

INTERNATIONAL JOURNAL
ON INFORMATICS VISUALIZATIONjournal homepage : www.joiv.org/index.php/joiv

Extreme Gradient Boosting Algorithm to Improve Machine Learning Model Performance on Multiclass Imbalanced Dataset

Yoga Pristyanto ^{a,*}, Zulfikar Mukarabiman ^a, Anggit Ferdita Nugraha ^a^a Faculty of Computer Science, Universitas Amikom Yogyakarta, Yogyakarta, Indonesia

Corresponding author: *yoga.pristyanto@amikom.ac.id

Abstract— Unbalanced conditions in the dataset often become a real-world problem, especially in machine learning. Class imbalance in the dataset is a condition where the number of minority classes is much smaller than the majority class, or the number is insufficient. Machine learning models tend to recognize patterns in the majority class more than in the minority class. This problem is one of the most critical challenges in machine learning research, so several methods have been developed to overcome it. However, most of these methods only focus on binary datasets, so few methods still focus on multiclass datasets. Handling unbalanced multiclass is more complex than handling unbalanced binary because it involves more classes than binary class datasets. With these problems, we need an algorithm with features that can support adjustments to the difficulties that arise in multiclass unbalanced datasets. One of the algorithms that have features for adjustment is the ensemble algorithm, namely Xtreme Gradient Boosting. Based on the research, our proposed method with Xtreme Gradient Boosting showed better results than the other classification and ensemble algorithms on eight datasets with five evaluation metrics indicators such as balanced accuracy, the geometric-mean, multiclass area under the curve, true positive rate, and true negative rate. In future research, we suggest combining methods at the data level and Xtreme Gradient Boosting. With the performance increase in Xtreme Gradient Boosting, it can be a solution and reference in the case of handling multiclass imbalanced problems. Besides, we also recommended testing with datasets in the form of categorical and continuous data.

Keywords— Class imbalanced; ensemble algorithm; XGBoost; classification; multiclass.

Manuscript received 12 Aug. 2022; revised 31 Dec. 2022; accepted 12 Jan. 2023. Date of publication 10 Sep. 2023.
International Journal on Informatics Visualization is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.



I. INTRODUCTION

Class imbalance in the dataset is a situation or condition where the value of the minority class is much smaller than that of the majority class or so inadequate that the model recognizes patterns in the majority class more than the minority class. For example, in the medical world, the amount of data from patients with diabetes is less than from patients who do not have diabetes. This problem can result in an inaccurate classification and be fatal if implemented in the real world [1]–[3]. The problem of imbalanced datasets is one of the most critical challenges in the machine learning research community. Various methods have been developed to overcome these problems, such as resampling methods, cost-sensitive approaches, ensemble learning algorithms, kernel-based methods, and active learning methods. These techniques can be categorized into several approaches based on how to overcome the problem of imbalanced data. The first approach at the algorithm level is to create or modify an algorithm to consider the significance of the positive class.

Algorithm-based approaches include cost-sensitive methods and recognition-based approaches [4], [5]. The second approach, namely the data level, is carried out at the pre-data processing stage. The class distribution in the data is rebalanced to reduce the effect of the number of majority classes being too dominant in the learning process. One alternative to improving the model's performance in the imbalance class is to use the ensemble method. The ensemble method, in principle, combines a set of classifiers trained to create a mixed classification model with good performance so that the classifier formed is more accurate than the original classifier in performing a classification. Boosting, Stacking, and Bagging are the most popular ensemble methods used [6]–[10].

The majority of research conducted focused on binary imbalanced datasets. The multiclass imbalance learning problem is much more challenging than the binary scenario because the decision boundary involves distinguishing between more classes [2], [11]. Unfortunately, directly applying the proposed method to deal with binary dataset

imbalances on multiclass datasets may not be valid. In addition, imbalanced multiclass also often appear in real-world problems. Three types of challenges are associated with imbalanced class conditions in multiclass datasets: one majority class and many minority classes, one minority and many majority classes, and many minorities and many majority classes. In addition, the slope of the distribution of instances among classes is one of many sources of difficulty for classification algorithms to deal with multiclass unbalanced data sets. Other difficulties in the data structure are always present, such as overlapping classes, slight separation (a minority class may consist of several sub-concepts), and small sample sizes (lack of representative minority examples) [12]–[14].

The following are previous studies that have been carried out related to handling multiclass imbalanced datasets. Research conducted by Abdi and Hashemi [15] proposed a new method at the data level. The method is named Mahalanobis Distance Oversampling (MDO). This method creates synthetic data with the same Mahalanobis Distance from the corresponding class average. This method has been tested on 20 multiclass imbalanced datasets. With MAUC, Geometric Mean (G-Mean), precision, recall, and f-measure metrics, this method can significantly outperform Borderline-SMOTE, SMOTE, and ADASYN with the base classifiers KNN and RIPPER. Ding et al. [16] proposed a new method at the algorithm level in their research. The method is a Weighted Online Sequential Extreme Learning Machine with Kernels (WOS-ELMK) for Class Balanced Learning (CIL). Different from the Online Sequential Extreme Learning Machine (OS-ELM) method, which generally uses random feature mapping, WOS-ELMK uses kernel mapping for online class imbalance learning to overcome the problem of hidden node non-optimization, which is often associated with random feature mapping. This method has been tested on 17 binary and eight multiclass imbalanced datasets. With the g-mean metric, WOS-ELMK can outperform VWOS-ELM and KOS-ELM. Meanwhile, a study by Bi and Zhang [17] proposed the Diversified Error-Correcting Output Codes (DECOC) method. The DECOC method is a method that combines Error-Correcting Output Codes (ECOC) to overcome class imbalances and various ensemble learning frameworks to find the best classification algorithm for each individual sub-dataset derived from the resampling process from the original data. DOC has been tested on 19 datasets with metric accuracy; AUC, g-mean, and f-measure outperformed 17 state-of-art methods on imbalance learning. The limitation of the DECOC method is that it is time-consuming or takes much time in the execution process of the method.

On the other hand, Krawczyk, Koziarski, and Woźniak [18] used a method called Multiclass Radial-Based Oversampling (MC-RBO). This method performs oversampling that does not use NNs from minority agencies, so it differs from algorithms such as SMOTE. This method has been tested on 20 multiclass imbalanced datasets. With Average Accuracy (AvAcc), CBA, MGM, and CEN metrics, the process can outperform SMOTE-all, S-SMOTE, MDO, SMOM, HTREE, NB-SVM, and AdaBoost.NC, SMOTEBag, OVO-RUS, OVO-SMOTE, OVO-BSMOTE, OVO-ADASYN [18]. Research related to handling class imbalances was also

carried out by Fernandes and de Carvalho [19]. This study uses the method of multi-objective evolutionary algorithm (MOEA). This method can outperform SMOTEBagging, RUSBagging, ROSBagging, AadaBoostM1, and RUSBoost in terms of the g-mean average.

Most previous research was limited to handling class imbalances in class binary datasets. Therefore, this study will focus on imbalanced classes in multiclass datasets. On the other hand, imbalanced multiclass is often found in real-world problems. The novel proposed method in this study uses the XGBoost algorithm. XGBoost is an ensemble method that can be used for imbalanced multiclass problems. The two basic ideas why using XGBoost first are to maintain the naturalization of the dataset. The second is to create a reliable machine-learning model in imbalanced multiclass conditions. This research's main novelty and contribution can be a solution to handling the imbalanced class on the machine learning model, especially in the multiclass imbalanced case, to maintain the naturalization of the datasets. Furthermore, the proposed method will be compared with several popular methods, single-classifiers, and other ensemble models. This comparison aims to measure how well the XGBoost algorithm performs compared to popular existing models, especially in multiclass imbalances.

The rest of this paper is organized as follows. Section II describes the methods used to handle the imbalanced class. In Section III, we discuss the results of the proposed method. Finally, Section IV presents our conclusions.

II. MATERIAL AND METHODS

This research was conducted through several stages. Fig 1. below shows the steps of research that have been done.

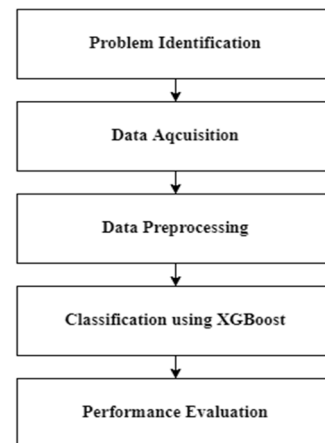


Fig 1 Research Workflow

The following steps are the explanation for each process:

A. Problem Identification

This part identifies the existing problems in the form of background, formulation, problem limitation, objectives, benefits, and methodology.

B. Datasets

The dataset used in this study is a specific public dataset in the imbalance dataset obtained from the KEEL and UCI Machine Learning repositories consisted of nine imbalanced

datasets with varied ratios and classes. The following Table I and Table II show the details of the dataset used.

TABLE I
DATASETS

| Name | Code | Instances | Features | Classes |
|-------------------|------|-----------|----------|---------|
| car | D1 | 1728 | 7 | 4 |
| contraceptive | D2 | 1473 | 10 | 3 |
| glass | D3 | 214 | 10 | 6 |
| hayes-roth | D4 | 132 | 6 | 3 |
| new-thyroid | D5 | 215 | 6 | 3 |
| pageblocks | D6 | 5473 | 11 | 5 |
| winequality-red | D7 | 1599 | 12 | 6 |
| winequality-white | D8 | 4898 | 12 | 7 |
| yeast | D9 | 1484 | 10 | 10 |

TABLE II
CLASS DISTRIBUTION OF DATASETS

| Name | Code | Class Distribution |
|-------------------|------|----------------------------------|
| car | D1 | 1210:384:69:65 |
| contraceptive | D2 | 629:511:333 |
| glass | D3 | 76:70:29:17:13:9 |
| hayes-roth | D4 | 51:51:30 |
| new-thyroid | D5 | 150:35:30 |
| pageblocks | D6 | 4913:329:115:88:28 |
| winequality-red | D7 | 681:638:199:53:18:10 |
| winequality-white | D8 | 2198:1457:880:175:163:20:5 |
| yeast | D9 | 463:429:244:163:51:44:35:30:20:5 |

C. Data Preprocessing

At this stage, checking for missing values, instance duplication, data inconsistency, and data outliers is carried out. However, this study did not find missing values, duplicate instances, inconsistent data, or outliers.

D. Classification using Extreme Gradient Boosting

This study proposed the Extreme Gradient Boosting (XGBoost) ensemble method to handle multiclass imbalanced datasets. This method is a tree-based boosting method derivative of Gradient Boosting but in an optimized version [20]. Fig 2. below is an illustration of the XGBoost model:

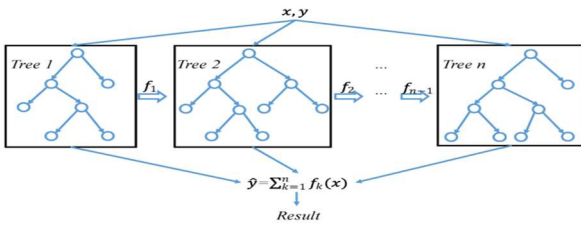


Fig. 2 XGBoost model illustration

Fig 2. shows that the XGBoost structure is similar to the Random Forest structure. However, each tree model in XGBoost minimizes the residuals from the previous tree model. Unlike the GBDT, which generally uses the first derivative of error information, XGBoost performs a second-order Taylor expansion on the cost function and uses the first and second derivatives. Each tree is trained with a dataset and learns the residuals from the previous three to be minimized gradually. Then the results from the training of each tree are summed using the following equations.

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^K f_k(x_i), f_k \in \mathcal{F} \quad (1)$$

Where F is a function in the functional space F , where $F = \{f(x) = w_q(x)\} (q : R^m \rightarrow T, w \in R^T)$ being the set of all

possible classification trees, where q represents the structure of each tree, maps an instance to the appropriate leaf index, T is the number of leaves in the tree, w is the weight of the leaves, and K represents the number of trees. The equation below represents the objective function for optimization, trained in an additive manner by adding f_t , which helps to minimize the objective, where $\hat{y}_i^{(t-1)}$ represents the prediction of instance i at iteration $t-1$, $l(y_i, \hat{y}_i^{(t-1)})$ is the training loss function and Ω is the regularization condition. The following process is calculating the objective function using the following equation.

$$\mathcal{L}^{(t)} = \sum_i^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \quad (2)$$

Using the following equation, the next step is to use a complexity control function to prevent overfitting.

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2 \quad (3)$$

This boosting model differs from Gradient Boost, which builds trees in series. XGBoost builds trees in parallel, similar to the Random Forest method. XGBoost builds trees in parallel using all CPU cores during training, resulting in shorter compute time [21]–[23].

E. Evaluation

The last stage in this research is to evaluate the resulting classification model. The evaluation indicators used in this study include balanced accuracy, Multiclass Area Under Curve Score (MAUC), True Positive Rate (TPR), True Negative Rate (TNR), and geometric mean (G-Mean) [24]. These five are comprehensive indicators to be used to evaluate the classification model in cases of class imbalance in the dataset [25], [26]. XGBoost evaluation results were compared with classification algorithms such as Logistic Regression, Decision Tree, Gaussian Naïve Bayes, K-Nearest Neighbor, and Support Vector Machine. In addition, we also compare with the ensemble algorithm, namely bagging and stacking. Bagging algorithm structure, use the Decision Tree base classifier. The Stacking algorithm structure uses the K-Nearest Neighbor algorithm, Support Vector Machine, and Decision Tree. The tests of all comparison algorithms go through the same process flow and datasets.

III. RESULTS AND DISCUSSION

In this section, the discussion focuses more on evaluating the performance of the XGBoost algorithm and comparing its performance with other classification models based on the value of balanced accuracy, g-mean, sensitivity, specificity, and MAUC. The data set in the XGBoost test scenario would be divided into two parts: the training set and the test set. The proportion of the distribution is as follows the training of 80% of the total data. Meanwhile, the test data is 20% of the total data. The followings are the evaluation results for each indicator tested on XGBoost and the results of a comparison with several other classification models such as Decision Tree (DT), Gaussian Naïve Bayes (NB), Support Vector Machine (SVM), Bagging, and Stacking.

A. Balanced Accuracy Score

In the balanced accuracy evaluation metric, the performance of the XGBoost algorithm is relatively good. The following Table III is the comparison result of the

XGBoost balanced accuracy score when compared to DT, Gaussian NB, SVM, bagging, and stacking algorithms.

TABLE III
COMPARISON OF BALANCED ACCURACY SCORE

| Dataset | Balanced Accuracy | | | | | |
|---------|-------------------|--------------|-------|---------|----------|--------------|
| | DT | NB | SVM | Bagging | Stacking | XGBoost |
| D1 | 0.930 | 0.517 | 0.882 | 0.969 | 0.934 | 0.980 |
| D2 | 0.449 | 0.483 | 0.539 | 0.487 | 0.453 | 0.552 |
| D3 | 0.566 | 0.478 | 0.704 | 0.637 | 0.544 | 0.770 |
| D4 | 0.936 | 0.803 | 0.795 | 0.903 | 0.906 | 0.970 |
| D5 | 0.894 | 0.989 | 0.894 | 0.857 | 0.894 | 1.000 |
| D6 | 0.823 | 0.737 | 0.565 | 0.824 | 0.786 | 0.849 |
| D7 | 0.326 | 0.370 | 0.282 | 0.312 | 0.320 | 0.364 |
| D8 | 0.355 | 0.272 | 0.236 | 0.381 | 0.346 | 0.395 |
| D9 | 0.392 | 0.487 | 0.485 | 0.427 | 0.402 | 0.446 |

Based on Table III above shows that the balanced accuracy value of the XGBoost algorithm can outperform the Decision Tree, Gaussian NB, SVM, bagging, and stacking datasets on the dataset car, contraceptive, glass, hayes-roth, new-thyroid, page blocks, wine quality-red, and wine quality-white. It even hits a perfect score of 1.00 on the new-thyroid dataset. However, in the yeast dataset, the XGBoost algorithm is less valuable than other algorithms. In the table, the values in bold are each dataset's highest balanced accuracy values.

B. Geometric Mean Score

The second evaluation indicator is based on the geometric mean values. On the geometric mean metric, the performance of the XGBoost algorithm is relatively good compared to others. Table IV compares the geometric mean XGBoost to the Decision Tree, Gaussian NB, SVM, bagging, and stacking algorithms.

TABLE IV
COMPARISON OF GEOMETRIC MEAN SCORE

| Dataset | Geometric Mean | | | | | |
|---------|----------------|--------------|-------|---------|----------|--------------|
| | DT | NB | SVM | Bagging | Stacking | XGBoost |
| D1 | 0.959 | 0.672 | 0.933 | 0.981 | 0.961 | 0.989 |
| D2 | 0.573 | 0.596 | 0.645 | 0.604 | 0.576 | 0.657 |
| D3 | 0.716 | 0.646 | 0.817 | 0.772 | 0.697 | 0.860 |
| D4 | 0.948 | 0.841 | 0.847 | 0.921 | 0.923 | 0.975 |
| D5 | 0.916 | 0.990 | 0.916 | 0.889 | 0.889 | 1.000 |
| D6 | 0.889 | 0.828 | 0.725 | 0.891 | 0.866 | 0.907 |
| D7 | 0.542 | 0.575 | 0.501 | 0.530 | 0.537 | 0.579 |
| D8 | 0.568 | 0.490 | 0.460 | 0.592 | 0.561 | 0.605 |
| D9 | 0.605 | 0.668 | 0.676 | 0.633 | 0.613 | 0.650 |

Table IV shows that the geometric mean value of the XGBoost algorithm could outperform the Decision Tree, Gaussian NB, SVM, bagging, and stacking datasets on the car, contraceptive, glass, hayes-roth, new-thyroid, page blocks dataset, wine quality-red, and wine quality-white. It even hits the perfect score of 1.00 on the new-thyroid dataset. However, in the yeast dataset, the XGBoost algorithm has a lower value than other algorithms. In the table, the values in bold are each dataset's highest g-mean score.

C. Multiclass Area Under Curve Score (MAUC)

The third evaluation indicator is based on the MAUC values. On the MAUC metric, the performance of the XGBoost algorithm is relatively good. Table V compares the MAUC XGBoost value to the Decision Tree, Gaussian NB, SVM, bagging, and stacking algorithms.

TABLE V
COMPARISON OF MAUC SCORE

| Dataset | Multiclass Area Under Curve | | | | | |
|---------|-----------------------------|-------|--------------|---------|----------|--------------|
| | DT | NB | SVM | Bagging | Stacking | XGBoost |
| D1 | 0.959 | 0.870 | 0.998 | 0.999 | 0.998 | 0.999 |
| D2 | 0.592 | 0.635 | 0.703 | 0.675 | 0.648 | 0.725 |
| D3 | 0.736 | 0.777 | 0.893 | 0.888 | 0.832 | 0.860 |
| D4 | 0.948 | 0.926 | 0.931 | 0.963 | 0.917 | 0.969 |
| D5 | 0.917 | 0.997 | 0.997 | 0.999 | 0.997 | 1.000 |
| D6 | 0.894 | 0.967 | 0.980 | 0.957 | 0.977 | 0.992 |
| D7 | 0.613 | 0.703 | 0.831 | 0.696 | 0.793 | 0.808 |
| D8 | 0.632 | 0.757 | 0.823 | 0.744 | 0.772 | 0.865 |
| D9 | 0.663 | 0.802 | 0.875 | 0.774 | 0.857 | 0.772 |

Table V shows that the MAUC value of the XGBoost algorithm could outperform Decision Tree, Gaussian NB, SVM, bagging, and stacking datasets on the dataset car, contraceptive, hayes-roth, new-thyroid, page blocks, and wine quality-white. It even hits a perfect score of 1.00 on the new-thyroid dataset. However, on the glass, wine quality-red, and yeast dataset, the XGBoost algorithm has a lower value than the other algorithms. The MAUC results are clarified in Table IV, where the value in bold is the highest in each dataset.

D. True Positive Rate (TPR)

The fourth evaluation indicator is based on the TPR values. On the sensitivity metric, the performance of the XGBoost algorithm is relatively good. Table VI shows a comparison of XGBoost's TPR values when compared to the Decision Tree, Gaussian NB, SVM, bagging, and stacking algorithms

TABLE VI
COMPARISON OF TRUE POSITIVE RATE

| Dataset | True Positive Rate | | | | | |
|---------|--------------------|--------------|-------|---------|----------|--------------|
| | DT | NB | SVM | Bagging | Stacking | XGBoost |
| D1 | 0.930 | 0.517 | 0.882 | 0.969 | 0.934 | 0.980 |
| D2 | 0.449 | 0.483 | 0.539 | 0.487 | 0.453 | 0.552 |
| D3 | 0.566 | 0.478 | 0.704 | 0.637 | 0.544 | 0.770 |
| D4 | 0.936 | 0.803 | 0.795 | 0.903 | 0.906 | 0.970 |
| D5 | 0.894 | 0.989 | 0.894 | 0.857 | 0.894 | 1.000 |
| D6 | 0.823 | 0.737 | 0.565 | 0.824 | 0.786 | 0.849 |
| D7 | 0.326 | 0.370 | 0.282 | 0.312 | 0.320 | 0.364 |
| D8 | 0.355 | 0.272 | 0.236 | 0.381 | 0.346 | 0.395 |
| D9 | 0.392 | 0.487 | 0.485 | 0.427 | 0.402 | 0.446 |

Table VI shows that the TPR value of the XGBoost algorithm can outperform the Decision Tree, Gaussian NB, SVM, bagging, and stacking datasets on the car, contraceptive, glass, hayes-Roth, new-thyroid, page blocks, and wine quality white-datasets. It even hits a perfect score of

1.00 on the new-thyroid dataset. However, in the wine quality-red and yeast dataset, the XGBoost algorithm has a lower value than the other algorithms. The results of the comparison of TPR values are clarified in Table V, where the value in bold is the highest in each dataset.

E. True Negative Rate (TNR)

The fifth evaluation indicator is based on the TNR values. On the sensitivity metric, the performance of the XGBoost algorithm is relatively good. Table VII compares the true negative rate (TNR) on XGBoost to the Decision Tree, Gaussian NB, SVM, bagging, and stacking algorithms.

TABLE VII
COMPARISON OF TRUE NEGATIVE RATE

| Dataset | True Negative Rate | | | | | |
|---------|--------------------|-------|-------|---------|----------|--------------|
| | DT | NB | SVM | Bagging | Stacking | XGBoost |
| D1 | 0.988 | 0.873 | 0.987 | 0.993 | 0.989 | 0.997 |
| D2 | 0.731 | 0.736 | 0.771 | 0.750 | 0.732 | 0.783 |
| D3 | 0.905 | 0.874 | 0.947 | 0.937 | 0.895 | 0.960 |
| D4 | 0.960 | 0.881 | 0.903 | 0.939 | 0.940 | 0.980 |
| D5 | 0.940 | 0.991 | 0.940 | 0.923 | 0.940 | 1.000 |
| D6 | 0.961 | 0.930 | 0.931 | 0.962 | 0.954 | 0.969 |
| D7 | 0.899 | 0.893 | 0.890 | 0.902 | 0.900 | 0.920 |
| D8 | 0.909 | 0.884 | 0.896 | 0.919 | 0.908 | 0.927 |
| D9 | 0.935 | 0.917 | 0.942 | 0.938 | 0.935 | 0.948 |

In addition, Table VII shows that the TNR value of the XGBoost algorithm can outperform the Decision Tree, Gaussian NB, SVM, bagging, and stacking algorithms on all datasets. It even hits a perfect score of 1.00 on the new-thyroid dataset. The results of the comparison of TNR values are explained in Table VII, where the values in bold are the highest in each dataset.

Based on the evaluation conducted using the five indicators above. Overall, XGBoost can perform better than other models on eight datasets, including car, contraceptive, glass, hayes-Roth, new-thyroid, page blocks, wine quality-red, and wine quality-white. At the same time, the yeast dataset XGBoost model still needs to be better compared to other models. The point that makes XGBoost superior can be seen in the TPR and TNR values. These two indicators are essential in cases of class imbalance in datasets, especially multiclass datasets. When a model can produce good performance on the TPR and TNR value indicators, the model can also produce good scores on other indicators. Therefore, XGBoost can be a solution in the case of multiclass imbalanced datasets by maintaining the natural condition of the dataset. However, XGBoost has drawbacks for multiclass datasets with a class number greater than ten classes.

IV. CONCLUSION

Based on the research conducted, there are three conclusions. First, the proposed XGBoost method has a better evaluation value than the Decision Tree, Gaussian NB, and SVM classification methods. Second, the XGBoost model has a better evaluation value than the ensemble bagging method based on the Decision Tree classifier and stacking based on the KNN, SVM, and Decision Tree algorithms. Third, the

XGBoost method has problems dealing with the class imbalance in datasets with multiple types greater than ten. However, this proposed model still needs to improve if it handles imbalanced cases with larger classes equal to ten. An example is the case of the yeast dataset, which has the largest number of classes compared to the other eight datasets.

In future research, researchers are suggested to combine methods at the data level, such as undersampling or oversampling and XGBoost. The basic idea is sampling to handle datasets with more than ten classes. In addition, it is also recommended to perform tests with categorical and continuous data types to ensure the reliability of the XGBoost model.

ACKNOWLEDGMENT

Thanks to the Department of Research and Community Service, Universitas Amikom Yogyakarta for funding and supporting this research.

REFERENCES

- [1] B. Kim and J. Kim, "Adjusting decision boundary for class imbalanced learning," *IEEE Access*, vol. 8, pp. 81674–81685, 2020, doi: 10.1109/ACCESS.2020.2991231.
- [2] C. L. Liu and P. Y. Hsieh, "Model-Based Synthetic Sampling for Imbalanced Data," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 8, pp. 1543–1556, 2020, doi: 10.1109/TKDE.2019.2905559.
- [3] Q. Wang, W. Cao, J. Guo, J. Ren, Y. Cheng, and D. N. Davis, "DMP_MI: An effective diabetes mellitus classification algorithm on imbalanced data with missing values," *IEEE Access*, vol. 7, pp. 102232–102238, 2019, doi: 10.1109/ACCESS.2019.2929866.
- [4] A. S. Tarawneh, A. B. A. Hassanat, K. Almohammadi, D. Chetverikov, and C. Bellinger, "SMOTEFUNA: Synthetic Minority Over-Sampling Technique Based on Furthest Neighbour Algorithm," *IEEE Access*, vol. 8, pp. 59069–59082, 2020, doi: 10.1109/ACCESS.2020.2983003.
- [5] S. Garcia, Z. L. Zhang, A. Altalhi, S. Alshomrani, and F. Herrera, "Dynamic ensemble selection for multiclass imbalanced datasets," *Inf. Sci. (Nijl.)*, vol. 445–446, pp. 22–37, 2018, doi: 10.1016/j.ins.2018.03.002.
- [6] S. Huda *et al.*, "An Ensemble Oversampling Model for Class Imbalance Problem in Software Defect Prediction," *IEEE Access*, vol. 6, pp. 24184–24195, 2018, doi: 10.1109/ACCESS.2018.2817572.
- [7] X. Xu, W. Chen, and Y. Sun, "Over-sampling algorithm for imbalanced data classification," *J. Syst. Eng. Electron.*, vol. 30, no. 6, pp. 1182–1191, 2019, doi: 10.21629/JSEE.2019.06.12.
- [8] G. G. Warsi, S. Saini, and K. Khatri, "Ensemble Learning on Diabetes Data Set and Early Diabetes Prediction," in *2019 International Conference on Computing, Power and Communication Technologies (GUCON)*, 2019, pp. 182–187.
- [9] M. A. Febriantono, S. H. Pramono, Rahmadwati, and G. Naghdy, "Classification of multiclass imbalanced data using cost-sensitive decision tree c5.0," *LAES Int. J. Artif. Intell.*, vol. 9, no. 1, pp. 65–72, 2020, doi: 10.11591/ijai.v9.i1.pp65-72.
- [10] T. M. Khan, S. Xu, Z. G. Khan, and M. U. Chishti, "Implementing multilabeling, ADASYN, and relief techniques for classification of breast cancer diagnostic through machine learning: Efficient computer-aided diagnostic system," *J. Healthc. Eng.*, vol. 2021, 2021, doi: 10.1155/2021/5577636.
- [11] Y. Yang, P. Xiao, Y. Cheng, W. Liu, and Z. Huang, "Ensemble Strategy for Hard Classifying Samples in Class-Imbalanced Data Set," *Proc. - 2018 IEEE Int. Conf. Big Data Smart Comput. BigComp 2018*, pp. 170–175, 2018, doi: 10.1109/BigComp.2018.00033.
- [12] J. Mathew, C. K. Pang, M. Luo, and W. H. Leong, "Classification of Imbalanced Data by Oversampling in Kernel Space of Support Vector Machines," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 9, pp. 4065–4076, 2018, doi: 10.1109/TNNLS.2017.2751612.
- [13] H. R. Sanabila and W. Jatmiko, "Ensemble Learning on Large Scale Financial Imbalanced Data," *2018 Int. Work. Big Data Inf. Secur. IWBI 2018*, pp. 93–98, 2018, doi: 10.1109/IWBI.2018.8471702.
- [14] R. Ghorbani and R. Ghousi, "Comparing Different Resampling Methods in Predicting Students' Performance Using Machine

- Learning Techniques.” *IEEE Access*, vol. 8, pp. 67899–67911, 2020, doi: 10.1109/ACCESS.2020.2986809.
- [15] L. Abdi and S. Hashemi, “To Combat Multiclass Imbalanced Problems by Means of Over-Sampling Techniques,” *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 238–251, 2016, doi: 10.1109/TKDE.2015.2458858.
- [16] S. Ding *et al.*, “Kernel based online learning for imbalance multiclass classification,” *Neurocomputing*, vol. 277, pp. 139–148, 2018, doi: 10.1016/j.neucom.2017.02.102.
- [17] J. Bi and C. Zhang, “An empirical comparison on state-of-the-art multiclass imbalance learning algorithms and a new diversified ensemble learning scheme,” *Knowledge-Based Syst.*, vol. 158, no. May, pp. 81–93, 2018, doi: 10.1016/j.knsys.2018.05.037.
- [18] B. Krawczyk, M. Koziarski, and M. Woźniak, “Radial-Based Oversampling for Multiclass Imbalanced Data Classification,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, no. 8, pp. 2818–2831, 2020, doi: 10.1109/TNNLS.2019.2913673.
- [19] E. R. Q. Fernandes and A. C. P. L. F. de Carvalho, “Evolutionary inversion of class distribution in overlapping areas for multiclass imbalanced learning,” *Inf. Sci. (Ny)*, vol. 494, pp. 141–154, 2019, doi: 10.1016/j.ins.2019.04.052.
- [20] O. Sagi and L. Rokach, “Approximating XGBoost with an interpretable decision tree,” *Inf. Sci. (Ny)*, vol. 572, pp. 522–542, 2021, doi: 10.1016/j.ins.2021.05.055.
- [21] A. B. Parsa, A. Movahedi, H. Taghipour, S. Derrible, and A. (Kouros) Mohammadian, “Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis,” *Accid. Anal. Prev.*, vol. 136, no. October 2019, p. 105405, 2020, doi: 10.1016/j.aap.2019.105405.
- [22] J. Nobre and R. F. Neves, “Combining Principal Component Analysis, Discrete Wavelet Transform and XGBoost to trade in the financial markets,” *Expert Syst. Appl.*, vol. 125, pp. 181–194, 2019, doi: 10.1016/j.eswa.2019.01.083.
- [23] W. Wang, G. Chakraborty, and B. Chakraborty, “Predicting the risk of chronic kidney disease (Ckd) using machine learning algorithm,” *Appl. Sci.*, vol. 11, no. 1, pp. 1–17, 2021, doi: 10.3390/app11010202.
- [24] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, “Learning from class-imbalanced data: Review of methods and applications,” *Expert Syst. Appl.*, vol. 73, pp. 220–239, 2017, doi: 10.1016/j.eswa.2016.12.035.
- [25] S. Wang, L. L. Minku, and X. Yao, “A Systematic Study of Online Class Imbalance Learning with Concept Drift,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 10, pp. 4802–4821, 2018, doi: 10.1109/TNNLS.2017.2771290.
- [26] F. Shakeel, A. S. Sabhitha, and S. Sharma, “Exploratory review on class imbalance problem: An overview,” 2017, doi: 10.1109/ICCCNT.2017.8204150.