Accredited Ranking SINTA 2
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 6 No. 6 (2022) 923 - 929 ISSN Media Electronic: 2580-0760

Detection of Credit Card Fraud with Machine Learning Methods and Resampling Techniques

Moh. Badris Sholeh Rahmatullah[1], Aulia Ligar Salma Hanani[2], Akmal M. Naim[3], Zamah Sari[4], Yufis Azhar[5]
[1,2,3,4,5]Informatics, Faculty of Engineering, University of Muhammadiyah Malang
[1]badrisrahmatullah@gmail.com, [2]aulialsh69@gmail.com, [3]Akmalmnaim12@gmail.com, [4]zamahsari@umm.ac.id,
[5] yufis@umm.ac.id

Abstract

Financial institutions in the form of banks provide facilities in the form of credit cards, but with the development of technology, fraud on credit card transactions is still common, so a system is needed that can detect fraud transactions quickly and accurately. Therefore, this study aims to classify fraudulent transactions. The proposed method is Ensemble Learning which will be tested using the Boosting type with 3 variations, namely XGBoost, Gradient Boosting, and AdaBoost. Then, to maximize the performance of the model, the dataset used is optimized with the Synthetic Minority Oversampling Technique (SMOTE) function from the Imblearn library in the data train to handle imbalanced dataset conditions. The dataset used in this study is entitled "Credit Card Fraud Detection" with a total of 284807 data which is divided into two classes: Not Fraud and Fraud. The proposed model received a recall of 92% with Gradient Boosting, where the results increased by 10.37% compared to the previous study using Random Forest with a recall result of 81.63%. This is because the use of SMOTE in the data train greatly influences the classification of Not fraud and fraud classes.

Keywords: machine learning, ensemble learning, classification, SMOTE, credit card fraud

1. Introduction

Banking is a system of economic scope that serves the public in the field of financial services. Most of the population in parts of the world use financial services in the form of banks, one of the banking efforts in determining the services needed by the community with the convenience of a facility called a credit card. However, it is very unfortunate that currently there are still many misuses of credit cards which are commonly referred to as carding or card fraud [1].

In the European Central Bank website publication, the total value of fraudulent transactions using credit cards published in the Single Euro Payments Area (SEPA) and obtained worldwide amounted to 1.87 billion euros in 2019 [3]. This is a fairly serious problem because credit card fraud losses are quite large. Therefore, to make transactions successful in the world of digital payments, credit card fraud detection must be considered [2].

To ensure fraud in credit card transactions, a classification process using machine learning technology is needed [2]. Machine learning aims to train machines to handle large amounts of data more efficiently, which is expected to be developed to make it easier to determine, search, and share data. It begins by dividing the data into two main parts, namely training data and test data [3]. Machine learning technology can be used to perform classification and detection with a fairly good success rate [4].

In machine learning, the type used is supervised learning or machine learning with labels that can later predict complete data patterns. In supervised learning, using any potential feature that increases the predictiveness of the model to meet certain requirements [5]. When there is new data in the extraction process, the features in the new data will be matched with the model pattern obtained from the data label [4]. Each label will be compared with the classification of data labels which will produce output in the form of classification results. In practice, one approach that is superior in machine learning is ensemble learning [4].

The three types of ensemble learning include Bagging, Stacking, and Boosting. To dig deeper into bagging means to combine many Decision Trees on different samples from the same dataset, with the result being a predictive average. Stacking involves many types of

Accepted: 02-07-2022 | Received in revised: 11-11-2022 | Published: 27-12-2022

models from the same data as well as using other models to learn the best combination of predictions. Boosting adds consecutive ensemble members, corrects the predictions made by the previous model and produces an average weight of the predictions [6].

Boosting is learning the sequence adaptively, namely the basic model that depends on the previous model and follows a deterministic strategy [4]. In this study, three boosts were used, including Gradient Boosting, AdaBoost, and XGBoost.

In this study, the dataset used is the same as the previous study [7], but previous studies used three different methods, namely Naive Bayes, LR, and Random Forest. The best method in this study is Random Forest with 99.96% accuracy, 96.38% precision, and 81.63% recall [7].

From the results of previous studies [7], the resulting accuracy is very good, but the recall value of 81.63% still needs to be improved again. Because the recall value is needed to detect fake transactions, where recall means counting the number of true fake transactions, which are caught in the model in positive labels. This is the main topic in the discussion of credit card fraud detection research.

Based on the description above, the purpose of this study is to increase the recall value from previous studies using three boosting scenarios and to try to determine the effectiveness with and without using SMOTE in the effect of the overall recall value. With the hope that the system created can provide benefits in accurately detecting fake credit cards and reducing the frequency of credit card fraud. Recall is considered more important for automating engineering task requirements than precision. Recall is the percentage of correct answers based on all possible correct answers [8].

2. Research Methods

2.1 Research Stages

The research method carried out has several stages as shown in Figure 1. The first stage is entering data which is then preprocessed by removing unused attributes. The next stage is data splitting, dividing the dataset into 80% train data and 20% test data. The third stage is divided into 2 scenarios, namely resampling using SMOTE so that both fraud and non-fraud classes are balanced. And without using SMOTE. Then proceed to the modeling section. The last stage is the evaluation of the results by paying attention to the confusion matrix and classification report.

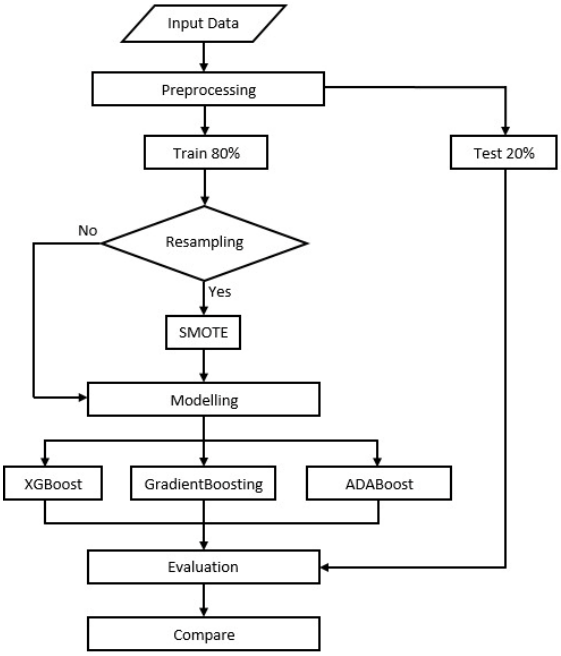

Figure 1. Research Flow

2.2 Dataset

The data used in this study is the same dataset from previous research [7] which is in the form of tabular data on European Cardholders' credit card transactions which can be accessed via Kaggle. The total amount of data is 284807 data. In this dataset, the amount of data between fraud and not-fraud classes is too unbalanced, where 0.173% of the total transactions are fake [7] with a total of 492 fake transactions and 244315 legal transactions. Then, in this study the data set will be divided into 80:20, where 80% is train data and 20% is test data. An example of the top five datasets can be seen in Table 1.

Table 1. Example dataset

	Time	V1	…	Amount	Class
0	0.0	-1,359,807	…	149.62	0
1	0.0	1,191,857	…	2.69	0
2	1.0	-1,358,354	…	378.66	0
3	1.0	-0.966272	…	123.50	0
4	2.0	-1,158,233	…	69.99	0

This dataset only contains numeric input variables that have the features "Time, V1, V2, V3, up to V28, Amount, and Class". Values V1 to V28 are using credit card transaction data at certain bank agencies, a statement to the dataset provider on the Kaggle website published by the machine learning group – ULB [9], that the provider does not provide original features and background information about the dataset. This is intended to maintain the confidentiality of credit card users. The original confidential dataset is then transformed using the PCA (Principal Component

DOI: https://doi.org/10.29207/resti.v6i6.4213

Analysis) method, which reduces the dimensions of the data without reducing the characteristics of the data [10]. Then leave the feature "time, amount, class" which is not transformed by PCA. Visualization of data imbalance between fake transaction data and genuine transactions can be seen in Figure 2.

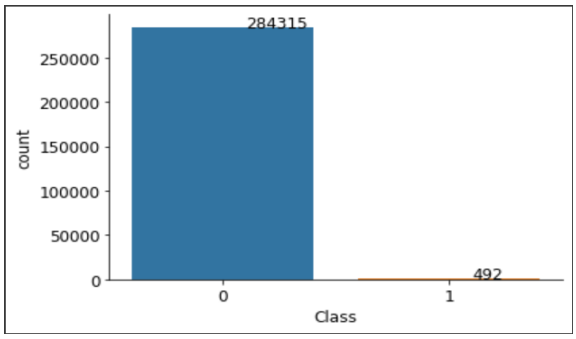

Figure 2. Graphics amount of data each class

2.3 Synthetic Minority Oversampling Technique (SMOTE)

The problem with the dataset used is that the dataset is unbalanced because there are too few minority classes. This problem is known as the imbalance data set problem [11]. One approach that is widely used in synthesizing new samples in the model is the Synthetic Minority Oversampling Technique (SMOTE). SMOTE is the development of the oversampling method. The way this works is by selecting the closest sample in the feature space, then drawing a line between the examples in the feature space and drawing a new sample along that line without changing the essence of the whole dataset [12].

This aims to improve the performance of the model used. Research related to the application of SMOTE to classification using Multi-Level Perceptron (MLP), k-Nearest Neighbor (K-NN), and other methods has succeeded in improving classification performance with the best results using the Naive Bayes method which has been SMOTE at 90.7% [13] .

Instantiation class minority x_i selected as the basis for creating new synthetic data points. Based on the distance metric, several nearest neighbors of the same class (points x_i up to x_{i4}) are selected from the *training set.* Finally, interpolate random conducted for get example new r_1 until r_4 [14]. Image for illustration SMOTE algorithm can seen in Figure 3.

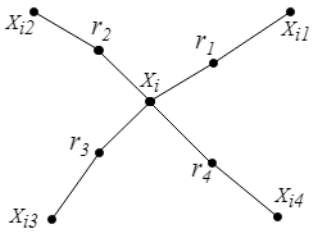

Figure 3. Illustration SMOTE algorithm

Taken from reference sources [14]. First, the minority class instances are randomly selected from the training set. Next, K nearest neighbors (5 by default) are obtained. Finally, this N of K instances are randomly selected to enumerate new instances.

2.4 Architecture Algorithm

XGBoost is an ensemble classification that uses gradient enhancement, whose model structure leads to a loss function which is further expanded by adding an expansion function [15]. By reviewing the dataset used in XGBoost m is symbolized for the features in the dataset, and N is the number of data in the dataset used with equation 1.

$$E = \{i = 1 \dots n\}, if\{a_i \in R^m \; b_i \in R \quad (1)$$

On variable $n$is the sample train data, and m the features of each sample.b_i show circumstances payload in sample i. *Tree boosting* made one group in equation 2.

$$F = \{f_k(a) = w_q(a)\}, if k \in R^m \quad (2)$$

Following is the formula for predict score from the tree boosting model with outputs b_i and f as function equation . With followed score Kon trees in equation 3 .

$$\hat{b}_i = \sum_{K=1}^{k} (f_k(a_i)) \; | \{f_k \in F\} \quad (3)$$

which is in equation 4.

$$\Omega(D) = \alpha N + \frac{\beta \|\omega\|^2}{2} \quad (4)$$

On the equation, Ω defines the complexity of the model. $\propto$as the controlling parameter and the βleaf number of N. $W$is the magnitude of the leaf weight. During each round of model training data f_k, the XGBoost algorithm adds a new function to the model, keeping the final prediction result unchanged [15].

Gradient Boosting is method that can used for develop classification and regression models for optimize the model learning process, which partly big characteristic non-linear and more known as tree decision or regression [16].

Given a set of training $\{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$, where x_1 is the sample feature to i and $y_i \in \{0,1\}$ shows the label of the th sample i. The machine learning algorithm realizes credit scoring by designing a function $F(x_i)$to minimize the loss function $L(y_i, F(x_i))$:

$$F^* = \underset{F}{argmin} \sum_{i=1}^{N} L(y_i, F(x_i)). \quad (5)$$

Algorithm gradient boost realize equation 5 with method integration additives:

$$F(x) = \sum_{i=1}^{T} f_t(x). \quad (6)$$

where T is the number of iterations. Based on equation 6 that $F(x_i)$ is integrated gradually in an additive

DOI: https://doi.org/10.29207/resti.v6i6.4213

manner. In iteration to - t, f_t realizing further optimization of the overall disadvantage of the preconceived ensemble $\{f_j\}_{j=1}^{t-1}$. In implementing the Gradient Boosting Decision Tree, each function f implemented by a Decision Tree that can considered as base learner. because of that, f can be realized as $f(\alpha; x)$, α is the structural parameter of each decision tree that determines features and splitting threshold on each internal splitting node in the decision tree.

Because iteration to – t realize further optimization of the loss function. Function loss declared with equation 7.

$$L\big(y_i, F_{t-1}(x_i) + f_t(x_i)\big) \approx L(y_i, F_{t-1}(x_i)) + g_i f_t(x_i) + \frac{1}{2} f_t(x_i)^2 \quad (7)$$

where g_i is the first derivative of the loss function which can be calculated as:

$$g_i = \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x_i)=F_{t-1}(x_i)} \quad (8)$$

Therefore, equation 8 can be turned into an optimization problem:

$$f_t^* = argmin\ f_t\ \sum_{i=1}^{N} \frac{1}{2}(f_t(x_i) - g_i)^2 \quad (9)$$

It can be seen from equation 9 that the target fitting of f_t is the negative gradient of the loss function. Therefore, before training each tree in the Gradient Boosting Decision Tree, the target training is updated in each tree in equation 10 [17].

$$\{y_i\}_{i=1}^{N} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x_i)=F_{t-1}(x_i)} \quad (10)$$

Adaboost is defined as an iterative algorithm that uses different classifiers for the same training set, then combines them to create the strongest classifier at the end [18]. The Adaboost algorithm is specially made to handle classification problems which serve to improve the accuracy of weak learners [19]. Adaboost has several advantages, which are simple and easy to implement [20].

AdaBoost Algorithm explained as following. Assume that X is sample space, Y represents gathering category identified samples. Arranged with $Y = \{-1, +1\}$ so that is the $S = \{(xj, yj) \mid j = 1,2, \ldots, m\}$ train sample set, where xj is X, and yj is Y.

Then next initialize weight n The sample so that Dt it is evenly distributed represents the sample weight (xj, yj) specified in the iteration t with the equation 11.

$$D_j(j) = \frac{1}{n} \quad (11)$$

For example T is the number of iterations for each $t = 1 \ldots T$, based on the sample distribution Dt, resulting in a sample to form a set St. practice classifying ht on sets St. Use classifiers ht to classify all sample set S. Classification is carried out in this round and the minimum error rate is E_j in equation 12.

$$E_j = \sum_{j=1}^{n} |h_1(x_j) - y_j| \quad (12)$$

count weight with equation 13.

$$\beta_j = E_j/(1 - E_j) \quad (13)$$

then for look for score a calculated in equation 14.

$$a = log(1/\beta_j) \quad (14)$$

then update weight with equation 15.

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right) \quad (15)$$

2.5 Scenario Test

Referring to research [7]. This study uses the same feature selection as previous studies. Where is the dropped class "time, amount, class, v28".

In this study, classification was carried out into two classes, namely fraudulent transactions, and non-fraud transactions. Then the dataset is split with a ratio of 80% for train data and 20% for test data. Furthermore, the data train is balanced using SMOTE. Meanwhile, the test data is left as the actual data. The amount summarized in Table 2 is the total dataset.

Table 2. Total Data Before Conducted Oversampling

Class	Training Data	Testing Data	Total Data
Not Fraud	227453	56862	284315
Fraud	392	100	492

Table 3 is the number of SMOTE train data and test data.

Table 3. Total Data After Done SMOTE

	Training Images	Testing Images	Total Images
Not Fraud	227453	56862	284315
Fraud	227453	100	227553

Furthermore, each algorithm is tested in 2 scenarios with 3 different algorithm tests. Namely, XGBoost, Gradient Boosting, and AdaBoost are described in Table 4.

Table 4. Scenario Test

Algorithm	Scenario 1	Scenario 2
XGBoost	No SMOTE	With SMOTE
GradientBoosting	No SMOTE	With SMOTE
Adaboost	No SMOTE	With SMOTE

3. Results and Discussion

The results of this study were carried out based on the methodological arrangement described in the research

DOI: https://doi.org/10.29207/resti.v6i6.4213

method. The first step in this research is the selection of a dataset entitled Credit Fraud Detection, which consists of two classes, 492 fraud data and 284,315 data for Not Fraud.

Next, the dataset is downloaded and saved into Google Drive using an API integrated with Kaggle via Google Colab. Google Colab was chosen because it has collaboration features so researchers can easily share projects. Before processing, the dataset is split into a ratio of 80% train data and 20% test data which is continued using the SMOTE imblearn library [21], to balance the amount of data in each class.

The Fraud class has less data than the Not Fraud class so that the amount of data in the Fraud class will be added as much as the difference between the number of data between Fraud and Not Fraud classes in the training data, which is 283,823 data. data visualization after SMOTE is performed in Figure 4.

Figure 4. Number of classes that have been balanced

3. 1 Analysis and Test Results

Each pre-defined algorithm, training dataset that has been preprocessed is carried out by the default iterative fit process in the sklearn library [22]. The next stage is analysis and test results, analysis can be done in several ways, one of which is by using a confusion matrix.

The confusion matrix is a tabular layout that compares the predicted class labels with the actual class labels in all data instances [23]. Based on Figure 11, the confusion matrix table for the XGBoost scenario using SMOTE can be understood that in the Fraud class there are 91 image data that are predicted correctly and 9 data that are predicted incorrectly. As well as in the Not Fraud class there are 56264 correctly predicted data and 598 incorrectly predicted data by the model. Then continued Figure 13.

In Figure 12, the confusion matrix table for the Gradient Boosting scenario using SMOTE can be seen in the Fraud class, there are 92 image data that are correctly predicted and 8 image data that are predicted incorrectly. As well as in the Not Fraud class there are 56264 correctly predicted data and 673 incorrectly predicted data by the model.

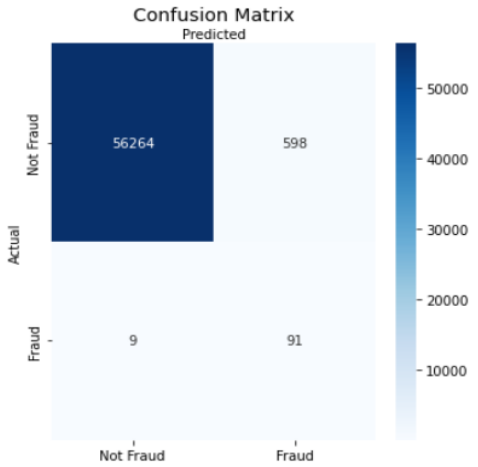

Figure 11. *Confusion matrix* XGBoost using SMOTE

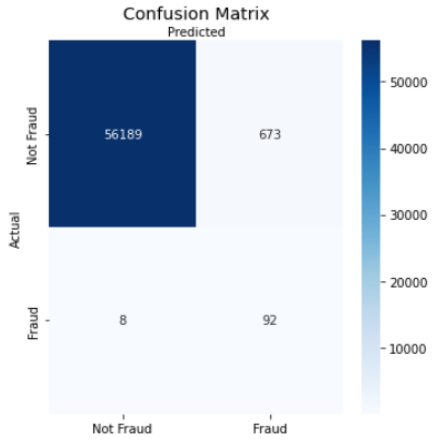

Figure 12. Confusion matrix Gradient Boosting using SMOTE

Finally, for Figure 13, the confusion matrix table for the AdaBoost scenario using SMOTE in the image in the Fraud class, there are 91 image data that are correctly predicted and 9 image data that are predicted incorrectly. As well as in the Not Fraud class there are 55480 data that are correctly predicted and 1383 data that are incorrectly predicted by the model.

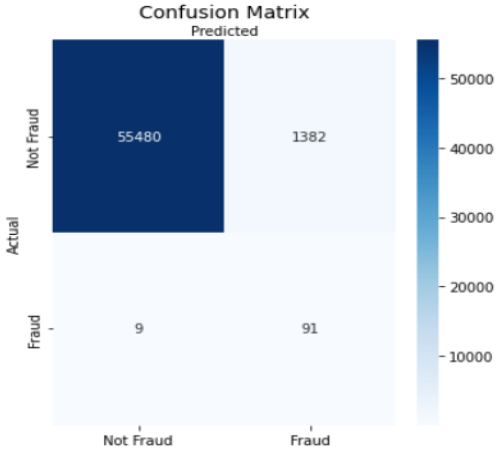

Figure 13. Confusion matrix AdaBoost using SMOTE

DOI: https://doi.org/10.29207/resti.v6i6.4213

The three models are able to detect Fraud class well. The highest model of the three models is the Gradient Boosting model.

Each trial has a different recall value. Based on Table 5, the recall value in the three scenarios with SMOTE has a value above 90% except for Adaboost with SMOTE in the fraud class producing a value of 89%.

In the third scenario without SMOTE, the results obtained in the three algorithms are seen from the recall value for the not fraud class obtained a value of 100% but for the Fraud class it has not produced a value above the value of 90%.

Table 5. Test Results of Each Scenario

Scenario	Class	recall
XGBoost with SMOTE	Not Fraud	99%
	Fraud	91%
XGBoost without SMOTE	Not Fraud	100%
	Fraud	75%
Gradient Boosting with SMOTE	Not Fraud	99%
	Fraud	**92%**
Gradient Boosting without SMOTE	Not Fraud	100%
	Fraud	68%
AdaBoost with SMOTE	Not Fraud	98%
	Fraud	89%
AdaBoost without SMOTE	Not Fraud	100%
	Fraud	76%

3.3. Comparison of the Best Model Performance with Previous Research

After a series of test scenarios have been carried out, the next process is to compare the performance of the best model with the results obtained in previous studies. Based on Table 6, the classification report in the XGBoost scenario obtained 99% accuracy and 100% precision in the Not Fraud class and 93% in the Fraud class, where the data increased overall after using without SMOTE.

Table 6. The proposed classification report model

	precision	recall	f1-score	support
Not Fraud	1.00	0.99	0.99	56862
Fraud	0.12	0.92	0.21	100
accuracy			0.99	56962
macros avg	0.56	0.95	0.60	56962
weighted avg	1.00	0.95	0.99	56962

According to Table 7, this study produced a Gradient Boosting model using SMOTE as the best model, where this model can exceed the recall results of the model built in previous research by 10.37%.

Table 7. Recall Results of Each Scenario

Model	recall
Random Forest	81.63%
The best proposed model	**92%**

4. Conclusion

Based on the research that has been done. SMOTE greatly affects the overall recall results. In previous research, Recall from the Random Forest model assisted by SMOTE obtained a yield of 81.63%. Therefore, this research succeeded in increasing the recall value using the best model, namely Gradient Boosting with SMOTE by 92%. The other 2 models, namely XGBoost and AdaBoost, produced a recall value of 91% for both, only 1% difference from the best model previously described.

Then, to continue this research, it is recommended to use deep learning. This is intended to find out whether the same dataset when using deep learning can produce better accuracy, precision, and recall values.

Reference

[1] D. Tanouz, RR Subramanian, D. Eswar, GVP Reddy, AR Kumar, and CHVNM Praneeth, "Credit card fraud detection using machine learning," *Proc. - 5th Int. conf. Intell. Comput. Control System. ICICCS 2021*, pp. 967–972, 2021, doi: 10.1109/ICICCS51141.2021.9432308.

[2] A. Kurniawan and Y. Yulianingsih, "Estimating Fraud Detection on credit cards with Machine Learning," *Kila*, vol. 10, no. 2, pp. 320–325, 2021, doi: 10.33322/kilat.v10i2.1482.

[3] M. Algorithm, B. Ayunda, B. Classifier, T. Na, and B. Classifier, "Classification Of Visitor Satisfaction at The Museum Using The Naïve Bayes Algorithm," pp. 89–100, 2021.

[4] J. Brownlee, "A Gentle Introduction to Ensemble Learning Algorithms," *https://machinelearningmastery.com/*, 2021. https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/ (accessed Aug. 17, 2022).

[5] P. Grabowicz, N. Perello, and A. Mishra, *Marrying Fairness and Explainability in Supervised Learning*, vol. 1, no. 1. Association for Computing Machinery, 2022.

[6] S. Hussein, "Ensemble learning in Machine Learning: Bagging and Boosting," *https://geospasialis.com/*, 2021. https://geospasialis.com/ensemble-learning/ (accessed Jun. 27, 2022).

[7] D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla, "2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH): proceedings: March 20-21, 2019, Jahorina, East Sarajevo, Republic of Srpska, Bosnia and Herzegovina," *2019 18th Int. sym. INFOTEH-JAHORINA*, no. March, pp. 1–5, 2019.

[8] JP Winkler, J. Grönberg, and A. Vogelsang, "Optimizing for recall in automatic requirements classification: An empirical study," *Proc. IEEE Int. conf Require. eng*, vol. 2019-September, pp. 40–50, 2019, doi:10.109/RE.2019.00016.

[9] MLG-ULB, "Credit Card Fraud Detection | Kaggle," 2017. https://www.kaggle.com/mlg-ulb/creditcardfraud/data%0Ahttps://www.kaggle.com/mlg-ulb/creditcardfraud (accessed Aug. 24, 2022).

[10] R. Firliana, R. Wulanningrum, and W. Sasongko, "Implementation of Principal Component Analysis (PCA) for Human Face Recognition," *J. Eng.* , vol. 2, no. 1, pp. 65–69, 2015.

[11] YS Aurelio, GM de Almeida, CL de Castro, and AP Braga, "Learning from Imbalanced Data Sets with Weighted Cross-Entropy Function," *Neural Process. Lett*, vol. 50, no. 2, pp. 1937–1949, 2019, doi:10.1007/s11063-018-09977-1.

[12] J. Brownlee, "SMOTE for Imbalanced Classification with Python," *Machinelearningmastery.Com*, 2020. https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/ (accessed Jun. 29, 2022).

DOI: https://doi.org/10.29207/resti.v6i6.4213

[13] B. Karlik, A. Mohammed, Y. Bahir, and B. Koçer, "Comprising Feature Selection and Classifier Methods with SMOTE for Prediction of Male Infertility," *Artic. int. J. Fuzzy Syst*, no. November 2019, 2016, [Online]. Available: https://www.researchgate.net/publication/337307643.

[14] A. Fernández, S. García, F. Herrera, and NV Chawla, "SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary," *J. Artif. Intell. res*, vol. 61, pp. 863–905, 2018, doi:10.1613/jair.1.11192.

[15] S. Jafari, Z. Shahbazi, YC Byun, and SJ Lee, "Lithium-Ion Battery Estimation in Online Framework Using Extreme Gradient Boosting Machine Learning Approach," *Mathematics*, vol. 10, no. 6, 2022, doi:10.3390/math10060888.

[16] N. Chakrabarty, T. Kundu, S. Dandapat, A. Sarkar, and DK Kole, *Flight arrival delay prediction using gradient boosting classifier*, vol. 813. Springer Singapore, 2019.

[17] Y. Zou and C. Gao, "Extreme Learning Machine Enhanced Gradient Boosting for Credit Scoring," *Algorithms*, vol. 15, no. 5, 2022, doi: 10.3390/a15050149.

[18] F. Wang, D. Jiang, H. Wen, and H. Song, "Adaboost-based security level classification of mobile intelligent terminals," *J. Supercomput*, vol. 75, no. 11, p. 7460–7478, 2019, doi: 10.1007/s11227-019-02954-y.

[19] A. Andreyestha and A. Subekti, "Sentiment Analysis on Film Reviews Using Ensemble Learning Optimization," *J. Inform*, vol. 7, no. 1, pp. 15–23, 2020, doi: 10.31311/ji.v7i1.6171.

[20] A. Shahraki, M. Abbasi, and Ø. Haugen, "Boosting algorithms for network intrusion detection: A comparative evaluation of Real AdaBoost, Gentle AdaBoost and Modest AdaBoost," *Eng. appl. Artif. Intell*, vol. 94, no. February, p. 103770, 2020, doi: 10.1016/j.engappai.2020.103770.

[21] S. Kiyohara, T. Miyata, and T. Mizoguchi, "Prediction of grain boundary structure and energy by machine learning," vol. 18, pp. 1–5, 2015, [Online]. Available: http://arxiv.org/abs/1512.03502.

[22] DK Barupal and O. Fiehn, "Generating the blood exposome database using a comprehensive text mining and database fusion approach," *Environ. Health Perspective*, vol. 127, no. 9, pp. 2825–2830, 2019, doi:10.1289/EHP4713.

[23] J. Görtler *et al*, *Neo: Generalizing Confusion Matrix Visualization to Hierarchical and Multi-Output Labels*, vol. 1, no. 1. Association for Computing Machinery, 2022.

DOI: https://doi.org/10.29207/resti.v6i6.4213