Computers in Industry 142 (2022) 103740

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

Machine learning iterative filtering algorithm for field defect detection in the process stage

Young-Hwan Choi[a], Jeongsam Yang[b,*]

[a] Department of Digital Transformation, KD Navien Co. Ltd., 95 Suworam-gil, Pyeongtaek, Gyeonggi-do 17704, Republic of Korea
[b] Department of Industrial Engineering, Ajou University, 206 Worldcup-ro, Yeongtong-gu, Suwon 16499, Republic of Korea

ARTICLE INFO

Article history:
Received 8 May 2022
Received in revised form 28 June 2022
Accepted 1 July 2022
Available online 08 July 2022

Keywords:
Machine learning iterative filtering (MLIF) algorithm
Random undersampling
Classification
Recall
Sampling random number
Threshold for classification decision

ABSTRACT

Domestic gas boilers, which are seasonal appliances, undergo total inspections in the process stage and only the products that have passed the final shipment inspection are released. However, defects in the field do occur even in the products that have passed total inspection. In order to detect field defects as much as possible in the process stage, this study proposes the machine learning iterative filtering (MLIF) algorithm, which iteratively filters predicted defect-free products. This algorithm applies a unique method that uses n learners formed through multiple random undersampling in the majority class of defect-free products. In addition, the sampling random number and the threshold for classification decisions are specified under the condition where the recall result of classification is 100%. As a result, the classification prediction performance of the minority class is improved because it is extremely rare that an actual defective product is incorrectly predicted as a defect-free product. Experimental results showed that the final classification performance of the test data of the MLIF algorithm was 87%, which is approximately 11%p higher than that of single learner (76%). Finally, a process management dashboard, which shows various information about processes, field data, and the predicted results of the MLIF algorithm, to facilitate process decision making is presented.

1. Introduction

Domestic gas boilers, which are seasonal appliances, undergo total inspection in the process stage and only the products that have passed the final shipment inspection are released. However, defects in the field do occur even in the products that have passed total inspection. It is difficult to accurately determine the cause of defects in products in many cases, and this hinders the development of an optimal solution. One of the main reasons for the difficulty in detecting defects is the inability of the current production line measurement system and inspection equipment to adequately consider defective products owing to their low occurrences. In particular, defects in the field can result in grave consequences. For example, if the operational problems of products in the field persist, customer complaints increase, leading to erosion of brand image and adversely affecting sales. Moreover, when multiple gas boilers are installed at one site, an operational problem in one product may prompt a causal sequence, which can result in grave repercussions. Therefore, it is imperative to conduct a correlation analysis on the field problems of the products and the conditions of the process items to reduce the defects in the field. In addition, thorough process control should be performed to ensure that products that are likely to cause operational problems do not infiltrate the market. This can be ensured by applying an optimal method to classify the majority class as defect free and the minority class as defective based on the imbalanced dataset.

In recent studies, various solutions have been proposed to improve the classification performance of minority classes on the imbalanced dataset. Wang et al. (2020b) proposed a novel entropy and confidence-based undersampling boosting (ECUBoost) framework to solve imbalanced problems using KEEL datasets. Their framework ensures the validity and structural distribution of the majority samples during the undersampling using confidence and entropy in ECUBoost as benchmarks. Lee et al. (2021) presented a machine learning model for diagnosis prediction using patient information. They demonstrated that gradient boosting with the synthetic minority over-sampling technique can be used to predict a parasitic disease and serves as a promising diagnosis tool for binary and multi-classification schemes. Arefeen et al. (2022) proposed novel algorithms that employ neural network-based approaches to remove

[*] Corresponding author.
E-mail address: jyang@ajou.ac.kr (J. Yang).

https://doi.org/10.1016/j.compind.2022.103740
0166-3615/© 2022 Elsevier B.V. All rights reserved.

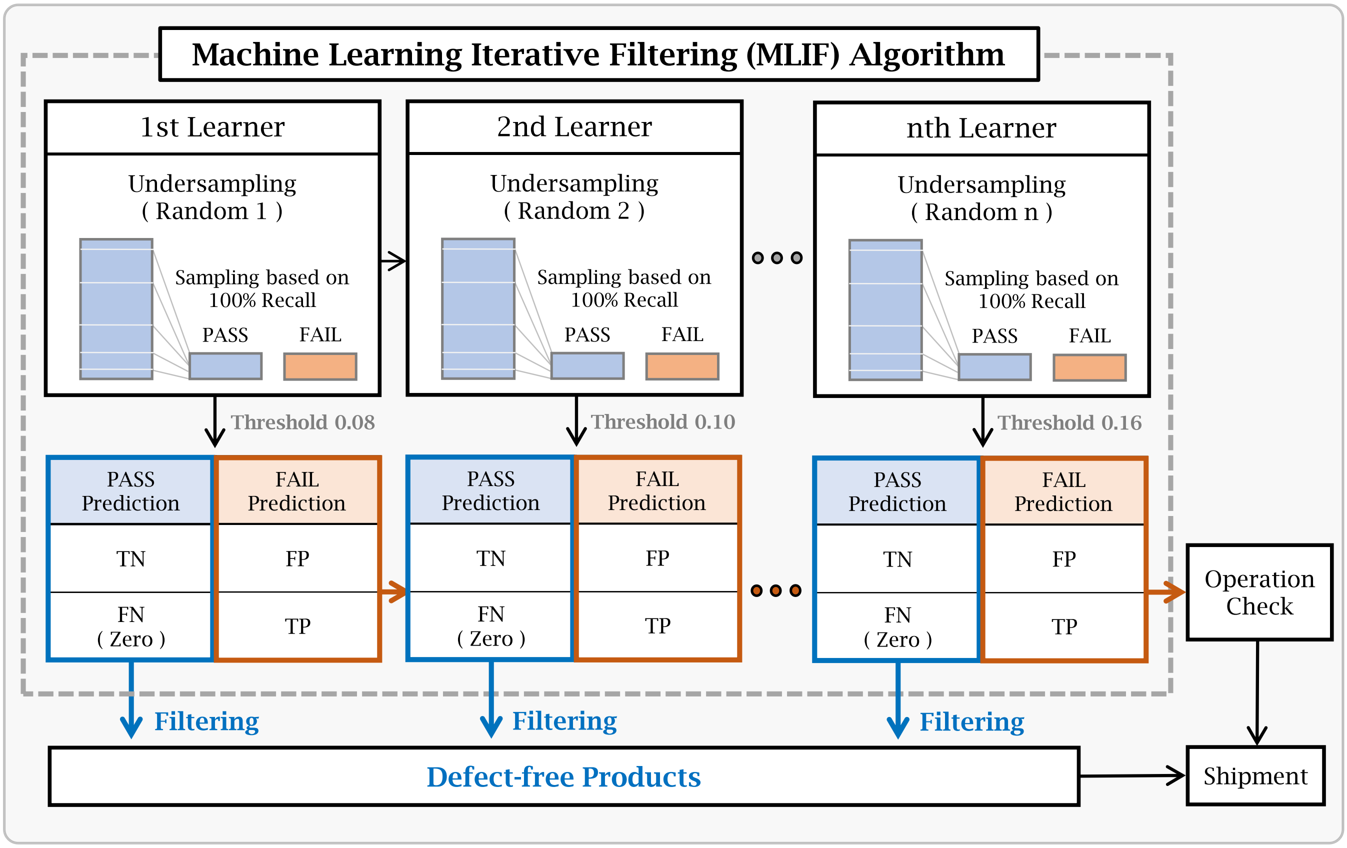

Machine Learning Iterative Filtering (MLIF) Algorithm
1st Learner
Undersampling (Random 1)
Sampling based on 100% Recall
PASS
FAIL
Threshold 0.08
2nd Learner
Undersampling (Random 2)
Sampling based on 100% Recall
PASS
FAIL
Threshold 0.10
nth Learner
Undersampling (Random n)
Sampling based on 100% Recall
PASS
FAIL
Threshold 0.16
PASS Prediction
FAIL Prediction
TN
FP
FN (Zero)
TP
Operation Check
Filtering
Filtering
Filtering
Defect-free Products
Shipment

Fig. 1. Machine learning iterative filtering algorithm.

majority samples that are found to reside in the vicinity of the minority samples, thereby undersampling to alleviate the imbalance issue. To summarize this brief literature review, classification methods proposed for the imbalanced dataset in the existing literature rely on the improvement of oversampling or undersampling methods rather than innovation of ML algorithms. Therefore, it is necessary to develop an innovative classification method that combines ML and the sampling method.

This study proposes the machine learning iterative filtering (MLIF) algorithm that can effectively predict and iteratively filter defect-free products that have an extremely low probability of defects in the field. Fig. 1 shows the overall structure and method of the MLIF algorithm. This method combines field defect data of each product with in-process measurement data, and iteratively filters predicted defect-free products using n learners that are formed through multiple random undersampling in the majority class of field defect-free products. Furthermore, our method avoids false predictions of defective products by specifying the sampling random number and the threshold for classification decisions, and forming learners under the condition that the recall result of classification is 100%. Finally, filtered defect-free products are shipped only with products that have thoroughly passed the final shipment inspection to prevent spillage into the field.

2. Literature review

ML, a technology that can predict results by training computers with considerable data patterns and entering new data, has significantly developed in various research fields. However, in real-world applications, such as medical diagnosis and fraud detection, ML algorithms can present a class imbalance problem, where there is a substantial difference between the distributions of the classes in a dataset. This greatly affects their predictive performance. In recent years, three main methods of handling class imbalance have been actively investigated.

The first method aims at identifying and detecting the minority class in an imbalanced dataset. The classification problem has been solved by applying various methods to identify minority classes. Di et al. (2018) presented a systematic approach to investigate the fault prediction of power converters in power conversion systems. Specifically, this method predicted faults after oversampling the defect classes using the synthetic minority oversampling technique (SMOTE) on imbalanced data sets. In addition, the method of predicting failures in power converters was validated for industrial use cases where only high-level system heartbeat signals were available. Malhotra and Kamal (2019) evaluated the efficiency of ML classifiers for 12 imbalanced NASA datasets to predict software defects. Specifically, they investigated five oversampling methods that replicate the instances of a minority class and proposed a new method, called SPIDER3. They applied the method to evaluate the performance of MetaCost learners for cost-sensitive training on an imbalanced dataset and improved the prediction performances of learning classifiers using oversampling. Wang et al. (2020a) proposed an imbalanced sampling approach that uses individualized self-paced learning (ISPL). Their proposed ISPL approach can effectively select high-quality samples for the classification of cancers that are imbalanced in an actual medical dataset. It exhibited approximately 16% higher performance compared to the average performance of other sampling methods. Gashi et al. (2021) proposed a method for predicting field defects using the End-of-line (EoL) test data that checks product performance and contextual information containing the service and location data of shipped products. This method utilized undersampling and SMOTE techniques to process imbalanced data and predicted defects using random forest and

LightGBM classifiers. Atoui and Cohen (2021) proposed a hybrid method for diagnosing single and multiple simultaneous faults while considering unknown operating conditions. This method predicted faults with a Bayesian classifier that combines statistical decisions and a fault signature matrix.

The second method aims to improve the classification performance by combining AI technology with a sampling method; various solutions have been proposed. Oh et al. (2019) proposed an oversampling method using the OD-GAN (outlier detectable generative adversarial network), which improves bias by considering many classes that influence classification boundaries. This method prevents the distortion of classification boundaries due to outliers by detecting and removing outliers through the discriminator used for training purposes only. In addition, the generator imitates the distribution of the minority class and produces artificial data to adjust the balance of the dataset. The experimental result showed that the OD-GAN method had superior prediction performance compared to other methods as the outliers in the dataset increased. Koziarski (2020) proposed a new radial-based undersampling (RBU) method based on the potential concept of mutual class. The proposed method expanded the concept of non-nearest neighbor-based resampling used in radial-based oversampling (RBO) to the undersampling process. The RBU method showed a statistically higher performance than the RBO method when combined with the CART decision tree. It also showed an optimal result when used in a complex dataset with a higher fraction of rare and anomaly values. Zhao et al. (2021) proposed the conditional variable autoencoder-based self-transmission (CVAE_SeTred) algorithm to solve the highly imbalanced classification problem with few training instances of a minority class. This method aimed at utilizing information from both the majority and minority classes, and transferring educational knowledge from the majority class to the minority class by employing oversampling that applies variational autoencoder (VAE) to a training sample for the minority class. The experimental results showed that the proposed method can generate samples with higher classification prediction performance and superior diversity than other compared sampling methods.

The third method improved the prediction performance of the minority class using either the classifier's ensemble method or by combining the ensemble method and undersampling. For this method, hybrid approaches were adopted. Thomas et al. (2018) proposed a proactive quality monitoring and control approach based on a classifier ensemble to predict defect occurrences and provide optimal values for factors critical to the quality processes. Specifically, the combination of different classifiers (NN, SVM, KNN, and Decision tree) in the ensemble improved the prediction results. Sestito et al. (2021) proposed a general and accurate anomaly detection technique suitable for any protocol based on real-time ethernet (RTE). In this method, if the first classifier cannot achieve the required accuracy, three ANN-based classifiers with different activation functions and three SVM-based classifiers with different kernels are employed. In addition, the practical feasibility and robustness of the proposed approach were demonstrated by applying it to an actual industrial automotive plant. Hoyos-Osorio et al. (2021) suggested a hybrid approach that combined clustering-based undersampling and bagging ensemble methods with the relevant information-based undersampling (RIUS) approach. The method selects an example that has the highest relevance in the majority class to improve the binary classification performance of imbalanced data.

Existing classification approaches proposed for imbalanced datasets have the limitation that they rely on improvements in oversampling or undersampling methods rather than innovations in ML algorithms. Specifically, because sampling is performed only once, the prediction result can vary significantly depending on the addition of new data and state changes. Moreover, undersampling methods have the disadvantage of substantial information loss in the majority class. This reduces the classification prediction performance. In addition, oversampling, which artificially increases a minority class for training, can cause overfitting and problems in data reliability owing to the generation of fake data.

Based on the limitations of this literature review, it was concluded that this study can contribute to the literature on imbalanced classification in the following ways.

(1) First, the MLIF algorithm improves the prediction performance of the minority class by performing random undersampling multiple times and repeating learning and prediction to overcome the disadvantages of conventional undersampling, which has a large information loss in the majority class.
(2) Second, each learner in the MLIF algorithm is trained and designed with 100% recall condition. Therefore, it reduces false prediction of defective products, which are a minority class.
(3) Third, as the MLIF algorithm does not use fake data, there is no problem with data reliability. Furthermore, MLIF enables continuous learning and prediction with fast execution times and can be applied to real business problems with excellent predictive performance of the minority class.

3. Design of machine learning iterative filtering

3.1. Overview

Supervised learning for imbalanced classification aims to predict the label values of a minority class in newly added data by pretraining the given features and label values in imbalanced data consisting of majority and minority classes. Fig. 2 illustrates the four-step process of the MLIF algorithm, which is suitable for imbalanced classification. In the first step, the optimal features are selected through box plot analysis and statistical testing of each feature. In the second step, the sampling method for balancing the imbalanced dataset is chosen. In the third step, the 2020 data employed for the study are randomly divided into 75% training data and 25% verification data. Subsequently, the classifier with the optimal performance is selected by comparing the prediction performances of various classification algorithms. In the fourth step, the performance is first evaluated by forming multiple learners to which the undersampling random number and the threshold for classification decisions were applied with 100% recall condition using the data produced in 2020. Subsequently, the final performance is evaluated with the test data produced in 2021.

3.2. Metrics

The geometric mean (G-Mean) between recall and specificity is mainly used to evaluate the classification performance for a highly imbalanced dataset (Ramos-Pérez et al., 2022). The G-Mean metric is calculated based on the confusion matrix, which shows the type of prediction errors the actual and predicted class values have in a quadrant matrix. Table 1 shows the confusion matrix mainly used in binary classification (Darzi et al., 2019). Here, TN (true negative) indicates a correct negative prediction, FP (false positive) indicates an incorrect positive prediction, FN (false negative) indicates an incorrect negative prediction, and TP (true positive) indicates a correct positive prediction. The objective of this study is to maximize the balance between TN and TP, which correctly predict actual PASS and FAIL, respectively. Therefore, the two ratios: recall and specificity that are related to TN and TP, are calculated before calculating the G-Mean.

Recall denotes the ratio correctly predicted as FAIL among all subjects that are actual FAIL, and is calculated using Eq. (1), whereas

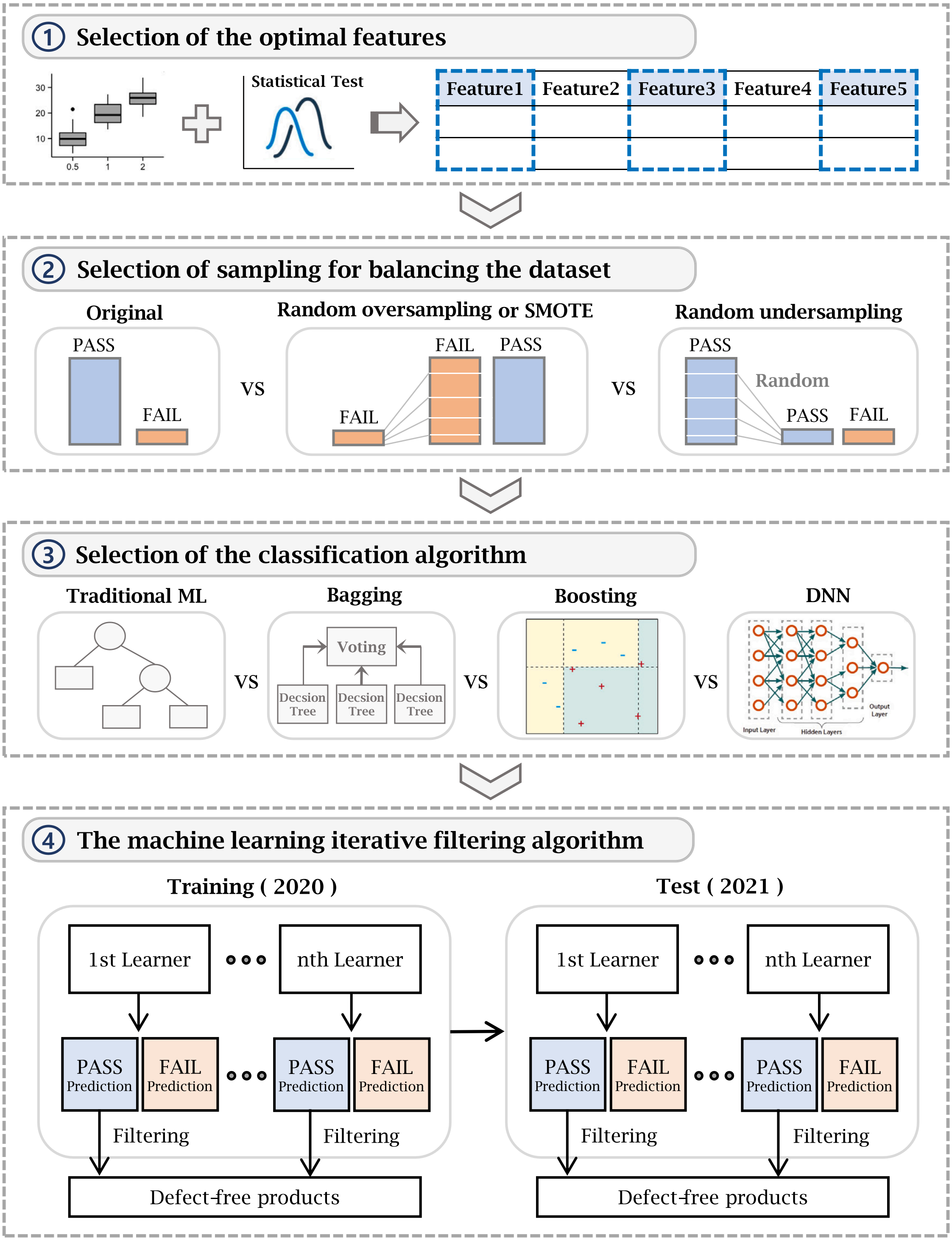


Fig. 2. Machine learning iterative filtering process.

Specificity implies the ratio correctly predicted as PASS among all subjects that are actual PASS, which is calculated using Eq. (2).

$$\text{Recall} = \frac{\text{TP}}{(\text{FN} + \text{TP})} \tag{1}$$

$$\text{Specificity} = \frac{\text{TN}}{(\text{TN} + \text{FP})} \tag{2}$$

G-Mean is the geometric mean between recall and specificity, and is calculated as Eq. (3).

$$\text{G} - \text{Mean} = \sqrt{recall \times specificity} \tag{3}$$

3.3. Selection of the optimal features

Table 2 shows the features and label values for 142,671 the gas boiler process measurement and field defect data from January 2020 to March 2021. Features largely consist of the measurement values of process inspection, the basic information of product and process equipment, and production dates. The gas boiler must be ignited for

Table 1
Confusion matrix of binary classification.

	Predicted Negative (PASS)	Predicted Positive (FAIL)
Actual Negative (PASS)	TN (True Negative)	FP (False Positive)
Actual Positive (FAIL)	FN (False Negative)	TP (True Positive)

operation and requires oxygen, air, and gas. Therefore, the O2, Fan, and Gas features are very significant. Month, jig, capa, and fuel features are object variables of strings that are not allowed as input values of ML. Thus, these variables are converted to numeric type using one-hot encoding. The label indicates whether a field defect occurred for products that are 1:1 matched to the manufacturing numbers of process data. Field defects are a minority class that is significantly smaller than the class of defect-free products. It is important to prevent the infiltration of defective products to the market by effectively increasing the classification performance by selecting the optimal features.

The optimal features must be selected from those that have a significant effect on field defects through box plot analysis and statistical testing. To begin with, the significance of quantitative features can be verified through the box plot visualization. Fig. 3 shows the box plot of each feature that visually expresses data using the minimum, maximum, first quartile, median, and third quartile. The O2(1) and O2(2) features are highly significant because FAIL in the field has a lower median and a wider dispersion than PASS (Fig. 3(a, b)). The fan feature is significant because the FAIL in the field has a lower value than PASS; however, the PASS and FAIL do not have a significant difference in dispersion. (Fig. 3(c)). The Gas(1) and Gas(2) features are highly significant because FAIL in the field has a lower median and a wider dispersion than PASS (Fig. 3(d, e)). The use-day feature is highly significant because FAIL in the field has much wider dispersion and a lower median than PASS (Fig. 3(f)). Therefore, it can be observed that O2(1), O2(2), Gas(1), Gas(2) and use-day features are critical for detecting defects in the field.

For statistical tests, two-sample t-test is performed for quantitative variables, and chi-square test is performed for qualitative variables (Montgomery, 2013). Defect-free and defective products in the field are classified into two groups, and a statistical significance test is performed. Table 3 shows the results of the statistical test for each feature. All features except the jig feature are statistically significant at the significance level < 0.05. Furthermore, the O2(1), O2(2), Gas(1), Gas(2), and use-day features, which are quantitative variables, have very high test statistic values, showing the same result as the visualization analysis of the box plot. However, the fan feature is also statistically significant. Thus, all features except jig are finally selected as the optimal features, and classification on imbalanced data is performed.

3.4. Selection of sampling for balancing the dataset

If training and prediction are performed when the dataset of ML is imbalanced, they have a negative impact on the classification performance (Benítez-Buenache et al., 2019). Therefore, it is essential to adjust the imbalanced data to a balanced ratio before training. Three sampling methods are widely used. The first method is random undersampling (RUS), which randomly selects instances in the majority class and deletes the other instances. The second method is random oversampling (ROS), which randomly replicates the instances of a minority class. The third method is synthetic minority oversampling technique (SMOTE), which oversamples a minority class after generating synthetic examples by randomly selecting from the neighbors closest to the parameter k in the original dataset (Chawla et al., 2002).

Table 4 shows the comparison of classification performance using the random forest algorithm with high speed and good performance after randomly dividing the production dataset of 2020 into training and verification data at the ratio of 75% and 25%, respectively, and balancing the training data using the three sampling methods. The SMOTE and ROS methods, which are oversampling methods, showed an unsatisfactory performance of approximately 40% for recall, which is the ratio of correctly predicting actual defective products as defective. In contrast, the RUS method showed a recall performance of approximately 73%, which is significantly higher than that of the oversampling method. Therefore, the prediction performances of the classification algorithms were compared by selecting the RUS method for the highly imbalanced dataset in this study.

3.5. Selection of the classification algorithm

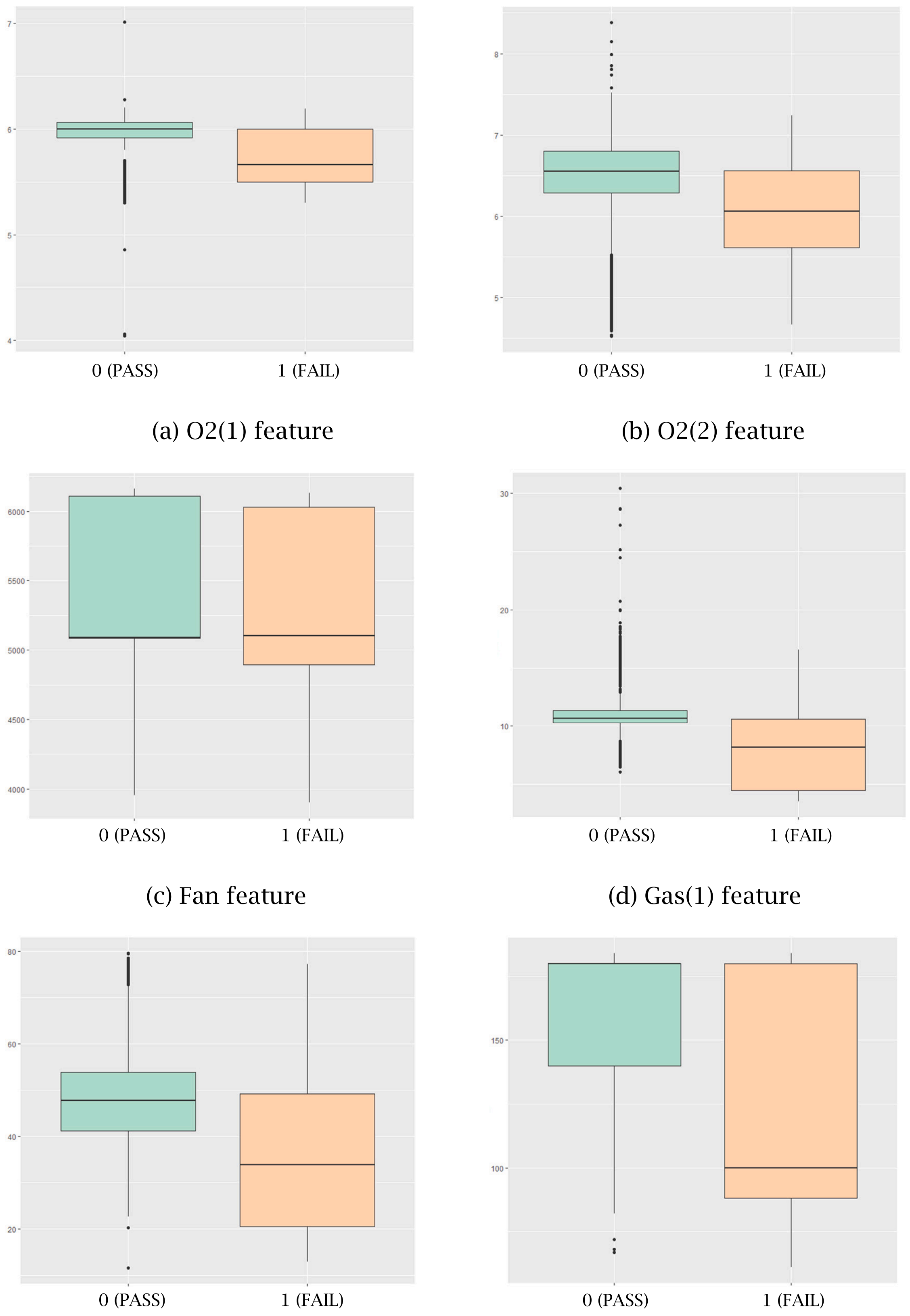
Table 5 shows the result of comparing the G-Mean of the various classification algorithms after randomly dividing the production dataset of 2020 into training and verification data at the ratio of 75% and 25%, respectively, and balancing the training data using the RUS method. It can be observed that random forest has the highest G-Mean at 0.7664 compared to the other classification algorithms. Specifically, logistic regression, DNNs, and SVM showed specificity values higher than 90%; however, their recall values were very low, resulting in a G-Mean performance lower than that of random forest. On the other hand, XGBoost, LightGBM, and decision tree showed recall values higher than 72%; however, their specificity values were low, resulting in a G-Mean performance lower than that of random forest. In conclusion, random forest with the highest G-Mean was selected as the classification algorithm and applied to the MLIF.

3.6. The machine learning iterative filtering algorithm

The MLIF algorithm is a method of iteratively filtering TN; it correctly predicts actual PASS products, using n learners formed by

Table 2
Information on feature and label values in data.

	No	Name	Type	Information
Feature	1	O2(1)	Float64	Initial process inspection data of oxygen
	2	O2(2)	Float64	End-of-process inspection data of oxygen
	3	Fan	Int64	Process inspection data of Fan (RPM)
	4	Gas(1)	Float64	Initial process inspection data of Gas
	5	Gas(2)	Float64	End-of-process inspection data of Gas
	6	Use_day	Float64	Average days of use by capacity and fuel
	7	Month	Object	Production month
	8	Jig	Object	Measuring equipment
	9	Capa	Object	Capacity of each product
	10	Fuel	Object	Fuel of each product
Label	11	Field defects	Int64	Result of field: 0 (PASS), 1 (FAIL)



(a) O2(1) feature

(b) O2(2) feature

(c) Fan feature

(d) Gas(1) feature

(e) Gas(2) feature

(f) Use_day feature

Fig. 3. Box plot for features.

Table 3
Statistical testing results of features (by Minitab).

Variable	Test	Hypothesis (H_1)	No	Feature	Test statistic	P-value	Result
Quantitative variable	2-sample t-test	There are differences between the two groups, good and bad.	1	**O2(1)**	53.80	0.000	**significant**
			2	**O2(2)**	33.30	0.000	**significant**
			3	**Fan**	-7.13	0.000	**significant**
			4	**Gas(1)**	39.94	0.000	**significant**
			5	**Gas(2)**	23.80	0.000	**significant**
			6	**Use_day**	38.01	0.000	**significant**
Qualitative variable	Chi-square test	There is a relationship between groups.	7	**Month**	77.27	0.000	**significant**
			8	Jig	16.37	0.291	not significant
			9	**Capa**	354.43	0.000	**significant**
			10	**Fuel**	6002.75	0.000	**significant**

The P-value of < 0.05 was considered to be statistically significant.

Table 4
Comparison of classification performance of sampling method.

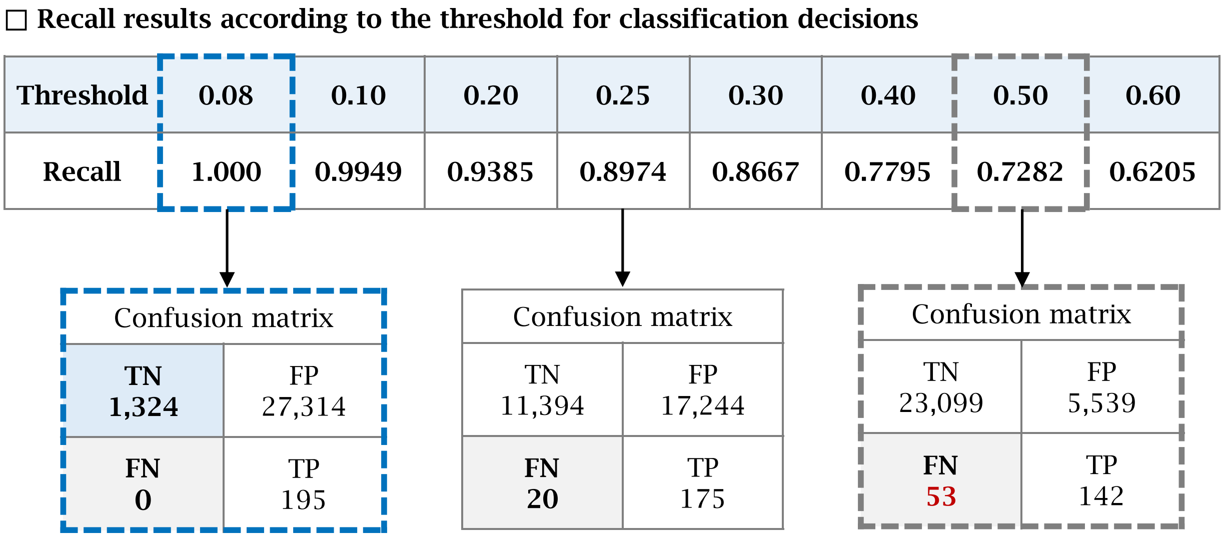
Training sampling	Training (75%)		verification (25%)		Classification performance		
	PASS	FAIL	PASS	FAIL	G-Mean	Recall	Specificity
Original	85,882	616	28,638	195	0.6284	0.3949	1.0000
SMOTE	85,882	**85,882**	28,638	195	0.6320	**0.4000**	0.9985
ROS	85,882	**85,882**	28,638	195	0.6284	**0.3949**	0.9999
RUS	**616**	616	28,638	195	**0.7664**	**0.7282**	0.8066

Table 5
Comparison of G-Mean according to various classification algorithms.

No	Classification algorithms	Hyper parameter	**G-Mean**	Recall	Specificity
1	**Random Forest**	Max depths (26) Min samples leaf (1) Min samples split (2)	**0.7664**	0.7282	0.8066
2	Logistic Regression	Penalty (L2) C (1)	0.7550	0.6205	**0.9186**
3	XGBoost	Learning rate (0.1) Num leaves (20)	0.7539	**0.7436**	0.7643
4	DNNs	Neurons (21, 10) Epochs (20) Batch size (100) Learning rate (0.01)	0.7514	0.6153	**0.9175**
5	Naïve Bayes	Var smoothing(1e-9)	0.7408	0.6103	0.8992
6	SVM	Kernel (RBF) C (5) Gamma (0.01)	0.7392	0.5743	**0.9515**
7	LightGBM	Learning rate (0.1) Num leaves (20)	0.7248	**0.7231**	0.7265
8	KNN	K (5)	0.7215	0.6615	0.7869
9	Decision Tree	Max depths (23) Min samples leaf (1) Min samples split (2)	0.6987	**0.7436**	0.6565

performing RUS multiple times in the majority class. Fig. 4 shows the principle of the MLIF algorithm that selects the threshold for classification decisions under the condition that the recall result is 100%. Generally, the threshold value in binary classification is set to 0.50, and this value becomes the criterion of probability that determines the positive prediction value. The prediction result is positive if the probability is larger than this criterion and negative if it is smaller. It has the characteristic that as the threshold value gradually decreases, the result of recall gradually increases. Eventually, when the threshold value becomes 0.08, the recall result becomes 100%, and the FN, which incorrectly predicts actual defective products as defect-free products becomes zero. Therefore, one learner is formed by reflecting this condition.

Fig. 5 shows the overall design of the MLIF algorithm that iteratively filters TN using n learners formed by performing RUS multiple times in the majority class 0. In the first learner with the sampling random number of 1 and the threshold of 0.08, FN is 0 and 1324 TN are filtered out. The remaining 27,509 are evaluated in the second learner with a random sampling number of 4 and a threshold value of 0.10. Eventually, FN becomes 0 and 1497 TN are filtered out. The remaining 26,012 are evaluated in the next learner. The MLIF algorithm repeats this process in this manner and stops when no TN is present. Therefore, for each learner, the sampling random number and the threshold for classification decisions are specified under the condition that the recall result is 100%, consequently making FN 0 and iteratively filtering TN. Eventually, TN is cumulatively summed after filtering, and FP gradually decreases. This gradually improves the classification prediction performance as the number of learners increases. Furthermore, the MLIF can perform continuous predictions faster because it is an algorithm that uses the same estimator for the sampling random number and the threshold for classification decisions, even though there are many learners. This shows that the MLIF algorithm is suitable for identifying the minority class 1 in a highly imbalanced dataset.



□ Recall results according to the threshold for classification decisions

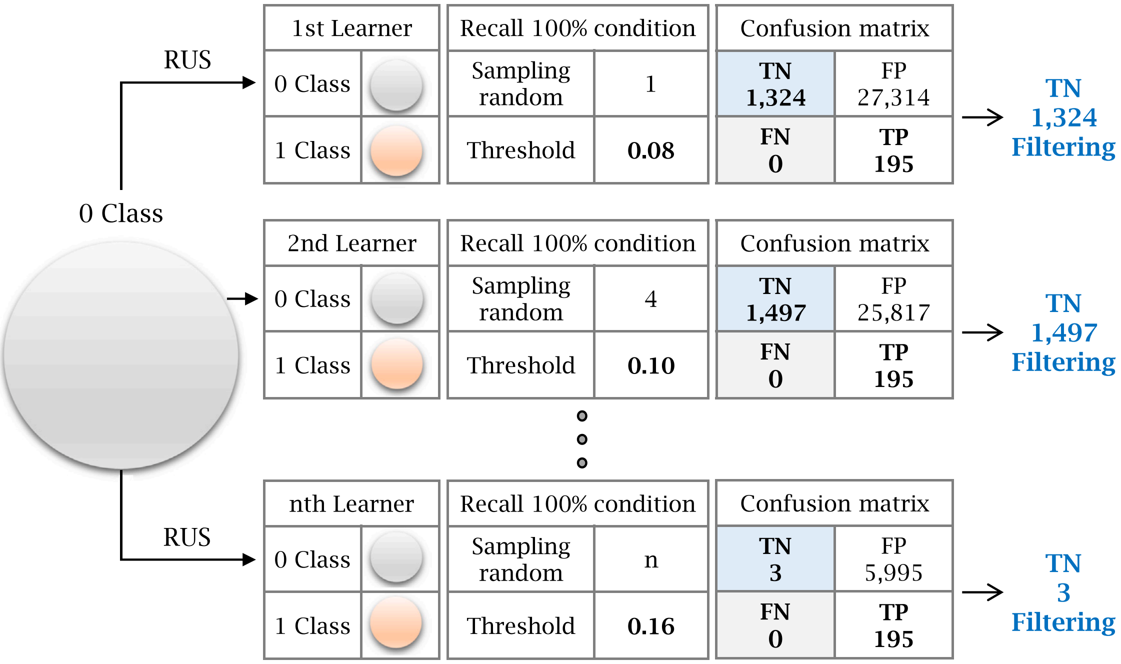
Threshold	0.08	0.10	0.20	0.25	0.30	0.40	0.50	0.60
Recall	1.000	0.9949	0.9385	0.8974	0.8667	0.7795	0.7282	0.6205

Confusion matrix	
TN 1,324	FP 27,314
FN 0	TP 195

Confusion matrix	
TN 11,394	FP 17,244
FN 20	TP 175

Confusion matrix	
TN 23,099	FP 5,539
FN 53	TP 142

Fig. 4. Principle of MLIF algorithm.



1st Learner		Recall 100% condition		Confusion matrix	
0 Class		Sampling random	1	TN 1,324	FP 27,314
1 Class		Threshold	0.08	FN 0	TP 195

TN 1,324 Filtering

2nd Learner		Recall 100% condition		Confusion matrix	
0 Class		Sampling random	4	TN 1,497	FP 25,817
1 Class		Threshold	0.10	FN 0	TP 195

TN 1,497 Filtering

nth Learner		Recall 100% condition		Confusion matrix	
0 Class		Sampling random	n	TN 3	FP 5,995
1 Class		Threshold	0.16	FN 0	TP 195

TN 3 Filtering

Fig. 5. Overall design of MLIF algorithm.

4. Evaluation of machine learning iterative filtering

4.1. The performance evaluation of ML

The performance of ML was evaluated by a primary evaluation distinguishing training and verification data and subsequently a secondary evaluation on the test dataset (Mahmoud et al., 2014). Fig. 6 shows the process of subdividing the dataset used in ML into training, verification, and test datasets, and subsequent performance evaluation. The data produced from January to December 2020 were randomly divided into 75% training data and 25% verification data. The training data were divided into 85,882 data for class 0 and 616 data for class 1. The verification data were divided into 28,638 data for class 0 and 195 data for class 1, showing that it is an extremely imbalanced dataset. Furthermore, the data produced from January to March 2021 were used as test data, and divided into 27,162 data for class 0 and 178 data for class 1.

4.2. The primary evaluation of the MLIF algorithm

Table 6 shows the primary evaluation result of the MLIF algorithm that iteratively filtered TN using a total of 435 learners formed by performing RUS multiple times in the majority class. It has the characteristic that as the number of learners increases, TN is cumulatively summed after filtering, and the FP, which incorrectly predicts actual defect-free products as defective products, decreases. As a result, the G-Mean performance gradually improved. Furthermore, the sampling random number and the threshold for classification decisions under the 100% recall condition were applied, which ensured that the FN that incorrectly predicts actual defective products as defect-free products becomes zero. Finally, the MLIF algorithm shows a high predictive performance with a G-Mean value of 0.8892 based on the 435th learner.

Table 7 shows the results of comparing the primary evaluation performance of the single learner and MLIF algorithm with data

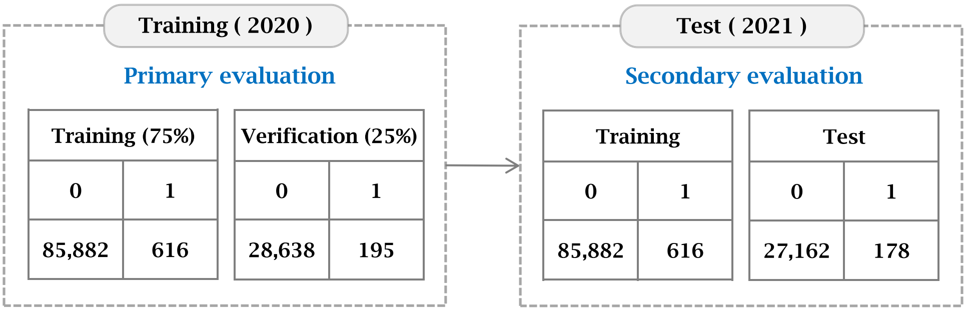


Fig. 6. Process of performance evaluation.

Table 6
Primary evaluation result of the MLIF algorithm.

No	Sampling random number	Threshold	Confusion matrix					**G-Mean**	Recall	Specificity
			TN Filtering	**TN** Accumulation	**FP**	**FN**	**TP**			
1	1	0.08	1324	**1324**	**27,314**	0	195	**0.2150**	1.0000	0.0462
2	4	0.10	1497	**2821**	**25,817**	0	195	**0.3139**	1.0000	0.0985
3	6	0.11	1426	**4247**	**24,391**	0	195	**0.3851**	1.0000	0.1483
4	8	0.13	1577	**5824**	**22,814**	0	195	**0.4510**	1.0000	0.2034
5	31	0.13	824	**6648**	**21,990**	0	195	**0.4818**	1.0000	0.2321
6	43	0.15	1218	**7866**	**20,772**	0	195	**0.5241**	1.0000	0.2747
7	57	0.13	769	**8635**	**20,003**	0	195	**0.5491**	1.0000	0.3015
8	137	0.13	411	**9046**	**19,592**	0	195	**0.5620**	1.0000	0.3159
9	147	0.14	461	**9507**	**19,131**	0	195	**0.5762**	1.0000	0.3320
10	284	0.13	343	**9850**	**18,788**	0	195	**0.5865**	1.0000	0.3439
11	346	0.15	421	**10,271**	**18,367**	0	195	**0.5989**	1.0000	0.3586
12	398	0.14	342	**10,613**	**18,025**	0	195	**0.6088**	1.0000	0.3706
13	446	0.14	355	**10,968**	**17,670**	0	195	**0.6189**	1.0000	0.3830
14	560	0.14	295	**11,263**	**17,375**	0	195	**0.6271**	1.0000	0.3933
15	565	0.15	353	**11,616**	**17,022**	0	195	**0.6369**	1.0000	0.4056
16	680	0.16	314	**11,930**	**16,708**	0	195	**0.6454**	1.0000	0.4166
17	840	0.15	329	**12,259**	**16,379**	0	195	**0.6543**	1.0000	0.4281
~ ~ ~ ~ ~ ~										
426	802,910	0.16	4	**22,602**	**6036**	0	195	**0.8884**	1.0000	0.7892
427	807,874	0.16	6	**22,608**	**6030**	0	195	**0.8885**	1.0000	0.7894
428	818,817	0.17	5	**22,613**	**6025**	0	195	**0.8886**	1.0000	0.7896
429	821,220	0.16	6	**22,619**	**6019**	0	195	**0.8887**	1.0000	0.7898
430	831,650	0.17	5	**22,624**	**6014**	0	195	**0.8888**	1.0000	0.7900
431	834,976	0.16	4	**22,628**	**6010**	0	195	**0.8889**	1.0000	0.7901
432	836,724	0.16	6	**22,634**	**6004**	0	195	**0.8890**	1.0000	0.7903
433	843,420	0.16	3	**22,637**	**6001**	0	195	**0.8891**	1.0000	0.7905
434	848,419	0.16	3	**22,640**	**5998**	0	195	**0.8891**	1.0000	0.7906
435	848,894	0.16	3	**22,643**	**5995**	**0**	**195**	**0.8892**	1.0000	0.7907

produced from January to December 2020. The single learner showed a recall of 0.7282, a specificity of 0.8066, and a G-Mean of 0.7664. Specifically, there were 53 cases of FN that incorrectly predicted actual defective products as defect-free products. This shows that field defects, which are a minority class, were not accurately detected. In contrast, the MLIF algorithm showed a recall of 1.000, a specificity of 0.7907, and a G-Mean of 0.8892. Therefore, the MLIF algorithm with a G-Mean of 0.8892 considerably improved the performance by 12.28%p compared to the G-Mean of 0.7664 of the single learner.

4.3. The secondary evaluation of MLIF algorithm

Table 8 shows the result of comparing the secondary evaluation performance of the single learner and MLIF algorithm after compositing test data from the data produced from January to March 2021. The single learner showed a recall of 0.7191, a specificity of 0.8007, and a G-Mean of 0.7588. Similar to the training result, there were 50 cases of FN. This indicates that the field defects, which are a minority class, were not accurately detected. In contrast, the MLIF algorithm showed a recall of 0.9607, a specificity of 0.7885, and a G-Mean of

Table 7
Comparison results of the primary evaluation of single learner and MLIF.

-	Confusion matrix			Recall	Specificity	G-Mean	Difference (%p)
	-	Predicted PASS	Predicted FAIL				
Single learner	Actual PASS	23,099 (TN)	5539 (FP)	0.7282	0.8066	**0.7664**	**▲12.28%p**
	Actual FAIL	**53** **(FN)**	**142** **(TP)**				
MLIF	Actual PASS	22,643 (TN)	5995 (FP)	1.0000	0.7907	**0.8892**	
	Actual FAIL	**0** **(FN)**	**195** **(TP)**				

Table 8
Comparison results of the secondary evaluation of single learner and MLIF.

-	Confusion matrix			Recall	Specificity	G-Mean	Difference (%p)
	-	Predicted PASS	Predicted FAIL				
Single learner	Actual PASS	21,749 (TN)	5413 (FP)	**0.7191**	0.8007	**0.7588**	**▲11.16%p**
	Actual FAIL	**50 (FN)**	**128 (TP)**				
MLIF	Actual PASS	21,418 (TN)	5744 (FP)	**0.9607**	0.7885	**0.8704**	
	Actual FAIL	**7 (FN)**	**171 (TP)**				

0.8704. It is rare that actual defective products, which are a minority class, are predicted as defect-free products owing to the design of the MLIF. Thus, the recall performance in the test dataset was 0.9607, indicating that the classification prediction performance of the minority class greatly improved. However, approximately seven cases of FN occurred because the defect reception period for the products produced in 2021 was shorter than in 2020. The final G-Mean of the MLIF algorithm was 0.8704, which is 11.16%p higher than the 0.7588 G-mean of the single learner.

5. Implementation of machine learning iterative filtering

5.1. Implementation environment and application scenario

This study proposed the MLIF algorithm, which iteratively filters predicted defect-free products to classify a minority class in an imbalanced dataset. Two software applications were employed to implement this algorithm and analyze results in real time. First, the scikit-learn library based on the Python programming language was used to implement the classification of ML. Second, Microsoft Power BI was used to implement a dashboard that shows various information of processes and the predicted results of the MLIF algorithm. The total inspection result for O2, Fan, and Gas, which are the process items, and the service data for defects in the field were saved in the database.

Fig. 7 shows the application scenario that consists of five steps for stable process management using the MLIF algorithm. In the first step, production starts from 08:30, and all the process items: (O2, Fan, and Gas) are measured. The measurement data are saved in the database in real time and subsequently combined into one table using the join query with field service data stored in the database based on the product code, which is the primary key. In the second step, the MLIF algorithm is performed, which iteratively filters predicted defect-free products using 435 learners formed by performing RUS on multiple products in the majority class of defect-free products. In the third step, various information related to processes and field data and the prediction result of the MLIF are presented through the process management dashboard implemented with Power BI. In the fourth step, after production is completed at 18:00, the predicted detect-free products are sampled from each lot based on the AQL 0.65, after which the final shipment inspection is performed. The passed lots are then shipped into the market. In the fifth step, the predicted defective products undergo a thorough operation check, after which qualified products are shipped into the market. Although time consuming, this process ensures the non-infiltration of defective products to the market.

5.2. Dashboard for process control

Fig. 8 shows the screen of the dashboard, which is composed of four areas showing the predicted result of the MLIF algorithm and various information of process data. From this dashboard, users can obtain information required for decision making by checking the prediction results for field defects and the process capability level of each process item. First, the area in Fig. 8(a) shows the predicted results of the MLIF algorithm for daily production. The products that are predicted as FAIL undergo a thorough operation check to judge for any abnormalities. The area in Fig. 8(b) shows the confusion matrix of four quadrants and the performance result of G-Mean. The ML engineers need to examine whether the G-Mean performance

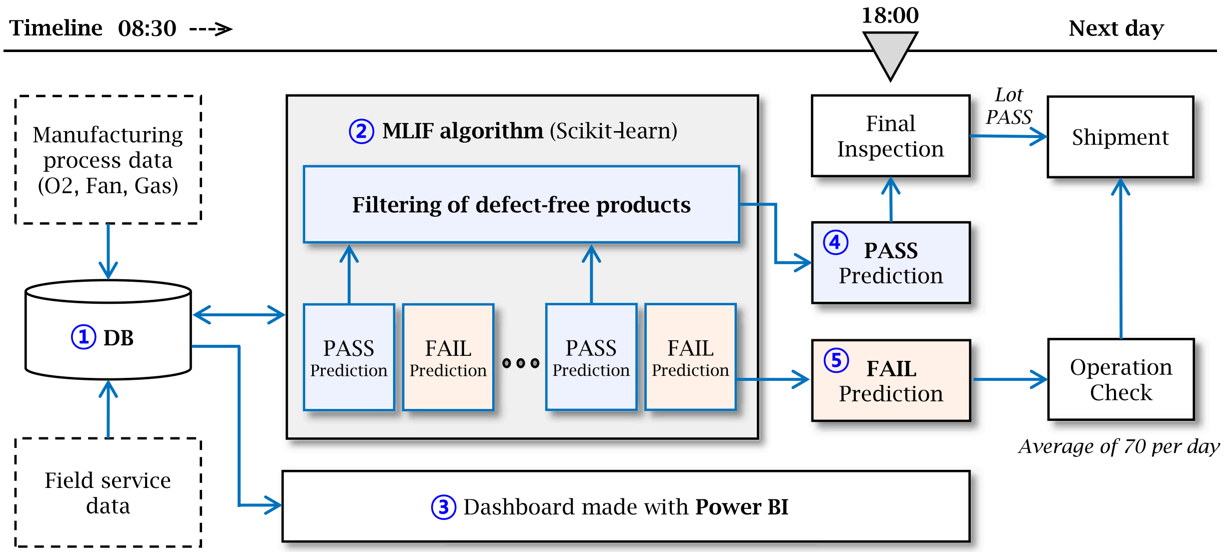
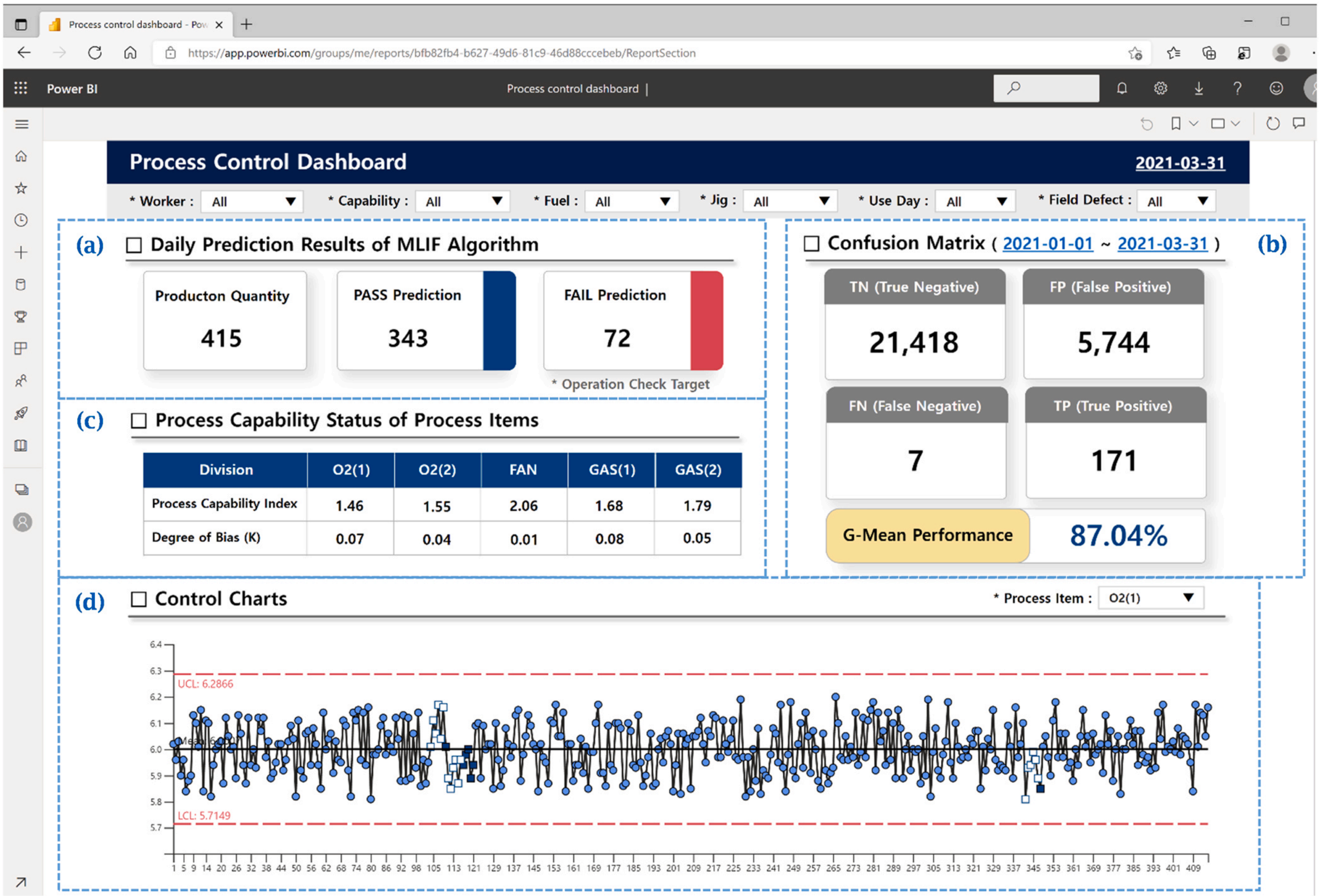


Fig. 7. Application scenarios of MLIF algorithm.


Process control dashboard - Pow
https://app.powerbi.com/groups/me/reports/bfb82fb4-b627-49d6-81c9-46d88cccebeb/ReportSection
Power BI
Process control dashboard |

Process Control Dashboard 2021-03-31

* Worker : All ▼ * Capability : All ▼ * Fuel : All ▼ * Jig : All ▼ * Use Day : All ▼ * Field Defect : All ▼

(a) □ Daily Prediction Results of MLIF Algorithm

Producton Quantity	PASS Prediction	FAIL Prediction
415	343	72

* Operation Check Target

(b) □ Confusion Matrix (2021-01-01 ~ 2021-03-31)

TN (True Negative)	FP (False Positive)
21,418	5,744
FN (False Negative)	**TP (True Positive)**
7	171

G-Mean Performance 87.04%

(c) □ Process Capability Status of Process Items

Division	O2(1)	O2(2)	FAN	GAS(1)	GAS(2)
Process Capability Index	1.46	1.55	2.06	1.68	1.79
Degree of Bias (K)	0.07	0.04	0.01	0.08	0.05

(d) □ Control Charts * Process Item : O2(1) ▼

Fig. 8. Dashboard for process control.

gradually decreases to keep it at the desired level. The area in Fig. 8(c) shows the process capability index (PCI) and the degree of bias (K) for each process item. In particular, the process capability is considered optimal if the PCI is higher than 1.33, and the bias of process is considered to be small if K is lower than 0.10 (Choi et al., 2020). Lastly, the area in Fig. 8(d) shows the control charts for process items. The x-axis represents the product number and the y-axis represents the measurements of process items. It is imperative to continuously monitor whether the measurement values of each product depart from the upper control limit (UCL) and lower control limit (LCL).

6. Conclusion

This study proposed an MLIF algorithm that can iteratively filter predicted defect-free products in the process stage. The functions and contributions of the MLIF algorithm are summarized as follows. First, the prediction performance was improved by preventing the loss of considerable information in the majority class and filtering data through many learners by performing multiple RUS. Second, the criterion of 0.50 for distinguishing defect-free and defective products in binary classification was significantly lowered to ensure the accurate detection of actual defective products, which are a minority class. This is because it is rare that a defective product is falsely detected as a defect-free product using our method. Third, the MLIF algorithm can perform faster because it uses the same estimator, even if there are many learners. The MLIF can thus be applied to real business problems. Fourth, a method for selecting the optimal features before applying the MLIF algorithm, and a process of selecting the best classifier after balancing imbalanced data were proposed. Fifth, a process management dashboard that shows various information related to process and field data, and the predicted result of the MLIF was presented. Decision making can be facilitated because process management engineers can examine the process data of products with a high probability of defects in the field on the dashboard. Therefore, the MLIF algorithm is a useful method to prevent field defects in various industrial applications.

Although the MLIF algorithm is suitable for identifying a minority class in a highly imbalanced dataset, it requires considerable amount of time to configure multiple optimal learners. Specifically, it takes a long time to find the optimal conditions for the sampling random number and the threshold for classification decisions. For example, in Table 6, it can be observed that the sampling random number progressed to 848,894 for the 435th learner. For future work, we intend to devise a method to facilitate the configuration of the optimal learners while improving the prediction performance.

CRediT authorship contribution statement

Young-Hwan Choi: Conceptualization, Data curation, Methodology, Writing – original draft, Software, Investigation, Visualization. **Jeongsam Yang:** Formal analysis, Validation, Project administration, Writing – review & editing.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Jeongsam Yang reports financial support was provided by National Research Foundation of Korea (NRF). Jeongsam Yang reports a relationship with National Research Foundation of Korea (NRF) that includes: funding grants.

Acknowledgement

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (grant number 2018R1D1A1B07050199).

References

Arefeen, M.A., Nimi, S.T., Rahman, M.S., 2022. Neural network-based undersampling techniques. IEEE Trans. Syst., Man, Cybern.: Syst. 52, 1111–1120.

Atoui, M.A., Cohen, A., 2021. Coupling data-driven and model-based methods to improve fault diagnosis. Comput. Ind. 128, 103401.

Benítez-Buenache, A., Álvarez-Pérez, L., Mathews, V.J., Figueiras-Vidal, A.R., 2019. Likelihood ratio equivalence and imbalanced binary classification. Expert Syst. Appl. 130, 84–96.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357.

Choi, Y.H., Na, G.Y., Yang, J.S., 2020. Fuzzy-inference-based decision-making method for the systematization of statistical process capability control. Comput. Ind. 123, 103296.

Darzi, M.R.K., Niaki, S.T.A., Khedmati, M., 2019. Binary classification of imbalanced datasets: the case of CoIL challenge 2000. Expert Syst. Appl. 128, 169–186.

Di, Y., Jin, C., Bagheri, B., Shi, Z., Ardakani, H.D., Tang, Z., 2018. Fault prediction of power electronics modules and systems under complex working conditions. Comput. Ind. 97, 1–9.

Gashi, M., Ofner, P., Ennsbrunner, H., Thalmann, S., 2021. Dealing with missing usage data in defect prediction: a case study of a welding supplier. Comput. Ind. 132, 103505.

Hoyos-Osorio, J., Alvarez-Meza, A., Daza-Santacoloma, G., Orozco-Gutierrez, A., Castellanos-Dominguez, G., 2021. Relevant information undersampling to support imbalanced data classification. Neurocomputing 436, 136–146.

Koziarski, M., 2020. Radial-based undersampling for imbalanced data classification. Pattern Recognit. 102, 107262.

Lee, Y.W., Choi, J.W., Shin, E.H., 2021. Machine learning model for diagnostic method prediction in parasitic disease using clinical information. Expert Syst. Appl. 185, 115658.

Mahmoud, S.A., Ahmad, I., Al-Khatib, W.G., Alshayeb, M., Parvez, M.T., Märgner, V., Fink, G.A., 2014. KHATT: an open Arabic offline handwritten text database. Pattern Recognit. 47, 1096–1112.

Malhotra, R., Kamal, S., 2019. An empirical study to investigate oversampling methods for improving software defect prediction using imbalanced data. Neurocomputing 343, 120–140.

Montgomery, D.C., 2013. Statistical quality control: a modern introduction, seventh ed. John Wiley & Sons, New York.

Oh, J.H., Hong, J.Y., Baek, J.G., 2019. Oversampling method using outlier detectable generative adversarial network. Expert Syst. Appl. 133, 1–8.

Ramos-Pérez, I., Arnaiz-González, Á., Rodríguez, J.J., García-Osorio, C., 2022. When is resampling beneficial for feature selection with imbalanced wide data? Expert Syst. Appl. 188, 116015.

Sestito, G.S., Turcato, A.C., Dias, A.L., Ferrari, P., Spatti, D.H., da Silva, M.M., 2021. A general optimization-based approach to the detection of real-time Ethernet traffic events. Comput. Ind. 128, 103413.

Thomas, P., Haouzi, H.B.E., Suhner, M.C., Thomas, A., Zimmermann, E., Noyel, M., 2018. Using a classifier ensemble for proactive quality monitoring and control: the impact of the choice of classifiers types, selection criterion, and fusion process. Comput. Ind. 99, 193–204.

Wang, Q., Zhou, Y., Zhang, W., Tang, Z., Chen, X., 2020a. Adaptive sampling using self-paced learning for imbalanced cancer data pre-diagnosis. Expert Syst. Appl. 152, 113334.

Wang, Z., Cao, C., Zhu, Y., 2020b. Entropy and confidence-based undersampling boosting random forests for imbalanced problems. IEEE Trans. Neural Netw. Learn. Syst. 31, 5178–5191.

Zhao, Y., Hao, K., Tang, X.S., Chen, L., Wei, B., 2021. A conditional variational auto-encoder based self-transferred algorithm for imbalanced classification. Knowl.-Based Syst. 218, 106756.