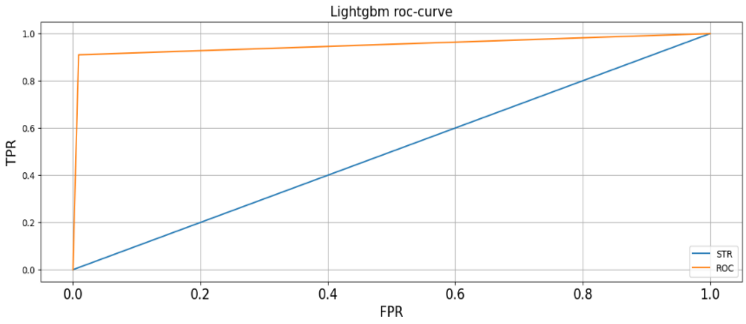
Fig 9. Light GBM Architecture.

V. RESULT AND DISCUSSION

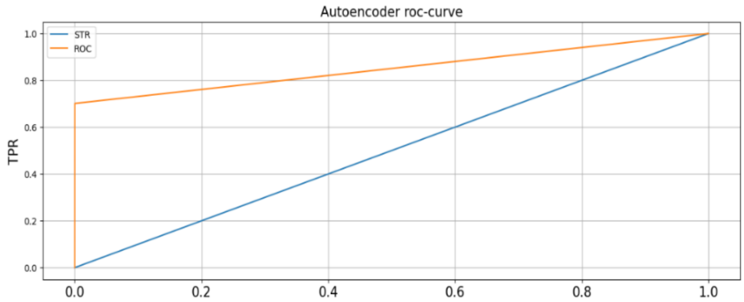
Using four LightGBM models, we selected the model with the best performance among Accuracy above 90% and F1-Score above 85%. After designing the model, Bayesian Optimization was performed to find the optimal parameter combination, and the model was verified using PoC data. The preprocessed data was divided into learning, validation, and testing data. At this time, the ratio of the divided data was divided into the training data set and the test data at a ratio of 80:20, and the split training data was split at a 20% ratio and divided into the verification data set. A model was first designed using the divided training data, and then hyperparameter optimization was performed using the validation data set. After hyperparameter optimization, training was performed on test data. Afterwards, it was verified by field application for two weeks. As a result, the results were obtained as shown in **Table 3** below. **Fig 10** displays the results of four algorithm experiments.

Table 3. Results of Performance Evaluation

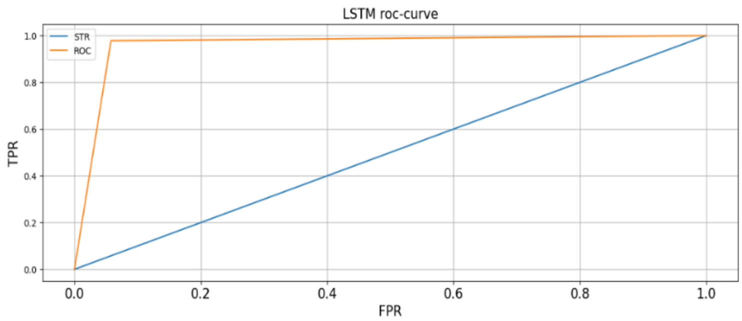
Model name	Accuracy	F1-Score
LSTM	96	68.75
GRU	95	64.71
Autoencoder	98.83	86.17
LightGBM	98.75	87.09



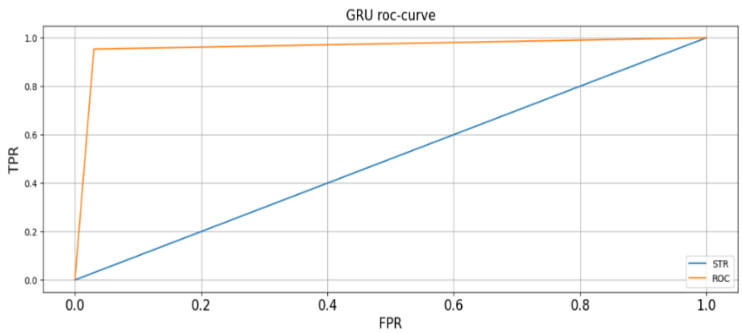
(a) Light GBM ROC-CURVE.



(b) Autoencoder ROC-CURVE.



(c) LSTM ROC-CURVE.



(c) GRU ROC-CURVE.

Fig 10. Four Algorithm Experiment Results.

As a result of the experiment, LightGBM showed the best performance. Afterwards, Bayesian Optimization was performed to optimize the parameters of LightGBM, as shown in **Table 4**.

Table 4. Bayesian Optimization Result

Parameter name	Value
bagging_fraction	0.896
feature_fraction	0.838
max_depth	21.517
min_child_weight	21.614
min_split_gain	0.034
num_leaves	88.068

The optimized LightGBM was installed in the field and verified for two weeks. The results are shown in **Table 5**. Through this, we were able to measure the performance of LightGBM. The measurement method is Accuracy and F1-Score.

Table 5. Confusion Matrix of Two Weeks of Field Installation Data

Confusion Matrix		Actual		
		Positive	Negative	SUM
Predicted	Positive	5672	50	5722
	Negative	25	253	278
	Sum	5697	303	6000

Table 6 shows that the tool was replaced a total of 4 times over 2 weeks. The average number of existing tool uses (replacement times) was 1,000 times (tool replacement when the CNC equipment setting value reached 1,000 times), and after the introduction of optimized LightGBM, the average number of times was 1,139, an increase of about 14% compared to before the introduction of LightGBM.

Table 6. Basis For Calculating the Increase in The Number of Replacements

division	before	After applied round 1	After applied round2	After applied round3	After applied round4	After applied Average
Tool replacement time	1000 times	1141 times	1137 times	1137 times	1139 times	about 1139 times

VI. CONCLUSION

Although tool consumption has increased after the introduction of optimized Light GBM, we plan to identify productivity issues when applying the same conditions to various production products through future research projects. Fig. Accordingly, 19 intends to provide guidance with the goal of reducing tool replacement costs by applying the LightGBM model optimized for multiple CNC equipment in various product groups in the future. Additionally, we plan to secure various feature data such as additional vibration sensors on CNC equipment to advance model performance and continue to collect data with the goal of increasing the current tool usage time by more than 5% to continuously improve model performance and inference.

Acknowledgment
This work was supported by Changshin University Research Fund of 2023-045.

Data Availability
No data was used to support this study.

Conflicts of Interests
The author(s) declare(s) that they have no conflicts of interest.

Funding
No funding agency is associated with this research.

Competing Interests
There are no competing interests.

References

[1]. J. Wan, X. Li, H.-N. Dai, A. Kusiak, M. Martinez-Garcia, and D. Li, "Artificial-Intelligence-Driven Customized Manufacturing Factory: Key Technologies, Applications, and Challenges," Proceedings of the IEEE, vol. 109, no. 4, pp. 377–398, Apr. 2021, doi: 10.1109/jproc.2020.3034808.

[2]. H. Ding, R. X. Gao, A. J. Isaksson, R. G. Landers, T. Parisini, and Y. Yuan, "State of AI-Based Monitoring in Smart Manufacturing and Introduction to Focused Section," IEEE/ASME Transactions on Mechatronics, vol. 25, no. 5, pp. 2143–2154, Oct. 2020, doi: 10.1109/tmech.2020.3022983.

[3]. Y. Teng, L. Li, L. Song, F. R. Yu, and V. C. M. Leung, "Profit Maximizing Smart Manufacturing Over AI-Enabled Configurable Blockchains," IEEE Internet of Things Journal, vol. 9, no. 1, pp. 346–358, Jan. 2022, doi: 10.1109/jiot.2021.3098917.

[4]. Y.-H. Kim and J. Lee, "Technology and Service Trends for Ensuring Safety in Smart Manufacturing," The Journal of Innovation Industry Technology, vol. 1, no. 3, pp. 123–128, Dec. 2023, doi: 10.60032/jiit.2023.1.3.123.

[5]. K. Liu, L. Song, W. Han, Y. Cui, and Y. Wang, "Time-Varying Error Prediction and Compensation for Movement Axis of CNC Machine Tool Based on Digital Twin," IEEE Transactions on Industrial Informatics, vol. 18, no. 1, pp. 109–118, Jan. 2022, doi: 10.1109/tii.2021.3073649.

[6]. A. S. Rajesh, M. S. Prabhuswamy, and S. Krishnasamy, "Smart Manufacturing through Machine Learning: A Review, Perspective, and Future Directions to the Machining Industry," Journal of Engineering, vol. 2022, pp. 1–6, Aug. 2022, doi: 10.1155/2022/9735862.

[7]. Z. M. Çınar, A. Abdussalam Nuhu, Q. Zeeshan, O. Korhan, M. Asmael, and B. Safaei, "Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0," Sustainability, vol. 12, no. 19, p. 8211, Oct. 2020, doi: 10.3390/su12198211.

[8]. N. Liu et al., "Modeling and Simulation of Robot Inverse Dynamics Using LSTM-Based Deep Learning Algorithm for Smart Cities and Factories," IEEE Access, vol. 7, pp. 173989–173998, 2019, doi: 10.1109/access.2019.2957019.

[9]. Y.-S. Park, D.-Y. Yoo, and J.-W. Lee, "Programmable Motion-Fault Detection for a Collaborative Robot," IEEE Access, vol. 9, pp. 133123–133142, 2021, doi: 10.1109/access.2021.3114505.

[10]. C. Y. Park, J. W. Kim, B. Kim, and J. Lee, "Prediction for Manufacturing Factors in a Steel Plate Rolling Smart Factory Using Data Clustering-Based Machine Learning," IEEE Access, vol. 8, pp. 60890–60905, 2020, doi: 10.1109/access.2020.2983188..

[11]. Y. Yang, T. Hu, Y. Ye, W. Gao, and C. Zhang, "A knowledge generation mechanism of machining process planning using cloud technology," Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 3, pp. 1081–1092, Mar. 2018, doi: 10.1007/s12652-018-0779-2.

[12]. Y. Xiao, Z. Jiang, Q. Gu, W. Yan, and R. Wang, "A novel approach to CNC machining center processing parameters optimization considering energy-saving and low-cost," Journal of Manufacturing Systems, vol. 59, pp. 535–548, Apr. 2021, doi: 10.1016/j.jmsy.2021.03.023.

[13]. He, Y. Cen, S. Alelaumi, and D. Won, "An Artificial Intelligence-Based Pick-and-Place Process Control for Quality Enhancement in Surface Mount Technology," IEEE Transactions on Components, Packaging and Manufacturing Technology, vol. 12, no. 10, pp. 1702–1711, Oct. 2022, doi: 10.1109/tcpmt.2022.3215109.

[14]. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Communications of the ACM, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[15]. Karen S, Andrew Z, Very Deep Convolutional Networks for Large-Scale Image Recognition. International Conference on Learning Representations. 2015 April:1-14. https://doi.org/10.48550/arXiv.1409.1556

[16]. A. Essien and C. Giannetti, "A Deep Learning Model for Smart Manufacturing Using Convolutional LSTM Neural Network Autoencoders," IEEE Transactions on Industrial Informatics, vol. 16, no. 9, pp. 6069–6078, Sep. 2020, doi: 10.1109/tii.2020.2967556.

[17]. W. Wu, L. Shen, Z. Zhao, M. Li, and G. Q. Huang, "Industrial IoT and Long Short-Term Memory Network-Enabled Genetic Indoor-Tracking for Factory Logistics," IEEE Transactions on Industrial Informatics, vol. 18, no. 11, pp. 7537–7548, Nov. 2022, doi: 10.1109/tii.2022.3146598.

[18]. F. Li, H. Zhou, M. Liu, and L. Ding, "A Medium to Long-Term Multi-Influencing Factor Copper Price Prediction Method Based on CNN-LSTM," IEEE Access, vol. 11, pp. 69458–69473, 2023, doi: 10.1109/access.2023.3288486.