Recursive Journal of Informatics
Vol. 1, No. 2, September 2023

p-ISSN 2963-5551 https://journal.unnes.ac.id/sju/index.php/rji e-ISSN 2986-6588

Machine Learning Model Using Extreme Gradient Boosting (XGBoost) Feature Importance and Light Gradient Boosting Machine (LightGBM) to Improve Accurate Prediction of Bankruptcy

Risma Moulidya Syafei[1], Devi Ajeng Efrilianda[2]
[1,2]Computer Science Department, Faculty of Mathematics and Natural Sciences,
Universitas Negeri Semarang, Indonesia

Abstract. Humans have limitations in processing and analyzing large amounts of data in a short time, including in terms of analyzing bankruptcy data. Bankruptcy data is one of the data that has complex information, so it requires technology that can assist in the process of analyzing and processing data more quickly and efficiently. Data science technology enables data processing and analysis on a large scale, using parallel processing techniques. Parallel processing can be implemented in machine learning models.
Purpose: Using parallel processing techniques, data science technologies enable data processing and analysis at scale. Parallel processing can be implemented in machine learning models. Therefore, this study aims to implement a machine learning model using the Light Gradient Boosting Machine (LightGBM) classification algorithm which is optimized using Extreme Gradient Boosting (XGBoost) Feature Importance to increase the accuracy of bankruptcy prediction.
Methods/Study design/approach: Bankruptcy prediction is carried out by applying LightGBM as a classification model and optimized using the XGBoost algorithm as a Feature Importance technique to improve model accuracy. the dataset used is the Taiwanese Bankruptcy dataset collected from the Taiwan Economic Journal for 1999 to 2009 and has 6,819 data. Taiwanese Bankruptcy is unbalanced data, so this study applies random oversampling. **Result/Findings:** The results obtained after going through the model testing process using the confusion matrix obtained an accuracy of the performance of LightGBM+XGBoost Feature Importance of 99.227%.
Novelty/Originality/Value: So, it can be concluded that the implementation of XGBoost Feature Importance can be used to improve LightGBM's performance in bankruptcy prediction.

Keywords: Bankruptcy, LightGBM, XGBoost Feature Importance, Machine Learning, Data Science
Received July 11, 2023 / **Revised** July 17, 2023 / **Accepted** September 14, 2023

INTRODUCTION

In the digital era, the data generated continues to increase exponentially [1]. Where it is difficult for humans to manually process and analyze large amounts of data in an efficient time. According to [2] humans have limitations in processing complex information and analyzing large amounts of data, including in terms of analyzing bankruptcy (Bankruptcy). Therefore, data science technology is needed. Data science technologies enable large-scale processing and analysis of data, whether through algorithms and distributed computing or parallel processing techniques. Parallel processing techniques in the context of data science refer to the ability to process data simultaneously using several computational resources in parallel [3]. In parallel processing, a complex task is divided into smaller tasks that can be executed independently, and then these tasks are executed concurrently by several computing resources. By utilizing several computing resources simultaneously, data processing can be done more quickly, so that the time needed to analyze data can be reduced. Parallel processing techniques can also be applied in machine learning models.

Machine learning is a model that improves system performance by learning from experience through computational methods [4]. In computer systems, experience exists in the form of data, and the main task of machine learning is to develop learning algorithms that build models from data. Data problem-solving often uses classification techniques. Classification techniques can be used to group data to make it easier

[1]*Corresponding author.
Email addresses: syafeirisma@students.unnes.ac.id (Syafei)

DOI: 10.15294/rji.v1i2.71229

to detect abnormal data. One of the classification techniques commonly used for prediction is the Light Gradient Boosting Machine (LightGBM) algorithm.

LightGBM is a gradient-boosting framework for data classification and prediction [5]. Improved classification performance is achieved using advanced classifiers and the selection of the most important features for classification. Therefore, to get the best accuracy and increase efficiency and reduce computational complexity, a feature selection process needs to be carried out [6]. The feature selection process can be done by implementing the feature importance technique. Feature importance is a technique for selecting a subset of optimal features/attributes using certain criteria. The process of selecting attributes using feature importance is expected to be able to reduce the number of irrelevant features, eliminate data redundancy, eliminate features that contain noise, and will have the effect of increasing speed in processing data, increasing learning accuracy, and producing good predictive model performance. The feature importance selection technique can be performed using the Extreme Gradient Boosting (XGBoost) algorithm. XGBoost has advantages in performance and time complexity as well as affordable memory which has been used in various research fields since it was proposed starting from the medical, financial, and metagenomic fields.

Predicting corporate bankruptcy requires detection stages from bankruptcy datasets [7]. The detected bankruptcy dataset will provide information to improve the accuracy of the results. Datasets that have data imbalance characters will affect the accuracy of the prediction results. Overcoming the problem of an unbalanced number of classes can be done using the Random Oversampling (ROS) method [8]. ROS is the provision of data from the minority class into the training data randomly. This process of providing data is repeated until the number of minority class data is equal to the number of the majority class. The first step is to calculate the difference between the majority class and the minority class. After that, it is repeated as many times as the results of calculating the difference in data while reading the minority class data randomly and entering it into the training data. Related research was conducted by Quang in 2022, using three ensemble algorithms namely Random forest, Catboost, and LightGBM to compare the performance of the three algorithms on the bankruptcy classification problem and found the best results achieved at 98.21% coming from LightGBM [9]. Ben Jabeur made a comparison of FS-XGBoost with seven machine learning algorithms based on three well-known feature selection methods that are often used in bankruptcy prediction. The results of this study prove that the use of FS-XGBoost as a feature selection technique can provide more accurate predictions, thus outperforming conventional feature selection methods [10].

METHODS

This study applies the LightGBM algorithm as a classification model and is optimized using the XGBoost algorithm as a Feature Importance technique to increase accuracy in predicting corporate bankruptcy. The flow of research that will be carried out refers to the flowchart that has been made in Figure 1.

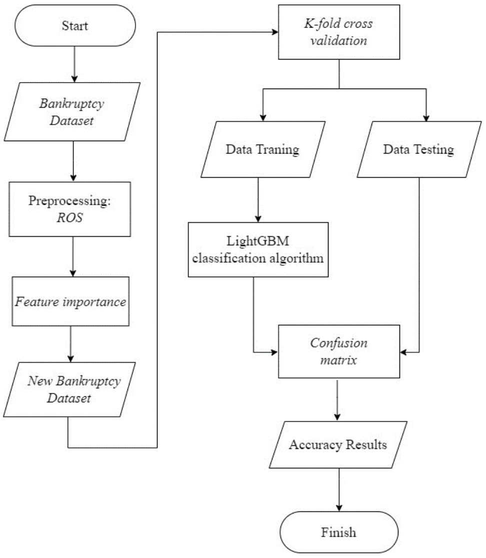

Figure 1. Research Flowchart

Data Preprocessing

Before applying data to a machine learning model, it is usually necessary to carry out the data preprocessing stage. Preprocessing is the initial stage in data processing which aims to clean, prepare, and organize data prior to analysis or modeling. Preprocessing is an important step that needs to be done. Preprocessing improves data quality because decisions will be predicted according to the data. For better decision-making, data anomalies can be detected and corrected, and reduced before analysis [11]. Preprocessing includes a series of techniques and procedures, such as eliminating missing data, normalizing data, removing duplicate data, and transforming data.

Feature Importance

The feature importance stage aims to determine which variables or features have the most influence on the target or class variables in the model. In machine learning, feature importance or importance ranking is one way to evaluate the features or variables used in the model. By determining feature importance, it can be determined which variables must be considered and the accuracy improved to improve model performance.

In selecting feature importance, the threshold is used to set the relevant importance value limit in determining which features are considered important or not. By setting the threshold, it is possible to select features that have an importance score above that limit. Features that do not reach the threshold will be considered unimportant and can be ignored in the next process.

K-Fold Cross Validation

K-Fold Cross Validation is a method that divides the data set randomly into two sub-sections, namely, training and test datasets [12]. At this stage the model validation is carried out using cross-validation with a number of iterations of 10, where the data is randomly divided into 10 parts then the testing process begins with the model built with the data in the first part, and the model formed will be tested on 9 parts of the remaining data.

LightGBM Algorithm Classification

At this stage, the classification is done by applying the LightGBM algorithm. The purpose of the classification is to make predictions about Bankruptcy. The classification algorithm is called using its library with the Python programming language. From the results of balancing the data and selecting the feature importance, then the classification stage is carried out. Classification with selected data will speed up the process and more optimal results. Through the process of cross-validation and data testing, algorithm function calls are carried out.

The LightGBM algorithm uses a boosting approach to produce an optimal level of accuracy in predictions [13]. Initially, LightGBM is initialized with predefined parameters, such as learning rate (learning_rate). This algorithm is a set of decision trees that aims to combine the predicted results from each tree into a final prediction. After the training process, LightGBM is evaluated using test data to measure the level of accuracy and performance of the model. If the evaluation results are still not satisfactory, the parameters in the model are reset to optimize performance.

Confusion Matrix

After the data modeling phase is carried out with the classification model, then the performance of the model will be analyzed for several metrics that determine how the model works in real-time implementation. In this stage, the researcher uses the Confusion Matrix to evaluate the model. The confusion matrix is built to evaluate, test, and check class performance in the model. The confusion matrix table can be seen in Table 1.

Table 1. *Confusion Matrix*

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

In the performance analysis that will be carried out, accuracy is chosen as a basic metric that can be calculated using the calculation formula written in Equation 1 [14].

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

Information:
TP (true positive) means that the actual value is positive, and the model prediction is also positive.
TN (true negative) means the true value is negative, and the model prediction is also negative.
FP (false positive) means that the actual value is negative, and the model prediction is positive.
FN (false negative) means that the actual value is positive, and the model prediction is negative.

RESULT AND DISCUSSION

This study conducted tests with a public dataset, namely Taiwanese Bankruptcy data. The dataset consists of 6,819 data with 96 variables, consisting of 95 independent variables and 1 dependent variable, namely Bankrupt status. The observation results in Figure 4 shows that the data is still imbalanced. Therefore, in this study, the data balancing process was carried out first in the preprocessing stage. Class data imbalance is known from the results of checking using the sns. catplot function. Visualization of the results in graphical form can be seen in Figure 3 which shows an imbalance in the target variable 'Bankrupt?' used. Manuscripts can be presented with the support of tables, graphs or images which are needed to clarify the results of the presentation verbally. Results and discussion are shown clearly and concisely.

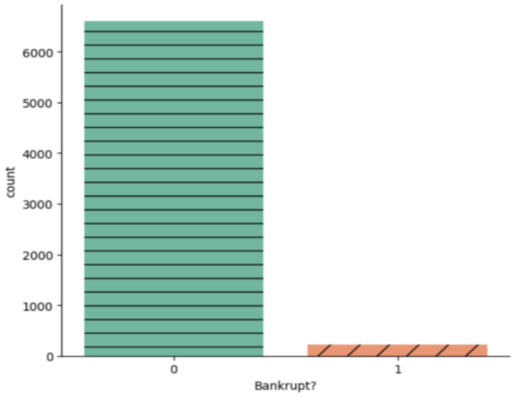

Figure 4. Original Taiwanese Dataset

After analyzing the data by checking for missing values, it was found that no data with a 'null' value was found. So the next step in preprocessing this data is handling the problem of imbalance in the data by using the random oversampling method. The results of handling unbalanced datasets using the random oversampling method can be shown in Figure 5.

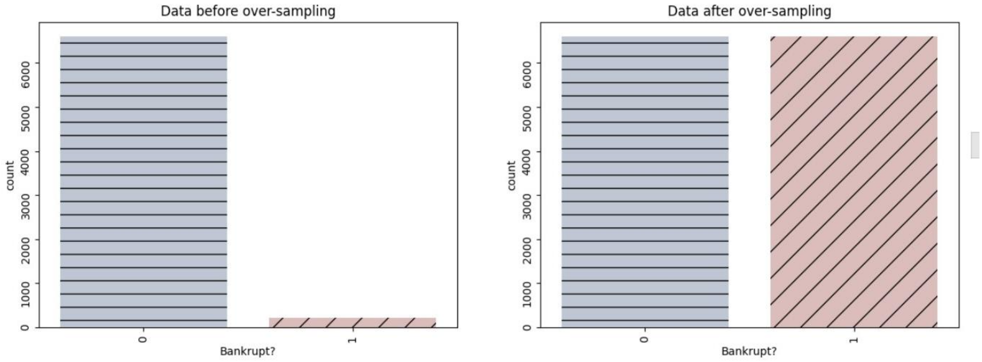

Figure 5. Random Oversampling Results

It can be seen from the results of the data before random oversampling was applied, in the majority class (0) there were 6,599 and in the minority class (1) there were 220, this shows that the comparison of data classes has an unequal percentage. Then after going through the process of random oversampling, the final data becomes comparable to a comparison of 6,599 data in the majority class and the minority class.

After balancing the data, the selection of feature importance is done using XGBoost. Feature importance is calculated explicitly for each attribute in the data set, which allows the attributes to be ranked and compared with each other. The feature selection process is carried out by taking the 13 best features resulting from implementing the feature_importances_ attribute in the XGBoost algorithm. feature_importances_ measures the importance of each feature in the prediction of the target variable. This attribute calculates Gain Importance, which is the increase in gain produced by each feature in the model. The feature that has the highest value has a strong correlation with the predicted results in the classification algorithm.

In the feature selection process, thresholds are used to set the relevant importance value limits to determine which features are considered important or not. Threshold settings can select only features that have an importance score above that limit. Features that do not reach the threshold will be considered unimportant and can be ignored in the next process. The threshold value is set as a percentage of the maximum value or the average feature importance score. Determination of the threshold value is based on research by [15]. which uses a threshold value of 0.01. The results of selecting feature importance are displayed based on ranking in Figure 6.

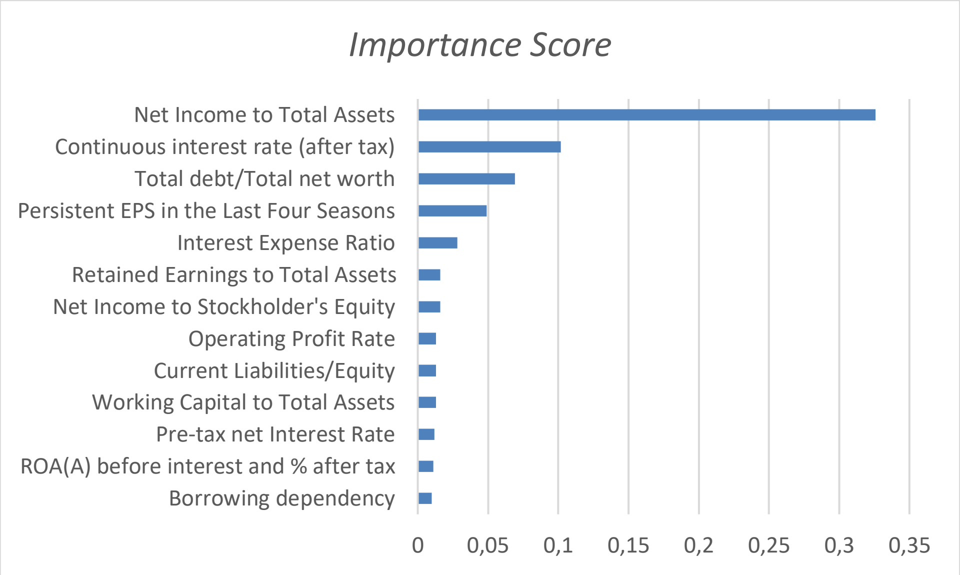

Figure 6. Results of the XGBoost Feature Importance Process

Based on the visualization results of the feature importance ranking shown in Figure 6, the feature with the highest importance score is the Net Income to Total Assets feature with a value of 0.32 and the lowest score is the Borrowing dependency feature with a value of 0.01.

At the cross-validation stage in this study using 10 folds. Implementation of cross-validation using the function cross_val_score. Furthermore, in the process, the dataset is divided into 2 parts, namely 90% training data and 10% test data. A more stable model performance estimate and better generalization can be obtained by dividing the data. The advantage of cross-validation is that it reduces the bias of using only one division of training data and a particular test data.

After dividing the data using cross-validation, the LGBMClassifier () function is called. Classification experiments were carried out per stage starting from classification without going through the data balancing process using ROS, classification after the data balancing process without implementing XGBoost Feature Importance, and classification after implementing ROS+XGBoost Feature Importance.

The first step is to calculate the accuracy of the LightGBM classification algorithm without applying Random Oversampling. The results after going through the calculations of 10 iterations using Cross Validation can be seen in Table 2.

Table 2. Results of 10 K-Fold Cross Validation Before ROS

Iteration	Results Accuracy
1	96,334%
2	97,067%
3	96,334%
4	94,281%
5	97,067%
6	96,334%
7	97,360%
8	96,067%
9	96,774%
10	97,209%
Average Accuracy	96,583%

Based on Table 2, the results of the average accuracy value obtained are 96.583%. Then the second experiment, carried out after going through the data balancing process using Random Oversampling can be seen in Table 3.

Table 3. K-Fold Cross Validation Results After ROS and Without XGBoost Feature Importance

Iteration	Results Accuracy
1	98,409%
2	97,878%
3	98,484%
4	97.575%
5	99.545%
6	99,469%
7	99,545%
8	99,242%
9	99,469%
10	99,545%
Average Accuracy	98,916%

Table 3 shows the results of the average accuracy value shown after applying Random Oversampling, proving an increase of 2.3%. Next, classify by applying XGBoost Feature Importance. From the results of the accuracy calculations carried out 10 times iterations based on the cross validation process, the predicted value is obtained with an accuracy of 99.227%. This proves an increase after the implementation of the feature selection technique. The results of 10 iterations of the LightGBM classification algorithm after implementing ROS+XGBoost Feature Importance are shown in Table 4.

Table 4. K-Fold Cross Validation Results After ROS+XGBoost Feature Importance

Iteration	Results Accuracy
1	98,560%
2	99,015%
3	99,015%
4	98,712%
5	99,696%
6	99,545%
7	99,545%
8	99,545%
9	99,924%
10	99,620%
Average Accuracy	99,227%

The model will be tested using test data that has previously been separated from the training data. The evaluation metric for model testing used in this study is accuracy. This stage is carried out after the classification process, which then performs model testing by applying calculations using cross-validation. The test results will be explained as each stage in the classification process is carried out.

Model testing is done using the confusion_matrix () function. In the first test, calculating the confusion matrix from the results of the classification without going through the data balancing process using ROS. The calculation results can be seen in Table 5.

Table 5. Confusion Matrix Without ROS+XGBoost Feature Importance

		Actual		
		Positive (1)	**Negative (0)**	**Amount**
Prediction	**Positive (1)**	6540	59	6599
	Negative (0)	174	46	220
	Amount	6615	105	6819

To get an objective measure of how good the model is at making predictions, it is necessary to do mathematical calculations. By measuring accuracy, we can obtain proportions or numerical values that indicate how accurate the model is in predicting class targets. Calculation of mathematical accuracy can measure as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100$$

$$\text{Accuracy} = \frac{6540 + 46}{6540 + 46 + 59 + 174} \times 100$$

$$= 96{,}583\%$$

In the second test, calculating the confusion matrix from the classification results after implementing ROS and without implementing XGBoost Feature Importance. The calculation results can be seen in Table 6.

Table 6. Confusion Matrix After ROS and Without XGBoost Feature Importance

		Actual		
		Positive (1)	**Negative (0)**	**Amount**
Prediction	**Positive (1)**	6456	143	6599
	Negative (0)	0	6599	6599
	Amount	6456	6742	13198

Based on Table 6, the calculation of mathematical accuracy can be described as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100$$

$$\text{Accuracy} = \frac{6456 + 6599}{6456 + 6599 + 143 + 0} \times 100$$

$$= 98{,}916\%$$

In the third test, calculating the confusion matrix from the classification results after implementing ROS + XGBoost Feature Importance. The calculation results can be seen in Table 7.

Table 7. Confusion Matrix After ROS+XGBoost Feature Importance

		Actual		
		Positive (1)	**Negative (0)**	**Amount**
Prediction	**Positive (1)**	6497	102	6599
	Negative (0)	0	6599	6599
	Amount	6497	6701	13198

The detailed explanation of the Confusion Matrix calculations is based on the classification results after applying ROS+XGBoost Feature Importance, namely the "Actual" column represents the actual class of the data, while the "Prediction" row represents the predicted class of the model. The values in the matrix are the amount of data included in the appropriate category. The reading of the results of the confusion matrix is based on the value information in Table 4.11, namely the value in the positive class results from positive predictions (1) which are also positive in actual conditions (1). If interpreted, predicted bankruptcy and in fact experienced bankruptcy. As can be seen in the table, there are 6,497 cases that are correctly predicted as a positive class (True Positive), and in fact, they are also included in the positive class, which means that the amount of data that is correctly predicted is "bankruptcy" and in fact, it is also "bankruptcy".

Furthermore, there are 6,599 cases that are correctly predicted as a negative class (True Negative), and in fact they are also included in the negative class, which means that the amount of data that is correctly predicted is not "bankruptcy" and in fact it is also not "bankruptcy". Then, there were 102 cases that were incorrectly predicted as a positive class (False Positive) and in fact were included in the negative class, which means that the amount of data that was predicted wrongly as "bankruptcy" but was not actually "bankruptcy". And finally, 0 cases were incorrectly predicted as a negative class (False Negative) which means that there was no wrong prediction of the positive class as "bankruptcy". In the context of bankruptcy prediction, this means that there were no cases predicted as "bankruptcy".

Based on Table 7, the calculation of mathematical accuracy can be described as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100$$

$$\text{Accuracy} = \frac{6497 + 6599}{6497 + 6599 + 102 + 0} \times 100$$

$$= 99{,}227\%$$

Accuracy provides information about how well the model predicts the target class. This can help understand whether the model has a low or high error rate. By calculating the accuracy, it is possible to compare the performance of several different models or algorithms. This can assist in the selection of the best model for a particular task.

CONCLUSION

In this study, the application of Light Gradient Boosting Machine (LightGBM) as a classification algorithm to produce a model that can predict bankruptcy. This experiment uses the Taiwan bankruptcy dataset. The dataset used has a match in the amount of target variable class data, so that balancing the amount of data is handled using the Random Oversampling technique. The results of the prediction accuracy obtained after balancing the dataset is 98.916%. Then, the process of selecting features of interest selected 13 out of 96 features based on a threshold value of 0.01. The results of the 13 selected features are then used in the classification process. After optimizing the performance of the LightGBM classification algorithm using XGBoost Feature Importance, a better prediction accuracy increases of 99.227% was obtained.

REFERENCES

[1] J. Kristiyono and A. Nurrosyidah, “Analisis Perilaku Pencarian Informasi Di Internet Melalui Fitur Visual Search,” *Scriptura*, vol. 11, no. 2, pp. 96–104, Dec. 2021, doi: 10.9744/scriptura.11.2.96-104.

[2] R. H. Hariri, E. M. Fredericks, and K. M. Bowers, “Uncertainty in big data analytics: survey, opportunities, and challenges,” *J. Big Data*, vol. 6, no. 1, p. 44, Dec. 2019, doi: 10.1186/s40537-019-0206-3.

[3] B. P. Bhattarai *et al.*, “Big data analytics in smart grids: state-of-the-art, challenges, opportunities, and future directions,” *IET Smart Grid*, vol. 2, no. 2, pp. 141–154, Jun. 2019, doi: 10.1049/iet-stg.2018.0261.

[4] Z.-H. Zho, “Machine learning,” in *Machine learning*, Springer Nature, 2021, p. 453.

[5] J. Yan *et al.*, “LightGBM: accelerated genomically designed crop breeding through ensemble learning,” *Genome Biol.*, vol. 22, no. 1, p. 271, Dec. 2021, doi: 10.1186/s13059-021-02492-y.

[6] A. Prabha, J. Yadav, A. Rani, and V. Singh, “Design of intelligent diabetes mellitus detection system using hybrid feature selection based XGBoost classifier,” *Comput. Biol. Med.*, vol. 136, p. 104664, Sep. 2021, doi: 10.1016/j.compbiomed.2021.104664.

[7] M. A. Muslim, Y. Dasril, A. Alamsyah, and T. Mustaqim, “Bank predictions for prospective long-term deposit investors using machine learning LightGBM and SMOTE,” *J. Phys. Conf. Ser.*, vol. 1918, no. 4, p. 042143, Jun. 2021, doi: 10.1088/1742-6596/1918/4/042143.

[8] S. Diantika, “Penerapan Teknik Random Oversampling untuk Mengatasi Imbalance Class Dalam Klasifikasi Website Phising,” *J. Mhs. Tek. Inform.*, vol. 7, no. 1, pp. 19–25, 2023, doi: https://doi.org/10.36040/jati.v7i1.6006.

[9] H. L. Quang Tien, L. Quang Tran, and T. Hop Do, “An Empirical Study on Bankruptcy Prediction using Ensemble Learning,” in *2022 RIVF International Conference on Computing and Communication Technologies (RIVF)*, Dec. 2022, pp. 173–178, doi:

10.1109/RIVF55975.2022.10013848.

[10] S. Ben Jabeur, N. Stef, and P. Carmona, “Bankruptcy Prediction using the XGBoost Algorithm and Variable Importance Feature Engineering,” *Comput. Econ.*, vol. 61, pp. 715–741, Jan. 2022, doi: 10.1007/s10614-021-10227-1.

[11] V. B. Gladshiya and K. Sharmila, “An Efficient Approach of Feature Selection and Metrics for Analyzing the Risk of the Students Using Machine Learning,” in *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, Oct. 2021, pp. 1–6, doi: 10.1109/ICAECA52838.2021.9675507.

[12] S. Saud, B. Jamil, Y. Upadhyay, and K. Irshad, “Performance improvement of empirical models for estimation of global solar radiation in India: A k-fold cross-validation approach,” *Sustain. Energy Technol. Assessments*, vol. 40, p. 100768, Aug. 2020, doi: 10.1016/j.seta.2020.100768.

[13] W. Liang, S. Luo, G. Zhao, and H. Wu, “Predicting Hard Rock Pillar Stability Using GBDT, XGBoost, and LightGBM Algorithms,” *Mathematics*, vol. 8, no. 5, p. 765, May 2020, doi: 10.3390/math8050765.

[14] B. Wang *et al.*, “Research on anomaly detection and real-time reliability evaluation with the log of cloud platform,” *Alexandria Eng. J.*, vol. 61, no. 9, pp. 7183–7193, Sep. 2022, doi: 10.1016/j.aej.2021.12.061.

[15] X. Shi, Y. D. Wong, M. Z. F. Li, C. Palanisamy, and C. Chai, “A feature learning approach based on XGBoost for driving assessment and risk prediction,” *Accid. Anal. Prev.*, vol. 129, no. March, pp. 170–179, 2019, doi: 10.1016/j.aap.2019.05.005.