ICIC Express Letters
Part B: Applications
Volume **13**, Number **6**, June **2022**

ICIC International ©2022 ISSN 2185-2766
pp. **639–646**

IMPROVING THE PERFORMANCE OF MIXING-PROCESS MACHINE CLASSIFICATION BASED ON FEATURE ENGINEERING TECHNIQUES: A CASE STUDY IN RUBBER-BELT INDUSTRY

Alif Nur Iman, Jinuk Kim, Su Jeong Son, Muhammad Hanif Ramadhan and Hyerim Bae*

Industrial Data Science and Engineering, Department of Industrial Engineering
Pusan National University
2, Busandaehak-ro 63beon-gil, Geumjeong-gu, Busan 46241, Korea
{ alifnuriman; jason.jinuk.kim; handsj; mhanifr }@pusan.ac.kr
*Corresponding author: hrbae@pusan.ac.kr

Received October 2021; accepted December 2021

Abstract. *The machine is the primary tool in the manufacturing process. The features produced by a standard machine (in this case, a mixing machine for fabrication of rubber belts) ordinarily are minimal and log-oriented. Additionally, because the machine normally produces a good product rather than a not-good product, data imbalance is inevitable. Therefore, to develop a good machine learning model, data reconstruction is necessary. This paper proposes feature engineering method formulated based on an actual machine data. Explicit, implicit, and knowledge-and-human-learning-guided feature engineering was carried out in the present study in order to extract additional features. In order to overcome the data imbalance problem, the Synthetic Minority Oversampling Technique (SMOTE) was implemented. As evaluation metrics of the proposed method, Random Forest and eXtreme Gradient Boosting (XGBoost) were employed. The experimental results showed that the proposed method improves the performance of the existing algorithm.*

Keywords: Classification, Feature engineering, Machine learning, Random Forest, SMOTE, XGBoost

1. **Introduction.** With the advancement of science and technology, machines have come to play an essential role in the manufacturing process. To ensure that machines perform to their full potential, comprehensive analysis is required. Monitoring, prediction, and classification of machine performance can be beneficial in improving decision making in industrial management. The more manufacturers apply the state-of-the-art machine learning concept in their production processes, the more likely they experience growth.

Machine learning will analyze and learn to gain knowledge and even predict information based on data. In handling data input, it is necessary to pay attention to how data are preprocessed. Moreover, feature data generated by machines are typically logged data per second and separated by batches. This kind of data requires to be reconstructed before being utilized as input into machine learning. The mixing machine considered in the present study produces temperature, voltage, and RAM data as features. Shah et al. [1] emphasized that features engineering is essential to the development of a successful data-driven machine learning model.

In the machine industry, it is expected that a machine produces a good product rather than a not-good product. However, this incurs a data imbalance problem. Krawczyk [2] analyzed the different aspects of imbalanced learning, and found that one of possible solutions to imbalanced data problems is to focus on the structure and nature of examples in minority classes. Recent studies on the application of Synthetic Minority Oversampling

DOI: 10.24507/icicelb.13.06.639

Technique (SMOTE) in treating imbalanced data problems have shown promising results [3-5]. Comprehensive analysis of SMOTE by Elreedy and Atiya [6] indicated that it is an effective method for generating additional examples from the minority class.

For analysis of classification performance, evaluation metrics are utilized. Recent studies on application eXtreme Gradient Boosting (XGBoost) and Random Forest have been investigated. Budholiya et al. [7] applied the XGBoost classifier to predicting heart disease. Kiangala and Wang [8] implemented XGBoost and Random Forest to develop an effective adaptive customization platform for encoding of the customized data histories of small manufacturing plants. Both studies' results (i.e., accuracies > 90%) were promising. These methods, notably, are applicable to classification of mixing-machine data products.

This paper proposes a feature engineering technique to improve the performance of machine learning models. The entire process is illustrated in Figure 1. SMOTE was implemented to deal with imbalanced dataset problems. The present study collected a dataset from the real data of the mixing process in a rubber-belt industry. XGBoost and Random Forest were implemented as comparison models to analyze the effectiveness of feature engineering and dataset balancing performed by the proposed method.

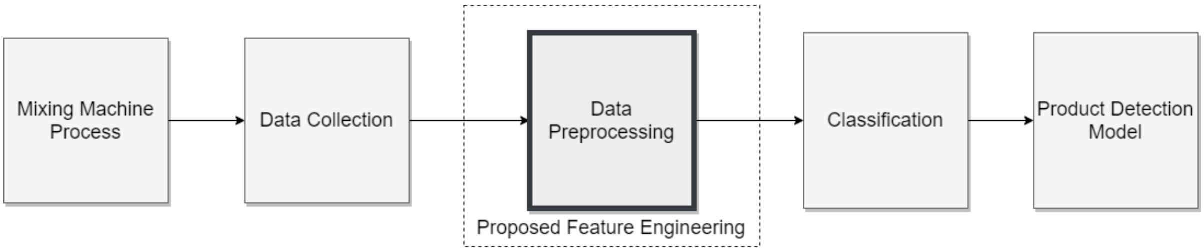

Figure 1. Framework of the proposed method

The remaining of the paper is structured as below. Section 2 summarizes the related work. Section 3 provides the problem statement, and Section 4 presents the proposed method's solutions. Section 5 discusses the experimental results, and Section 6 draws conclusions.

2. Related Work.

Relevant research on feature engineering techniques has been carried out [9-11]. Additionaly, the applications of SMOTE have been investigated in previous studies [3-5].

Tsay and Baldea [9] prescribed nonlinear transformations of model inputs in well-known dimensionless quantities as feature engineering to be processed in Artificial Neural Networks (ANNs). They had observed that feature engineering helps to improve prediction accuracy. Maphanga et al. [10] explicitly performed feature engineering by calculating density functional theory from material properties as an input dataset for machine learning models. They compared several regression models including linear regression, support vector machine, deep neural network, KNN, and Random Forest to find the best prediction results. Siradjuddin et al. [11] combined feature engineering and feature learning to classify the level dirtiness of image. They extract four features (structure, noise, diversity, and number of regions) to represent the level of dirtiness of an image.

Wang et al. [3] combined tree-based feature selection, the Synthetic Minority Oversampling Technique (SMOTE), and eXtreme Gradient Boosting (XGBoost) ensemble learning in classifying diesel fuel brands. Goyal et al. [4] combined SMOTE and Random Forest which incorporates entropy and data gain as function of fitness to validate and evaluate 4 standard datasets (Pima, ecoli, yeast, and segement). Raghuwanshi and Shukla [5] proposed SMOTE based Class-Specific Extreme Learning Machine (SMOTE-CSELM). The efficacy of SMOTE-CSELM is demonstrated by statistical test analysis. Results conducted by regarding combination of SMOTE were promising. Nevertheless, neither study implemented any feature engineering methods.

3. **Problem Statement.** An exploratory case study was performed with an industrial partner operating in rubber-belt production. There are several stages of that production process, one of which is the mixing process. The mixing process carries out two essential processes in the manufacture of rubber belts: cutting the molecular chain of raw material rubber by using chemicals and mechanical forces to impart plasticity to the raw rubber (Mastication) and uniformly dispersing formulation material in the masticated rubber (Milling).

The mixing machine can be briefly illustrated as follows (see Figure 2). The ingredients are fed into the machine, and the ramming process occurs in batches. Data such as voltage, temperature, and RAM are recorded and represented as features utilized in machine learning models. However, this data need to be interpreted, because the machine provides only a log of time-series data from each batch. Thus, feature engineering techniques will be needed to ensure reliable data for machine learning.

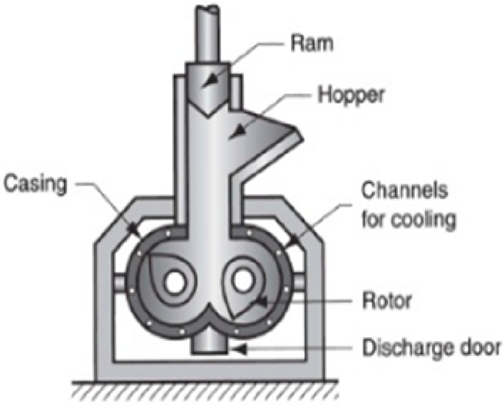

FIGURE 2. Mixing machine for rubber belts

4. **Proposed Feature Engineering.** Feature engineering entails modification of features in the data, usually by applying mathematical functions. Currently, there is no rule of thumb for how feature engineering is carried out, though the approaches can be divided into explicit feature engineering, implicit feature engineering, and knowledge-and-human-learning-guided feature engineering.

In our current case, we use explicit feature engineering to transform new features based on the primary features (temperature, voltage, and RAM) obtained from the machine, implicit feature engineering to balance the amount of good and not-good products, and knowledge and human learning guided feature engineering to get labels from the product in each batch.

4.1. **Explicit feature engineering.** Explicit feature engineering applies mathematical model functions to the transformation of features.

An example of data from a single batch is depicted in Figure 3. Based on the dataset, the mixing process occurs around 275-350 seconds. Features temperature and voltage can be extracted from average value while RAM by taking count continues interval of RAM ON. We assume these three features as primary features. However, these features are not sufficient to represent the complexity of the batch. Moreover, it shows that the RAM condition greatly affects the temperature and voltage values. Additional features are needed to explain the relationship between temperature, voltage, and RAM.

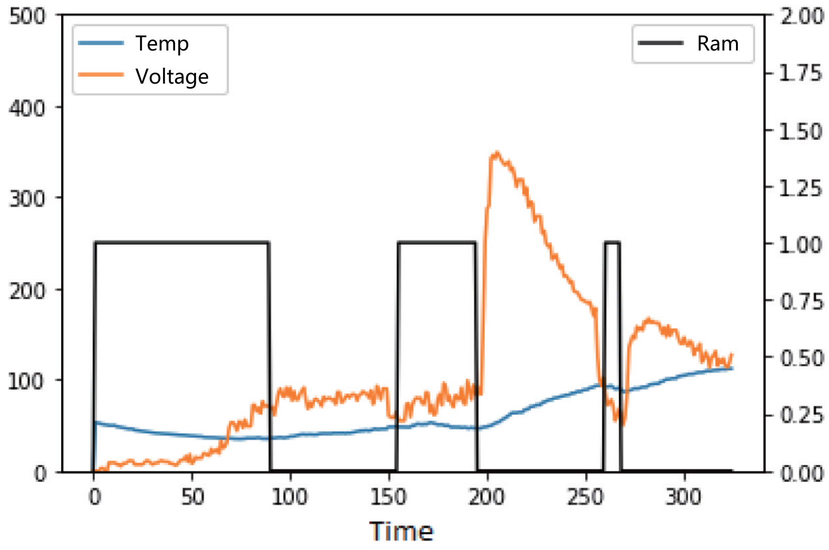

Figure 3. Data sample of the mixing process in one sequence

Therefore, explicit feature engineering is applied based on the characteristics of the data. We propose 25 engineering-feature results shown in Tables 1, 2, and 3. These features characterize the existence of the primary features. Features are constructed using mathematical functions (i.e., count, sum, max, average, difference, and flux) applied to each primary feature. Additionally, RAM ON and OFF conditions are also considered when transforming the temperature and voltage features.

Table 1. Proposed feature engineering based on RAM

Features	Description
Feature 1	Count continues interval for RAM ON in single batch
Feature 2	Sum cumulated time for RAM ON in single batch
Feature 3	Sum cumulated time for RAM OFF in single batch
Feature 4	Max cumulated time for RAM ON in single batch
Feature 5	Max cumulated time for RAM OFF in single batch

Table 2. Proposed feature engineering based on temperature

Features	Description
Feature 6	Average temperature in single batch
Feature 7	Average temperature in single batch when RAM ON
Feature 8	Average temperature in single batch when RAM OFF
Feature 9	Max temperature in single batch when RAM ON
Feature 10	Max temperature in single batch when RAM OFF
Feature 11	Difference max-min of temperature in single batch when RAM ON
Feature 12	Average variation rate of temperature in single batch when RAM ON
Feature 13	Maximum variation rate of temperature in single batch when RAM ON
Feature 14	Heat flux in single batch
Feature 15	Heat flux in single batch when RAM ON
Feature 16	Heat flux in single batch when RAM OFF

Table 3. Proposed feature engineering based on voltage

Features	**Description**
Feature 17	Average voltage in single batch
Feature 18	Average voltage in single batch when RAM ON
Feature 19	Average voltage in single batch when RAM OFF
Feature 20	Max voltage in single batch when RAM ON
Feature 21	Max voltage in single batch when RAM OFF
Feature 22	Difference max-min of voltage in single batch when RAM ON
Feature 23	Physics flux in single batch
Feature 24	Physics flux in single batch when RAM ON
Feature 25	Physics flux in single batch when RAM OFF

4.2. **Implicit feature engineering.** Several machine learning methods have implicitly implemented feature engineering, such as Principal Component Analysis (PCA) for dimensionality reduction, and kernel K-means clustering.

As we determined that the amount of data does not balance between good and not-good products in the current dataset, the oversampling technique was applied to meeting this challenge. Oversampling is a technique that generates new data in order to balance the quantity of data categories. One such technique, SMOTE, works by selecting adjacent data samples and then drawing a line connecting the data. A new sample continues to be generated based on the lines thus formed, until the minority class is equal to the majority class.

4.3. **Knowledge-and-human-learning-guided feature engineering.** Knowledge and human learning are acquired through experience and exploration.

In this case study, there were four indicators to determine rubber quality: Hardness, D2MN (Mooney Viscometer), M1T10 (number of seconds to reach 10% point), and M1T90 (number of seconds to reach 90% point). The value of each property must be within the range shown in Table 4. If one quality indicator is insufficient, the product will be labeled as not-good.

Table 4. Standard range of quality indicators

Hardness	**D2MN**	**M1T10**	**M1T90**
80 ± 3	85 ± 10	$4 : 10 \pm 40$	$11 : 10 \pm 80$

4.4. **Data classification.** The classification techniques were carried out using Random Forest and XGBoost with the same parameter values.

The first model is a Random Forest with parameters of 10 max depth and 100 number of trees. Primary features (voltage, temperature, and RAM) and 25 engineering-features are compared as the model's input. The performance of SMOTE is also analyzed for each input model.

The second model uses XGBoost, which has the same parameters as Random Forest in the first model. The model's input and performance of SMOTE were also carried out to determine the performance of this model.

Accuracy, sensitivity, specificity, and F1-score, as given in Equations (1), (2), (3), and (4) below, are used as evaluation metrics. True Positive (TP) and True Negative (TN) are good and not-good products, respectively, that are successfully identified. False Positive (FP) indicates a good product identified as a not-good product, and False Negative (FN), a not-good product identified as a good product.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{2}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3}$$

$$F1\text{-}score = \frac{2TP}{2TP + FP + FN} \tag{4}$$

Accuracy means the ratio of all data identified as good products to all data identified as not-good products. Sensitivity is the ratio of identified good products to total number of good products. Specificity is the ratio of identified not-good products to products that should be identified as not-good products. The F1-score is the harmonic mean of precision and recall in identifying products as not-good.

5. **Results.** The method is implemented using the python programming language with the help of the sklearn library to build a classification model. Classification models are implemented using XGBoost and Random Forest as metrics to measure the performance of the proposed engineering features. The dataset used is the log mixing process from a partner rubber-belt industry company which consists of 2914 records. The classification results were analyzed to evaluate the proposed feature engineering performance in terms of accuracy, specificity, sensitivity, and F1-score.

5.1. **Data preprocessing.** As explained, raw data from machines have to be preprocessed before application to machine learning models. The explicit feature engineering technique was applied as described in Section 4.1. Twenty-five features were produced in the process. Then, 70% of the data were used for training and 30% for testing. Three primary features, namely RAM ON count, average temperature all time, and average voltage categorized without engineering features, were used for comparative purposes.

The labels that are obtained through the process of knowledge-and-human-learning-guided feature engineering, standard machines tend to produce good products, and so it is natural that not-good products are rarely obtained. Implicit feature engineering has been implemented by applying the SMOTE technique to balancing the amount of data on good and not-good products in the training set.

5.2. **Random Forest results.** In this experiment, the Random Forest algorithm was applied as a classifier. The number of trees and the max depth were 100 and 10, respectively. We examined the performance without feature engineering (3 primary features) and the proposed method (25 engineered features). In addition, in both experiments, we also applied SMOTE.

As shown in Table 5, the proposed feature engineering improved all of the evaluation metrics. However, the implementation of SMOTE slightly reduces the accuracy, specificity, and F1-score of the proposed method. The reduced performance occurs because the false positive value is increasing. That means a product that is non-good is classified as good. Nevertheless, the proposed feature engineering was demonstrably better with than without feature engineering.

Table 5. Classification results in Random Forest

Technique	**Accuracy**	**Sensitivity**	**Specificity**	**F1-score**
Without Feature Engineering	87.31%	85.62%	89.06%	87.29%
Without Feature Engineering + SMOTE	88.00%	86.52%	89.53%	88.00%
Proposed Feature Engineering	90.40%	87.64%	93.25%	90.28%
Proposed Feature Engineering + SMOTE	90.06%	87.64%	92.42%	89.97%

5.3. **XGBoost results.** In the second experiment, the XGBoost classifier was applied. The experiment was carried out with the same hyperparameters and under the same conditions as was the first experiment.

Similarly to the first experiment, the proposed feature engineering improved the results, as indicated by the highest evaluation metric shown in Table 6. Interestingly, the implementation of SMOTE reduces the performance of without feature engineering. This occurs because the implementation of SMOTE increases both false negative and false positive values.

Table 6. Classification results in XGBoost

Technique	**Accuracy**	**Sensitivity**	**Specificity**	**F1-score**
Without Feature Engineering	87.66%	86.52%	88.83%	87.70%
Without Feature Engineering + SMOTE	86.97%	85.39%	88.60%	86.96%
Proposed Feature Engineering	90.29%	87.42%	93.25%	90.15%
Proposed Feature Engineering + SMOTE	90.51%	87.64%	93.48%	90.38%

Based on the overall results presented in Tables 5 and 6, it was concluded that the proposed feature engineering could significantly improve classification results in both Random Forest and XGBoost. However, the implementation of SMOTE has imprecise results between improving/reducing the classification results.

6. **Conclusions.** The proposed feature engineering method improves classification results significantly. Machine learning obtained additional information from the features that the explicit feature engineering process had provided. Based on all of the results, the evaluation metrics including accuracy, sensitivity, specificity, and F1-score were all increased by 2%-3% in XGBoost and Random Forest classification.

In overcoming dataset imbalance, the implementation of SMOTE does not provide substantial results. This happens because diversity in the minority class is not as important as a majority class. Thus, the oversampling technique will either cause the algorithm to fit the noise or incorporate minor classification data, which will impact the result of evaluation metrics, even though it slightly increases the performance of the proposed feature engineering only in the XGBoost model.

In the future, we will apply the proposed feature engineering to several types of machines. Our aim will be to find the rule of thumb on which features should be produced by feature engineering. Detailed analyses on the importance of each feature also will be needed.

Acknowledgment. This work is partially supported by the National Research Foundation of Korea (NRF) and the Korean government (MSIT) (No. 2020R1A2C110229411). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] D. Shah, J. Wang and Q. P. He, Feature engineering in big data analytics for IoT-enabled smart manufacturing – Comparison between deep learning and statistical learning, *Computers & Chemical Engineering*, vol.141, DOI: 10.1016/j.compchemeng.2020.106970, 2020.

[2] B. Krawczyk, Learning from imbalanced data: Open challenges and future directions, *Progress in Artificial Intelligence*, vol.5, no.4, pp.221-232, 2016.

[3] S. Wang et al., A new method of diesel fuel brands identification: SMOTE oversampling combined with XGBoost ensemble learning, *Fuel*, vol.282, DOI: 10.1016/j.fuel.2020.118848, 2020.

[4] A. Goyal, L. Rathore and A. Sharma, SMO-RF: A machine learning approach by random forest for predicting class imbalancing followed by SMOTE, *Materials Today: Proceedings*, 2021.

[5] B. S. Raghuwanshi and S. Shukla, SMOTE based class-specific extreme learning machine for imbalanced learning, *Knowledge-Based Systems*, vol.197, DOI: 10.1016/j.knosys.2019.06.022, 2020.

[6] D. Elreedy and A. F. Atiya, A comprehensive analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance, *Information Sciences*, vol.505, pp.32-64, 2019.

[7] K. Budholiya, S. K. Shrivastava and V. Sharma, An optimized XGBoost based diagnostic system for effective prediction of heart disease, *Journal of King Saud University – Computer and Information Sciences*, 2020.

[8] S. K. Kiangala and Z. Wang, An effective adaptive customization framework for small manufacturing plants using extreme gradient boosting-XGBoost and random forest ensemble learning algorithms in an Industry 4.0 environment, *Machine Learning with Applications*, vol.4, DOI: 10.1016/j.mlwa.2021.100024, 2021.

[9] C. Tsay and M. Baldea, Non-dimensional feature engineering and data-driven modeling for microchannel reactor control, *IFAC-PapersOnLine*, vol.53, no.2, pp.11295-11300, 2020.

[10] R. R. Maphanga, T. Mokoena and M. Ratsoma, Estimating DFT calculated voltage using machine learning regression models, *Materials Today: Proceedings*, vol.38, pp.773-778, 2021.

[11] I. A. Siradjuddin, A. Sakinah and M. K. Sophan, Combination of feature engineering and feature learning approaches for classification on visual complexity images, *International Journal of Innovative Computing, Information and Control*, vol.17, no.3, pp.991-1005, 2021.