Prediction of Fused Magnesium Operating Mode Based on ADASYN-XGBoost

Weijian Kong, Zhiyong Su, and Yinpeng Ding

Abstract—The operating modes in the smelting process of fused magnesium are cyclically shifted, resulting in severe fluctuations in electricity load. Accurate prediction of its operating mode shifting can optimize the power supply curve of an electric furnace, to improve electric energy efficiency and reduce electricity expenses. In this paper, we propose a prediction model of fused magnesium operating mode based on ADASYN-XGBoost. Four supervised machine learning algorithms including a eXtreme Gradient Boosting (XGB), ADASYN-LGB and ADASYN-RF, ADASYN-SVM were compared with the proposed ADASYN-XGB method. The results indicate that the ADASYN-XGB has the best prediction accuracy (92.5%), high average precision (>0.8), low hamming loss (0.03) and low ranking loss (0.075). Based on these results for classification performance and prediction accuracy, the ADASYN-XGB is a solid candidate for a correct classification of operating modes. These findings suggest that ADASYN-XGB systems trained with real data may serve as a new tool to assist in fused magnesium smelting process.

Index Terms—Machine learning, operating mode prediction, eXtreme gradient boosting, multi-label classification, adaptive sampling.

I. INTRODUCTION

The electrical fused magnesium furnace uses magnesite or lightly-burned magnesite powder as raw materials, and by heating the raw materials to about 2800℃, so they melt, separate out impurities, and recrystallize to obtain qualified fused magnesium. Fused magnesium is widely used in the metallurgical industry, aerospace, nuclear power and other fields. Five operating modes are identified by experience of experts and field operators during smelting process of fused magnesium, such as semi-melt mode, normal-melt mode, over-melt mode, a-load mode, and gas-expel mode. Additionally, taken relationship between operating modes into account, the smelting process can also be divided into no-shifted mode and shifted mode.

According to different operating modes, the power required for fused magnesium is also different. Some operating modes require high power, such as normal-melt mode; some operating modes require less power, such as gas-expel mode. In addition, the duration of each operating mode is also different. On the contrary, shifted time is very short, which leads to a dilemma that when predicted by

algorithms the shifted mode samples will be deleted as abnormal data, their predictions will be previous mode. If the model is directly built with limited data, the generalization performance of shifted mode is poor, and the prediction accuracy is low. Therefore, there are two problems to be solved: predicting multi-label classification and the imbalanced samples.

Nowadays, with the continuous development and application of information collection and data transmission in various industries, a large amount of multi-label data has been accumulated, and the research on multi-label classification has been gradually extended to image processing, gene function, emotion classification, and other fields. In recent years, a host of multi-label classification algorithms have been proposed, which are mainly divided into two major classes: problem transformation [1], [2] and algorithm adaptation [3]. The problem transformation methods decompose a multi-label problem into one or more single-label subproblems, which can be solved directly by traditional classification models. To achieve this goal, there are three main methods for problem transformation: Binary Relevance [4], Label Powerset [5], and Pairwise (two-two association method). BR is a typical binary relevance method. It constructs the binary classifier independently and does not consider label relevance at all. To introduce label relevance into BR, Godbole and Sarawagi proposed a model containing two layers of BR. Read *et al.* proposed a classifier chain (CC) model for multi-label classification, which constructs a binary classifier chain with each classifier corresponding to one label. Then, the multi-label problem is transformed into a single-label multi-class problem, such as RAkEL [6], [7] and EPS. While the algorithm transformation method uses existing single-label algorithms for multi-label classification. This research focuses on the application of decision trees [8], k-nearest neighbors [9], [10], neural networks and support vector machines. A. Clare and R. D. King construct a decision tree by a top-down method, whose root contains all the training samples. For the non-leaf nodes in the tree, each feature is examined one by one to find the appropriate splitting point so that the data at that node can be split for maximum information gain.

In fact, severe class imbalances are prevalent in our lives, such as fraud detection [11] and medical image analysis [12], so effective identification of imbalanced data is an important area of machine learning [13]. Although there has been enough development in machine learning, there are few studies in this area due to imbalanced data [14]. As a discriminative model, the convergence and performance of machine learning can be significantly affected by skewed

data[15], [16]. Since majority and minority classes have different effects on the objective function, classifiers tend to ignore labels with lower frequencies in order to achieve higher overall accuracy, which leads to more errors referring to minority classes. In the study of multi-label learning, the dataset also has the same problem of uneven label distribution and have very limited related research. Resampling technique is a classifier-independent solution, and widely used in traditional imbalance learning [13], such as SMOTE, ADASYN, etc. Multi-label imbalance algorithms based on resampling methods have been proposed in [17]]-[[23] and have been shown to have the ability to improve classification results. Essentially, the purpose of data resampling is to adjust the bias of the classifier by changing the prior probability distribution of the classes in the training data. In the data preprocessing stage, resampling methods are effective in improving the performance of the model.

This paper proposes a method to build an ADASYN-XGBoost model for predicting operating modes based on the features of the original data of electrical fused magnesium operating modes. After obtaining the raw data of the fused magnesium smelting process, we first use feature engineering to remove redundant features, and then resample through ADASYN to generate five new training set, build XGBoost models, and finally predict the operating modes.

The rest of this paper is organized as follows: Section II presents the operating mode prediction method based on ADASYN-XGBoost. Section III describes the experiments. Section IV presents the analysis of experimental results. Finally, Section V presents conclusion and future.

II. Operating Mode Prediction Method Based on ADASYN-XGBoost
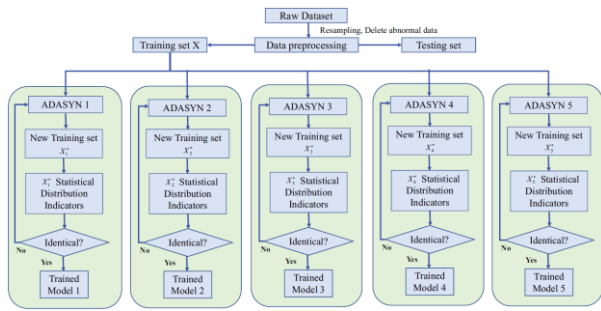
(a) Training Framework Diagram
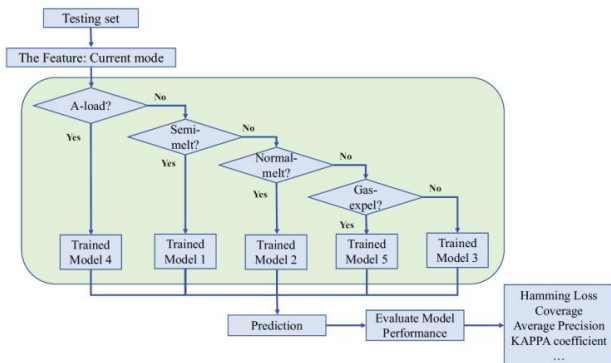
(b) Testing Framework Diagram
Fig. 1. Experiment Framework Diagram Based on ADASYN-XGBoost.

Five operating modes of electrical fused magnesium are cycled alternately according to a certain rule during the smelting process, but the duration of each operating mode is different, such as the duration of the previous gas-expel mode is about 12 seconds, the current may be 24 seconds. The uncertainty of such duration brings great difficulties to the prediction of fused magnesium operating modes. In this paper, we propose an operating mode method based on ADASYN-XGBoost. We build five models respectively, and input the testing dataset to the respective models for prediction by recognizing the current operating mode. The method mainly includes two steps: Model Building based on ADASYN and Prediction of Operating Mode. The method step flow is shown in Fig. 1.

A. Model Building Based on ADASYN

Two modes mentioned in the first section: *no-shifted mode* and *shifted mode*. "No-shifted mode" means that the current operating mode and the next mode are the same. And "*shifted mode*" means that the current operating mode and the next mode are different. The specific performance of fused magnesium operating mode is as follows: (1) At the beginning operating mode will continue for a while, then the mode shifting will occur, which causes that no-shifted modes are majority classes and shifted modes are minority classes; (2) There is often a correlation between the current mode and the next mode, so the first issue we discussed in this section is the shift order of the operating modes.

B. Classification Mining Process Based on CART Tree

The CART tree determines the attribute division points by calculating the size of the Gini impurity coefficient, and the information contained in the dataset is measured by the Gini impurity coefficient. Assuming that there are K classes, and the probability of belonging to the k-th class in a sample is P_k, the formula of the Gini impurity coefficient of the probability distribution is shown in formula (1).

$$\text{Gini}(p) = \sum_{k=1}^{K} P_k(1 - P_k) = 1 - \sum_{k=1}^{K} P_k^2 \qquad (1)$$

According to the definition of the Gini coefficient, the Gini index of the sample set U is shown in formula (2), where I_k represents the subset of samples in U that belong to the k-th class.

$$\text{Gini}(U) = 1 - \sum_{k=1}^{K} \left(\frac{|I_k|}{|U|}\right)^2 \qquad (2)$$

According to the feature A is divided at a certain value a, U is divided into U_1 and U_2. Under feature A, the Gini coefficient of set U is shown in formula (3), where the Gini coefficient $Gini(U)$ reflects the probability that two samples randomly selected from U have inconsistent class labels. The smaller the Gini coefficient $Gini(U)$, the higher the purity of U and the better the branch. The Gini coefficient $Gini(U, A)$ represents the impurity of the set U after division by $A = a$, as shown in formula (3).

$$\text{Gini}(U, A) = \frac{|U_1|}{|U|}\text{Gini}(U_1) + \frac{|U_1|}{|U|}\text{Gini}(U_2) \qquad (3)$$

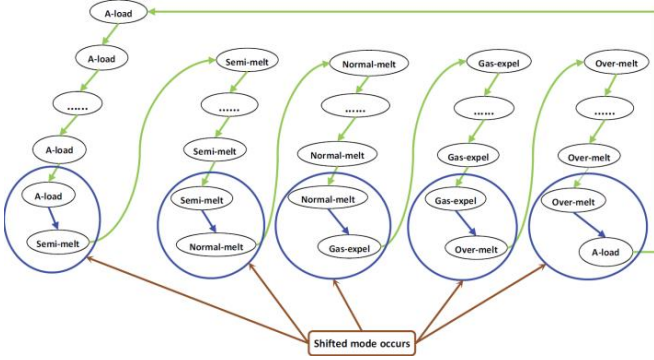The structure of the operating mode generated by the CART tree is shown in Fig. 2.

Fig. 2. Cycle of operating modes.

C. Generating New Training Sets Based on ADASYN

As there are 5 operating modes in the smelting process of electrical fused magnesium, it is necessary to generate 5 new training sets by ADASYN resampling technique and build 5 different models as follows: (1) Model 1: Compared with the other 4 models, when current mode is semi-melt mode, the accuracy of prediction is higher. (2) Model 2: If the current mode is the normal-melt mode, the accuracy of prediction is higher. (3) Model 3: The accuracy of prediction is higher if the current operating mode is over-melt. (4) Model 4: The accuracy of prediction is higher if the current operating mode is A-load, and (5) Model 5: The accuracy of prediction is higher if the current operating mode is gas-expel. Each model is resampled by its own ADASYN to generate a new training set of its own model. The details based on ADASYN in this paper are as follows.

The training set X has h samples $\{x_i, y_i\}, i = 1, \dots, h$, where x_i is a sample of the n-dimensional feature space X and y_i is the classification label associated with x_i. Define $m_{sj}(j = 1,2,3,4,5)$ is the number of minority class samples of the j-th model and m_{lj} is the number of majority class samples of the j-th model, $m_{sj} < m_{lj}$. And d_{thj} is a predefined threshold for the maximum tolerable degree of class imbalance ratio of the j-th model.

The main steps can be summarized as follows:

1) Determine the minority class of the j-th model, where $j = 1,2,3,4,5$.

2) Calculate the imbalance of the j-th model

$$d_j = m_{sj} / m_{lj} \text{ where } d_j \in (0,1] \tag{4}$$

3) Calculate the number of synthetic data instances that need to be generated for the j-th model minority class if $d_j < d_{thj}$, then

$$G_j = (m_{lj} - m_{sj}) \times \beta_j \tag{5}$$

$\beta_j \in (0,1]$ is a parameter that specifies the desired balance level after the j-th model generates synthetic data instances. $\beta_j = 1$ means that the j-th model generates a fully balanced dataset after the synthesis process.

4) For each sample x_{ij} belonging to a minority class in the j-th model, the k nearest neighbors are calculated in n dimensional space using Euclidean distance, the ratio is as follows:

$$r_{ij} = \frac{\Delta_{ij}}{K}, i = 1, \dots, m_s \tag{6}$$

where Δ_{ij} is the number of samples belonging to the majority class among the K nearest neighbors of x_{ij}, so $\Delta_{ij} \in [0,1]$.

5) Regularize r_{ij} according to $\hat{r}_{ij} = r_{ij} / \sum_{i=1}^{m_{sj}} r_{ij}$ to obtain the density distribution of $(\sum_{i=1} \hat{r}_{ij} = 1)$.

6) Calculate the number of synthetic data instances to be generated for each minority class sample x_{ij} of the j-th model.

$$g_{ij} = \hat{r}_{ij} \times G_j \tag{7}$$

where G_j is the total number of synthetic data instances that need to be generated for samples of minority classes, as shown in formula (5).

7) For each minority class sample x_{ij} of the j-th model, g_{ij} synthetic data instances are generated:

A data of minority class sample x_{zij} is randomly selected from the k nearest neighbors of x_{ij}. An example of synthetic data is generated as follows:

$$s_{ij} = x_{ij} + (x_{zij} - x_{ij}) \times \lambda_j \tag{8}$$

where s_{ij} is different vectors in n-dimensional space, the random variable $\lambda_j \in [0,1]$.

8) Determine whether the new training set obeys the identical distribution. If it obeys the identical distribution, the new training set will be send to XGB to build the model; if it does not obey the identical distribution, the above 2) to 7) will be repeated until it obeys the identical distribution.

Through the above steps, five new training sets $X_1^*, X_2^*, X_3^*, X_4^*, X_5^*$ can be generated, and Model 1, Model 2, Model 3, Model 4 and Model 5 are built respectively.

D. Prediction of Operating Mode

The current operating mode of each data in the testing set will be recognized, and then they will be input into the respective models for prediction. Specifically, if the current mode is semi-melt mode, the data will be input to model 1 and return a predicted mode; if the current mode is A-load mode, the date will be recognized by the semi-melt module, normal-melt module, over-melt module, until the A-load module, and after the recognition is the A-load mode, the testing data will be input to model 4 and return a predicted mode result. After a series of recognition and prediction, we can finish the predicted operating mode of the whole testing set, and generate a set of predicted operating modes.

III. EXPERIMENTS

The data used in the experiment is from the real production of an electrical fused magnesium enterprise. After preprocess the raw data, the training set contains over 1870 samples and the testing set contains 800 samples. But it is only 140 samples that mode shifting occurs in training set, accounting for only 7.38%. If the model is trained according to the original training set data, it will be difficult to make an accurate prediction of the data with shift mode when the testing set data is input. This is a typical sample imbalance problem. Therefore, we propose an ADASYN-XGBoost method to solve the problem caused by

imbalanced data.

A. Experimental Metrics

In this experiment, we use six evaluation metrics, including Hamming Loss, Ranking Loss, Coverage, One Error, Average Precision, and KAPPA coefficient. The six metrics are described in detail as follow:

1) Hamming loss

Hamming Loss is used to calculate the accuracy of multi-label classification models.

$$\text{Hamming Loss} = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{K}|h(x_i)\Delta Y_i| \tag{9}$$

where $h(x_i)$ is the predicted label of sample x_i.

2) Coverage

Coverage is used to calculate how far we need to go down the ranked label list to cover all possibility.

$$\text{Coverage} = \frac{1}{m}\sum_{i=1}^{m}\max_{y\in Yi}\text{rank}(x_i,y,f) - 1 \tag{10}$$

3) One error

One error indicates the proportion of samples whose label with the highest predicted probability value is not in the true label set.

$$\text{One error} = \frac{1}{m}\sum_{i=1}^{m}\max_{y\in Yi}H(x_i) \tag{11}$$

where,

$$H(x_i) = \begin{cases} 0, argmax\big(f(x_i,y)\big) \in Y_i \\ 1, others \end{cases}$$

4) Ranking loss

Ranking loss reflects the proportion of label pairs that are reversely ordered.

$$\text{Ranking loss} = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{|Y_i \parallel \bar{Y}_i|}|\{(y'y'') \mid f(x_i,y')\right.$$
$$\left. \leq f(x_i,y''), y' \in Y_i, y'' \in \bar{Y}_i\}|\right) \tag{12}$$

5) Average precision

Average precision indicates the proportion of labels that are higher than a particular label $y \in Y$.

$$\text{A.P.} = \frac{1}{m}\sum_{i=1}^{m} \times$$
$$\left(\frac{1}{|Y_i|}\sum_{y\in YY}\frac{|\{(y'\in Y_i)|\text{rank}(x_i,y',f)\leq\text{rank}(x_i,y,f)\}|}{\text{rank}(x_i,y,f)}\right) \tag{13}$$

6) KAPPA

The Kappa coefficient is used to test the consistency of the predicted results of the classifier with the actual results.

$$\text{Kappa} = \frac{1}{K}\sum_{k=1}^{K}\frac{p_{ok}-p_{ek}}{1-p_{ek}} \tag{14}$$

where,

$$p_{ok} = \frac{TP_k + TN_k}{m}$$

$$p_{ek} = \frac{TP_k \times (TP_k + FN_k) + TN_k \times (TN_k + FP_k)}{m^2}$$

B. Experimental Results

In this experiment, we select eXtreme Gradient Boosting (XGB), ADASYN-LightGBM (ADASYN-LGB), ADASYN-RF, ADASYN-SVM, ADASYN-XGBoost 5 machine learning algorithms to build the models. Based on the original training set, the best model is built by seeking the optimal combination of model parameters through grid search and other methods. Finally, the built models are evaluated with the testing set. The following are the test results of various machine learning algorithms.

IV. EXPERIMENTAL ANALYSIS

A. Performance Metrics

For each evaluation metric, "↓" indicates better performance for smaller values and "↑" indicates better performance for larger values. Each result consists of a mean and a rank. The best results for the five machine learning algorithms are highlighted in bold. If two or more algorithms achieve the same performance on a given evaluation metric, the value of the corresponding rank is assigned to their average rank. To present the results more clearly, the average rank on all evaluation metrics of each algorithm is calculated in this paper and recorded in the last column of each table.

TABLE I: TESTING RESULTS OF FIVE MACHINE LEARNING ALGORITHMS(FUSED DATA)

Machine Learning Algorithms	Evaluation Metrics										Avg. rank
	Hamming Loss(↓)		Coverage (↓)		Ranking Loss(↓)		One Error(↓)		Average Precision(↑)		
XGB	0.0345	4.5	0.3450	4.5	0.0863	4.5	0.0863	4.5	0.7677	5	4.6
ADASYN-LGB	0.0325	3	0.3250	3	0.0813	3	0.0813	3	0.7866	3	3
ADASYN-RF	0.0345	4.5	0.3450	4.5	0.0863	4.5	0.0863	4.5	0.7765	4	4.4
ADASYN-SVM	0.0315	2	0.3150	2	0.0788	2	0.0788	2	0.7945	2	2
ADASYN-XGBoost	0.0300	**1**	0.3000	**1**	0.0750	**1**	0.0750	**1**	0.8020	**1**	**1**

TABLE II: TESTING RESULTS OF FIVE MACHINE LEARNING ALGORITHMS(SHIFTED DATA)

Machine Learning Algorithms	Evaluation Metrics										Avg. rank
	Hamming Loss(↓)		Coverage (↓)		Ranking Loss(↓)		One Error(↓)		Average Precision(↑)		
XGB	0.3000	2	3.0000	2	0.7500	2	0.7500	2	0.2541	3	2.2
ADASYN-LGB	0.3105	4	3.1053	4	0.7763	4	0.7763	4	0.2497	4	4
ADASYN-RF	0.3263	5	3.2632	5	0.8158	5	0.8158	5	0.2231	5	5
ADASYN-SVM	0.3053	3	3.0526	3	0.7632	3	0.7632	3	0.2555	2	2.8
ADASYN-XGBoost	0.2895	**1**	2.8947	**1**	0.7237	**1**	0.7237	**1**	0.2567	**1**	**1**

In this experiment, Table I and Table II report in detail the results of the proposed experiments with XGB, ADASYN-LGB, ADASYN-RF, ADASYN-SVM, and ADASYN-XGB as classifiers. To compare their effectiveness more

intuitively, the average ranking of each algorithm is given in Fig. 3 and Fig. 4. In Fig. 3, for each performance metric, the average ranking of each algorithm on the testing set is described. For example, for Hamming Loss, XGB and ADASYN-RF have equal values, so their average ranking for Hamming Loss is $(4 + 5)/2 = 4.5$. And Fig. 4 describes the overall average ranking of each algorithm across all experiments. For example, ADASYN-SVM has an average ranking of {2,2.8} on the dataset, so its overall average ranking is $(2 + 2.8)/2 = 2.4$. Based on these experimental results, the following conclusions are drawn:
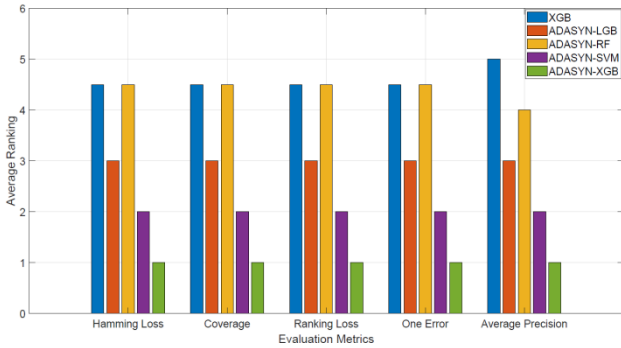

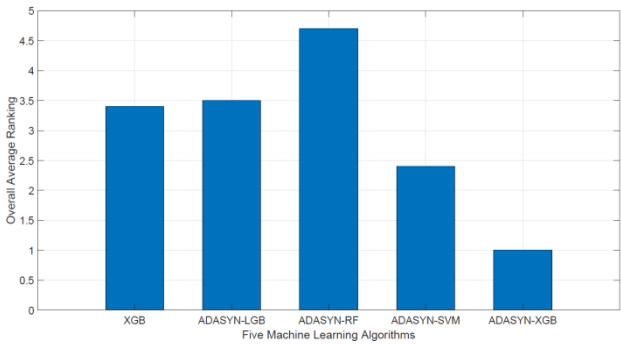Fig. 3. Average ranking of five machine learning algorithms.

Fig. 4. Overall average ranking of five machine learning algorithms.

1) As can be seen from Table I, XGB has the worst performance, ADASYN-RF is the next, and ADASYN-XGBoost has the best performance. Specifically, XGB performs significantly worse than other machine learning algorithms in all performance metrics. The performance metrics were all improved when the models were constructed by the new training set that had been resampled by ADASYN. ADASYN-XGBoost obtained the best performance on Hamming Loss, Coverage, Ranking Loss, One Error, and Average Precision. Fig. 4 also shows that for all evaluation metrics, ADASYN-RF performs the worst, while ADASYN-XGBoost performs the best. This indicates that while optimizing the classification ranking the classification performance can be improved by resampling.

2) For fused data, the ADASYN-XGBoost model outperforms other models, as shown in Table I. Specifically, ADASYN-XGBoost has the lowest Hamming Loss of about 0.03, Ranking Loss of 0.075, and Average Precision of 0.802. And for shifted data, ADASYN-XGBoost also outperforms the other models with 0.2895 Hamming Loss, 0.7237 Ranking Loss, and 0.2567 Average Precision. As can be seen from Fig. 4, ADASYN-XGBoost improves the performance of XGB on all evaluation metrics. These experimental results also verify that increasing the number

of minority class samples in the training set by ADASYN resampling can improve the performance of the XGB algorithm.

3) In summary, the ADASYN-XGBoost algorithm performs the best, the ADASYN-SVM and XGB algorithms are the second, and the other two algorithms are the worst. This fully proves that the algorithm of this paper is very effective in improving the traditional XGB algorithm by ADASYN. It also shows that the based on XGB should not only select features from the feature space, but also consider the influence of minority class samples in imbalanced samples on the classification results, so as to further improve the model performance.

B. Kappa Coefficient

Using the performance metric of Kappa coefficient, the overall accuracy and Kappa of the five machine learning algorithms, from the lowest value of 0.0 to the highest value of 1.0, are shown in Table III.

TABLE III: OVERALL ACCURACY AND KAPPA COEFFICIENT

Machine Learning Algorithms	Fused data	
	Overall Accuracy	Kappa Coefficient
XGB	0.9138	0.8849
ADASYN-LGB	0.9188	0.8916
ADASYN-RF	0.9138	0.8851
ADASYN-SVM	0.9213	0.8949
ADASYN-XGBoost	**0.9250**	**0.9000**

As can be seen from Table III, the overall accuracy and Kappa coefficient of ADASYN-XGBoost are the highest for fused data, while for shifted data, the overall accuracy and Kappa coefficient of ADASYN-XGBoost are improved compared to XGB.

V. CONCLUSION

In this paper, we propose an operating mode prediction model for electrical fused magnesium based on ADASYN-XGBoost. We use the ADASYN resampling technique to generate five different training sets and build five different models based on the traditional XGBoost model by recognizing the difference between current operating modes, and input the testing set data into the respective models for prediction according to the current operating mode. The improved model achieved the best results in six metrics, such as Hamming Loss, Ranking Loss, Coverage, One Error, Average Precision and KAPPA, which effectively improves the accuracy of the prediction of electrical used magnesium operating modes. So it is important to see that high performance has been achieved with the ADASYN-XGBoost method, so this system would become valid and effective for enterprise in the smelting process of fused magnesium.

Our study also has some limitations, such as not conducting experiments on a wider dataset. More industrial models and different sampling methods' effect on the results will be tested in future studies.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

The works of each author in this work are as follows: Zhiyong Su and Yinpeng Ding conducted research and experimenting with time series forecasting. Zhiyong Su also prepared the material for publication. Weijian Kong was consultant and gave advice about techniques, methods to be used in the process of the time series forecasting. All authors have approved the final version.

ACKNOWLEDGMENT

REFERENCES

[1] H. Qi, Y. Zhou, and G. Qi, "A hierarchical ML-kNN method for complex emotion analysis on customer reviews," presented at 2016 International Conference on Mechatronics Engineering and Information Technology, 2016.

[2] W. Weng, D. Wang, C. Chen, J. Wen, and S. Wu, "Label specific features-based classifier chains for multi-label classification," *Digital Object Identifier*, vol. 8, pp. 51265-51275, 2020.

[3] X. Yang, L. Zhou, and L. Wang, "An improved ML-kNN approach based on coupled similarity," *Springer International Publishing*, pp. 77-89, 2016.

[4] G. Q. Wu, R. B. Zheng, Y. J. Tian, and D. L. Liu, "Joint ranking SVM and binary relevance with robust low-rank learning for multi-label classification," *Neural Networks*, vol. 122, pp. 24-39, 2020.

[5] L. Cai, Y. Song, T. Liu, and K. Zhang, "A hybrid BERT model that incorporates label semantics via adjustive attention for multi-label text classification," *Digital Object Identifier*, vol. 8, pp. 152183-152192, 2020.

[6] K. Kimura, M. Kudo, S. Lu *et al.*, "Fast random k-labELsets for large-scale multi-label classification," presented at 2016 23rd International Conference on Pattern Recognition (ICPR), 2016.

[7] L. A. Cabrera-Diego, N. Bessis, and I. Korkontzelos, "Classifying emotions in stack overflow and JIRA using a multi-label approach," *Knowledge-Based Systems*, vol. 195, pp. 105633, 2020.

[8] X. Q. Guan, J. Y. Liang, Y. H. Qian, and J. F. Pang, "A multi-view OVA model based on decision tree for multi-classification tasks," *Knowledge-Based Systems*, vol. 138, pp. 208-219, 2017.

[9] M. Roseberry, B. Krawczyk, Y. Djenouri, and A. Cano, "Self-adjusting k nearest neighbors for continual learning from multi-label drifting data streams," *Neurocomputing*, vol. 442, pp. 10-25, 2021.

[10] N. Rastin, M. Z. Jahromi, and M. Taheri, "A generalized weighted distance k-Nearest Neighbor for multi-label problems," *Pattern Recognition*, vol. 114, p. 107526, 2021.

[11] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M. Hacid, and H. Zeineddine, "An experimental study with imbalanced classification approaches for credit card fraud detection," *IEEE Access*, vol. 7, pp. 93010-93022, 2019.

[12] G. J. S. Litjens, T. Kooi, B. E. Bejnordi *et al.*, "A survey on deep learning in medical image analysis," *Med Image Anal*, vol. 42, pp. 60-88, 2017.

[13] H. Guo, Y. Li, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class imbalanced data: Review of methods and applications," *Expert System*. vol. 73, pp. 220-239, 2017.

[14] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Big Data*, vol.6, p. 27, 2019.

[15] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249-259, 2018.

[16] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE Trans*, vol. 8, pp. 3573-3587, 2018.

[17] F. Charte, A. J. Rivera, M. J. del Jesús, and F. Herrera, "Addressing imbalance in multilabel classification: Measures and random resampling algorithms," *Neurocomputing*, vol. 163, pp. 3-16, 2015.

[18] F. Charte, A. J. Rivera, M. J. del Jesus, and F. Herrera, "MLSMOTE: Approaching imbalanced multilabel learning through synthetic instance generation," *Knowl. Based Syst*, vol. 89, pp. 385-397, 2015.

[19] F. Charte, A. J. Rivera, M. J. del Jesus, and F. Herrera, "Mlenn: A first approach to heuristic multilabel undersampling," *Computer Science,* Springer, vol. 8669, pp. 1-9, 2014.

[20] F. Charte, A. J. Rivera, M. J. del Jesus, and F. Herrera, "Resampling multilabel datasets by decoupling highly imbalanced labels," *Computer Science, Springer*, vol. 9121, pp. 489-501, 2015.

[21] F. Charte, A. J. Rivera, M. J. del Jesus, and F. Herrera, "Remedial-hwr: Tackling multilabel imbalance through label decoupling and data resampling hybridization," *Neurocomputing*, vol. 326-327, pp. 110-122, 2019.

[22] R. M. Pereira, Y. M. G. Costa *et al.*, "MLTL: A multi-label approach for the tomek link undersampling algorithm," *Neurocomputing*, vol. 383, pp. 95-105, 2020.

[23] F. Charte, A. J. Rivera, M. J. del Jesus, and F. Herrera, "Dealing with difficult minority labels in imbalanced mutilabel data sets," *Neurocomputing*, vol. 326-327, pp. 39-53, 2019.
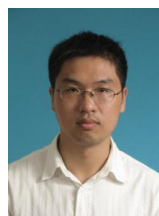
Weijian Kong received the M.E and PhD. degrees from the College of Information Science and Engineering, Northeastern University, Shenyang, China, in 2008 and 2014, respectively. He is currently an associate professor with the College of Information Science and Technology, Donghua University, Shanghai, China. His current research interests include computational intelligence and plant-wide optimization for the complex industrial systems.

Zhiyong Su received the B.S. degree from the College of Materials Science and Engineering, Nanchang Hangkong University, Nanchang, China. He is currently pursuing the M.E. degree from the College of Information Science and Technology, Donghua University, Shanghai, China. His current research interests include data mining and machine learning.

Yinpeng Ding received the bachelor's degree in automation engineering from North China University of Water Resources and Electric Power, Zhengzhou, China in 2020. She is currently pursuing her M.E. degree in control science and engineering at Donghua University. Her research focuses on the application of machine learning to industry.