Increasing Accuracy of The Random Forest Algorithm Using PCA and Resampling Techniques with Data Augmentation For Fraud Detection of Credit Card Transaction

Andhika Seno Tamtama [1] , Riza Arifudin [1*]

[1] Department of Computer Science, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang, Semarang, Indonesia
*Corresponding author: rizaarifudin@mail.unnes.ac.id

ARTICLE INFO

Article history
Received: 10 Maret 2022
Revised: 27 Maret 2022
Accepted: 17 April 2022

Keywords
Imbalanced Data
Random Forest
PCA
Resampling Technique
Data Augmentation

ABSTRACT

The credit-card transaction analysis uses a random forest algorithm as an algorithm for the classification process. The problem faced from the classification process using credit card fraud filing dataset fraud is an imbalanced data that causes an imbalanced data alignment on the model results from data training. To resolve the problem, a combination of PCA methods and resampling techniques with data augmentation for the optimum process on random forest classification algorithms. The PCA method is used in the preprocessing stage to do the process of transforming data into numerical data and resampling techniques and data augmentation are used in data resamples to bring the data to a balance. The data used is a data card fraud of Europe that has 284807 transactions. Model accuracy measurement was implemented using a confusion matrix. The highest accuracy results from a random forest combination using PCA and resampling techniques with data augmentation of 99.9976%.

1 Introduction

The development of the amount of data in large databases cannot be separated from the rapid development of technology. Currently the conditions in the world are rich in data, but not rich information. The wealth of data must be balanced with the right processing method so that the existing data can be processed using data mining techniques.

Data mining is said to be a method that is done by pulling existing data from a large database. According to Finogeev et al., (2017) data mining is known as Knowledge Discovery in Database (KDD). Data mining is the mining of new information using a method of finding certain patterns or rules from some large amounts of data (Jeihouni et al., 2020). There are several techniques in data mining that can be used, one of which is classification. Classification means a training or learning activity carried out on the target function by mapping each set of attributes (features) into one number of available class labels (Utomo & Mesran, 2020). Classification can be defined as a basic form of data analysis (Lee & Yoon, 2017). In the scientific field, classification is usually applied to detect a problem that can be solved by machine learning. In this study, we will discuss the detection of credit card fraud.

Credit card fraud is the act of someone using a credit card for personal reasons without the credit card owner's consent and without the intention to repay the purchases made (Carneiro et al., 2017). Credit card fraud events often occur and result in large financial losses where fraudsters can

use several technologies such as trojans or phishing to steal information from other people's credit cards (Xuan et al., 2018). In this case of credit card fraud, detection measures must continue because the occurrence of credit card fraud is unpredictable (Sisodia et al., 2017). Credit card fraud is a critical problem and has important accountability for the corporate sector, organizations, and state governments (Devi et al., 2019). Handling of credit card fraud cases in Indonesia is less effective due to laws and regulations that do not regulate in detail the perpetrators who have violated them and the implementation of the handling carried out by law enforcers has not been optimal (Alhakim & Sofia, 2021). According to the Indonesian Credit Card Association (AKKI), in 2021 the amount of credit card user transactions will be 277,051,232 so that number shows there are still many Indonesians who use credit cards (AKKI, 2022).

One of the methods that can be used to indicate fraudulent credit card transactions is the random forest algorithm. Random Forest Algorithm (RFA) is used for classification, regression, and other tasks performed by constructing multiple decision trees (Kumar et al., 2019). The random forest algorithm is included in the best performing supervised learning model (Klapwijk et al., 2019). Random forest is a combination algorithm that has the advantages of automatic balancing of an error, has an automatic selection feature, and is easy to parallelize so that it has good performance in classifying large-scale unbalanced data (Lin et al., 2017).

In addition to using the random forest algorithm, the feature extraction method used in this study is Principal Component Analysis (PCA). This PCA is used for feature selection in the credit card fraud detection dataset. PCA plays a role in the preprocessing process, namely the transformation of datasets into numeric numbers. The resampling method is one method that can be used in dealing with data imbalances (Andrian et al., 2018). There are three ways that can be used in this resampling method, namely undersampling, oversampling, or hybrid (a combination of undersampling and oversampling) (Siringoringo, 2017). In addition to using resampling, this study also uses data augmentation, namely Synthetic Minority Oversampling Technique (SMOTE). SMOTE is a popular augmentation method used to solve problems with class imbalance (Shorten & Khoshgoftaar, 2019). This SMOTE data augmentation method tries to correct the class imbalance by generating additional patterns in the minority class (Iwana & Uchida, 2021).

Research conducted by Xuan et al., (2018) applied a random forest algorithm with Classification and Regression Trees (CART) to identify fraud detection using a credit card fraud dataset from a Chinese e-commerce company. The accuracy results obtained using the random forest algorithm with CART is 96.77% with 62 attributes. Kumar et al., (2019) conducted research on the random forest algorithm with a neural network for fraud transaction detection. Detection carried out using a credit card fraud dataset from Kaggle obtained an accuracy of 90% with 120,000 training data and 60,000 test data.

Hordri et al. (2018) investigated the use of a combined random forest algorithm with resampling techniques to analyze fraudulent transactions and non-credit card fraud as well as checking for increased accuracy against the use of different resampling techniques. This study uses a credit card fraud dataset from Kaggle and gets the highest accuracy of 99.9951% on the random oversampling technique with a total of 284807 transactions. So, this study focuses on increasing the accuracy of fraud detection credit card transactions using the random forest classification algorithm with resampling (undersampling and oversampling) and data augmentation (SMOTE) techniques.

2 Method

In this study, a combination of random forest algorithm, PCA, resampling technique, and data augmentation. PCA is used for feature selection in the credit card fraud detection dataset and plays a role in the preprocessing process, transformation of datasets into numeric numbers. Resampling techniques is one method that can be used in dealing with data imbalances. Data Augmentation used is SMOTE. SMOTE is a popular augmentation method used to solve problems with class imbalance. The Random Forest Algorithm is used to classify datasets which indicates whether the transaction is a fraudulent transaction or a normal transaction. The results of classification accuracy are calculated using a confusion matrix. This study uses material in the form of a dataset "Credit Card Fraud Detection" from Kaggle (Lebichot et al., 2020; Lebichot et al., 2021). The dataset used contains transactions made with credit cards in September 2013 by European cardholders. This

dataset presents transactions that occurred in two days, where in the dataset had 492 frauds of 284,807 transactions. The data set is very unbalanced i.e. positive class fraud accounts for 0.172% of all transactions. The flowchart of the method used in this study is shown in Figure 1.

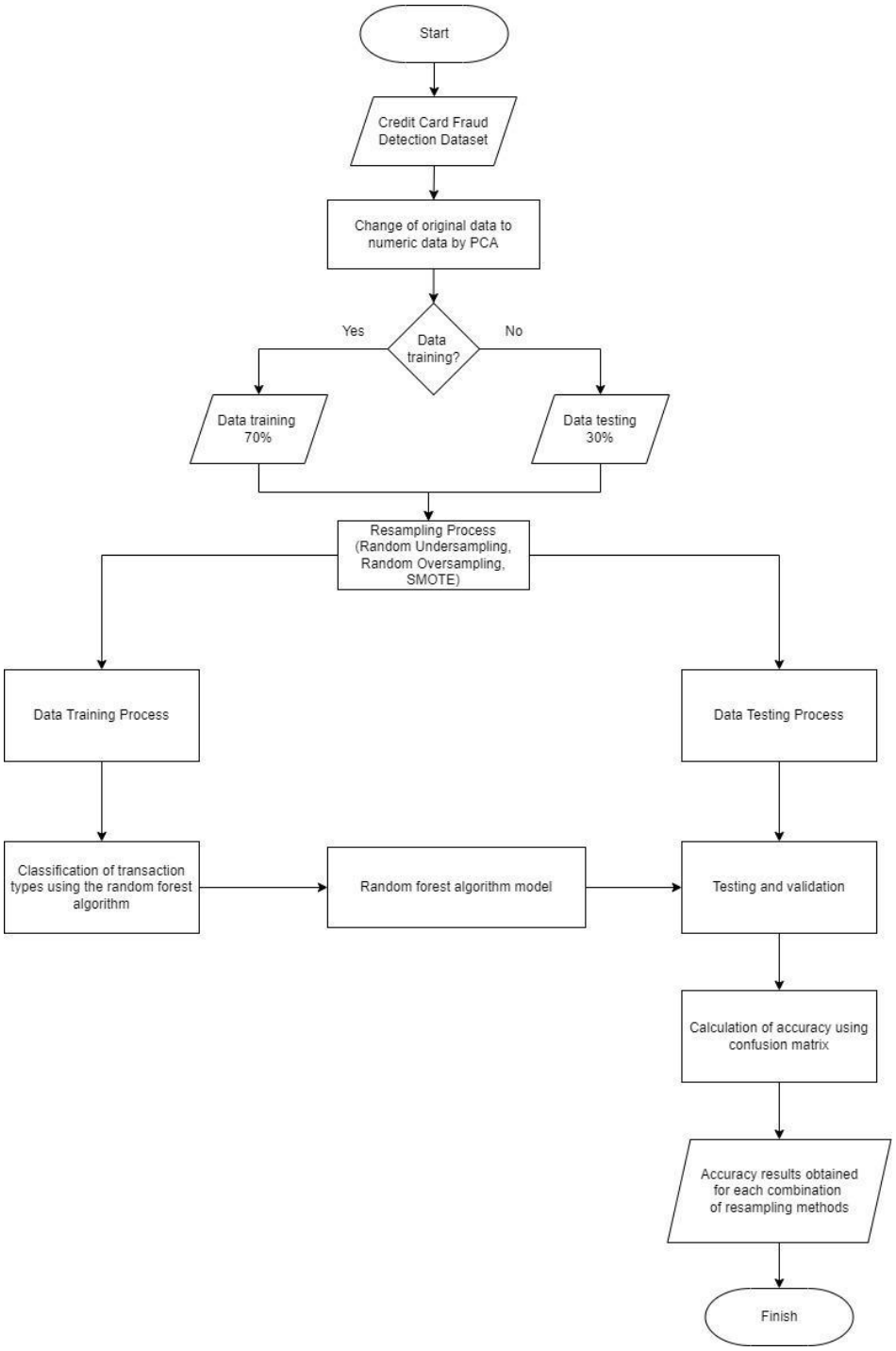

Figure 1. Proposed method flowchart

2.1 Data Preprocessing

2.1.1 *Data Transformation*

Data transformation or data transformation that involves several activities such as normalization, smoothing, aggregation, attribute construction, and generalization data (Yee et al., 2018). Data transformation serves to change the data within a certain range to normalize the data (Devi & Kavitha, 2017). With the data transformation, it will produce more accurate grouping of

data. Data transformation is performed by PCA. PCA is a statistical technique that allows resizing of datasets, taking into account uncorrelated and redundant information. The PCA technique found the most significant variation of the variables. The transformation process carried out is to change the feature data V1, V2, … V28 which is the input variable is changed to numeric value. This is because of confidentiality of user data information and features that are sensitive to public knowledge. And using PCA can also get 30 features as input data in processing calculating the accuracy of this study. Explanation of 30 features as input data can be seen in table 1.

Table 1. Features as input data

Feature Name	Description
Time	The 'Time' feature contains the seconds that elapsed between each transaction and the first transaction in the data set.
V1	The 'V1' feature shows the credit card number
V2	The 'V2' feature shows the gender of the user
V3	The 'V3' feature shows the user's marital status
V4	The 'V4' feature shows credit limit
V5	The 'V5' feature shows the previous month's bill
V6	The 'V6' feature shows the previous month's payment
V7	The 'V7' feature shows the status of the credit account account
V8	The 'V8' feature shows the user's salary value
V9	The 'V9' feature shows credit history
V10	The 'V10' feature shows other available credits
V11	The 'V11' feature shows the purpose of doing credit
V12	The 'V12' feature shows the number of credits made
V13	The 'V13' feature shows the user's current job status
V14	The 'V14' feature shows a savings account
V15	The 'V15' feature shows user status
V16	The 'V16' feature shows other debtors
V17	The 'V17' feature shows the properties owned by the user
V18	The 'V18' feature shows the user's age
V19	The 'V19' feature shows the user's home status
V20	The 'V20' feature shows the number of credit cards owned
V21	The 'V21' feature shows the user's work
V22	The 'V22' feature shows the user's phone number

Feature Name	Description
V23	The 'V23' feature shows the overseas jobs held
V24	The 'V24' feature shows user ID
V25	The 'V25' feature shows credit card PIN
V26	The 'V26' feature shows an error in the transaction
V27	The 'V27' feature shows a reduction in credit score
V28	The 'V28' feature shows the year the credit account was opened
Amount	The 'Amount' feature shows the total transaction value

Because the dataset has many dimensions (features), researchers will reduce the number of dataset feature variables using PCA which initially contained 31 features in the dataset after implementation. In this PCA, the features used as input data are 30 features.

2.1.2 *Data Resampling*

The resampling data includes undersampling and techniques of oversampling. This resampling process is a process carried out to distribute training data from different classes in the data space. Resampling technique is a preprocessing technique that equalizes the distribution of data classes algorithmically to increase the imbalance ratio and reduce the effects of unbalanced class distributions in the machine learning process. Several studies have shown that the resampling method can improve the model's capabilities to some extent by re-sampling the data to adjust the sample distribution. To achieve better results, unbalanced data is processed first with resampling using undersampling and oversampling in order to get better results later. In this study, the undersampling process can be seen in figure 2.

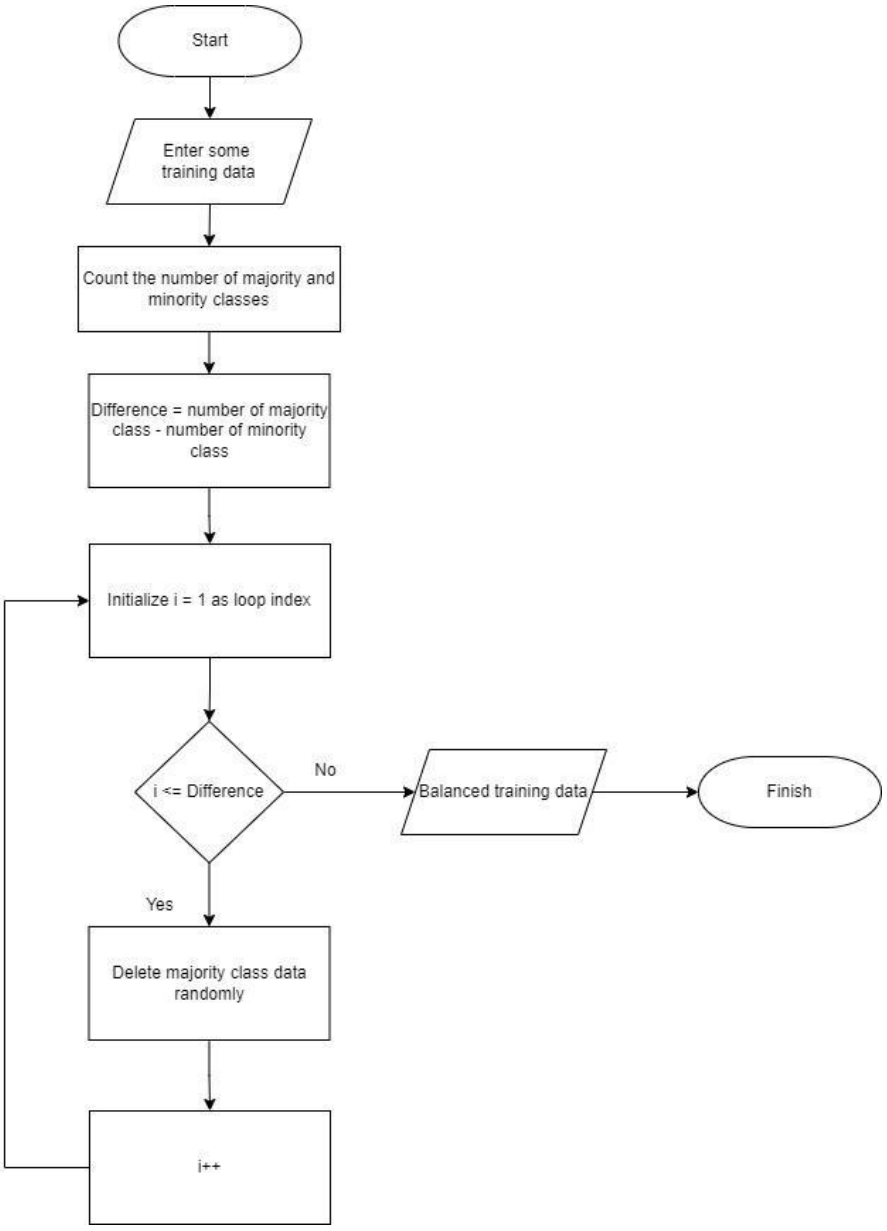

Figure 2. Undersampling techniques process

This undersampling process is a process that works on the majority class and has advantages over large datasets. The undersampling process will reduce the number of observations from the majority class to make the dataset balanced. In this study, researchers used random undersampling. In random undersampling, data from the majority class will be selected randomly until the dataset is balanced. Then the oversampling process aims to increase the minority class sample by duplicating it randomly. Figure 3 shows the process of occurrence oversampling to deal with unbalanced datasets.

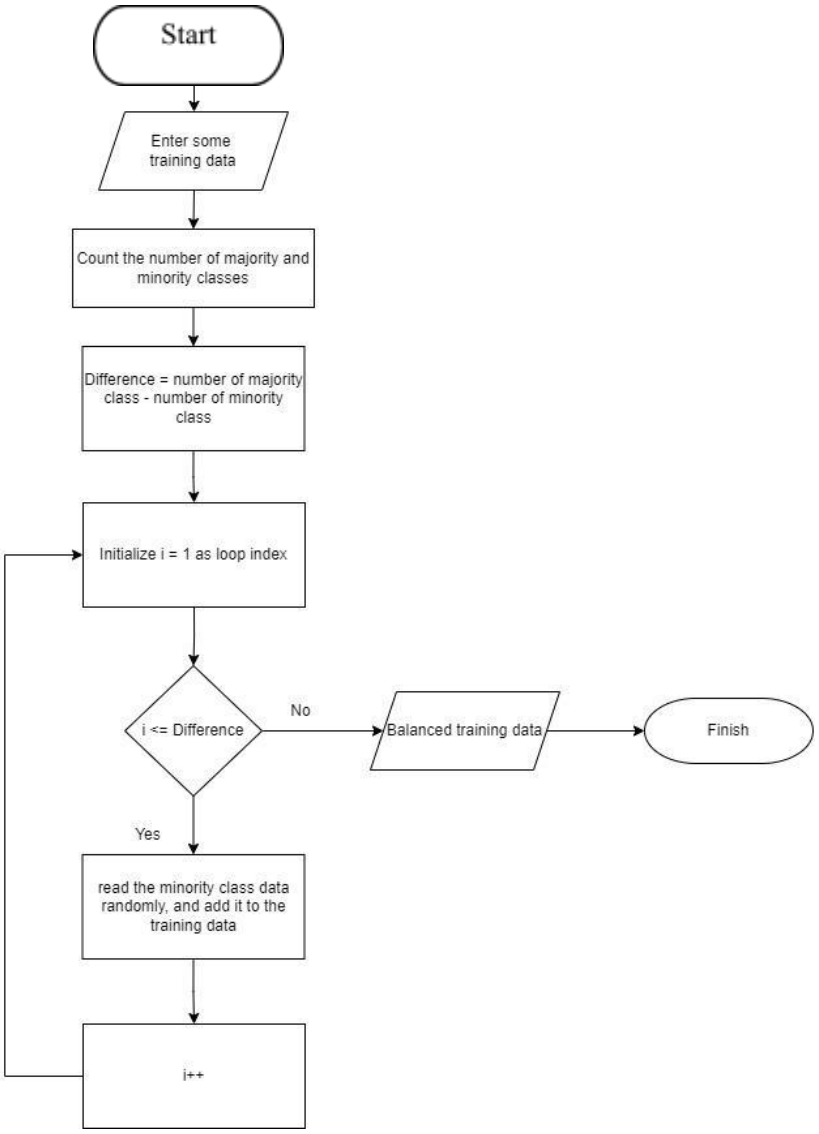

Figure 3. Oversampling techniques process

2.2 Synthetic Minority Oversampling Technique

SMOTE is a flexible data augmentation method that can be adapted to behavioral datasets directly and the basic idea is to analyze samples in the minority class and generate new samples based on the similarity of features between samples in the minority class (Chao & Zhang, 2021). SMOTE is an oversampling approach that was originally designed to handle unbalanced datasets (Gao et al., 2019). Figure 4 shows an illustration of the procedure performed by SMOTE.

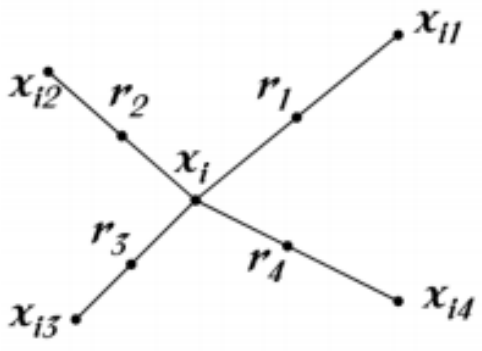

Figure 4. Illustration of The Procedure Performed by SMOTE

2.3 Random Forest Algorithm

Random forest is a powerful method for building ensembles of random decision tree (Cheng et al., 2019). Random forest is one of the most powerful ensemble methods with high performance when dealing with high-dimensional data (Nadi & Moradi, 2019). Error rate for random forest algorithm depends on two factors, namely (i) the error rate will increase if and only if the correlation between two trees from the forest will increase and (ii) the strength of the tree is determined by the lower error rate and which will strengthen the forest (Jaiswal & Samikannu, 2017). The following is an algorithm for random forest (Schonlau & Zou, 2020).

For $i \leftarrow 1$ **to** B **do**
  Draw a bootstrap sample of size N from the training data;
  While *node size* != *minimum node size* **do**
    randomly select a subset of m predictor variables from total p;
    for $j \leftarrow 1$ **to** m **do**
      if *jth predictor optimizes splitting criterion* **then**
        split internal node into two child nodes;
        break;
      end
    end
  end
end

2.4 Evaluation

At the evaluation stage, the accuracy is calculated using a confusion matrix. The calculation is done by calculating the number of correctly classified data divided by the number of predictions made. The steps are as follows.

1. Enter the results of the classification test in the confusion matrix table as shown in table 2.

Table 2. Confusion matrix table

		PREDICTED CLASS		
		Positive	*Negative*	Total
ACTUAL CLASS	*Positive*	*True Positive* (TP)	*False Negative* (FN)	TP+FN
	Negative	*False Positive* (FP)	*True Negative* (TN)	FP+TN
	Total	TP+FP	FN+TN	TP+FN+FP+TN

2. Calculate the accuracy value using Equation 1.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

3 State the conclusion from the accuracy results obtained.

4 Results and Discussion

In this study, the credit card fraud detection dataset is divided into training data and test data. The proportion of training data is 70% and test data is 30%. Broadly speaking, this research consists of two stages, for the first, the classification process uses a random forest algorithm on the dataset credit card fraud detection. Second, the credit card dataset classification process fraud detection using a combined random forest algorithm with PCA, resampling techniques and data augmentation (SMOTE). With the combination of these methods will produce a dataset that is balanced.

4.1 Splitting Data (Training and Testing)

After the data preprocessing has been carried out, the next stage is the data sharing or data splitting on the credit card fraud detection dataset. Splitting data is the process of dividing the dataset into training data and testing data using the "train_test_split" function from the sklearn library to split random datasets. The results of this splitting process are shown in Figure 5 where there is a configuration in the form of "test_size" which is the number of percentages of the number of datasets that will be used for data testing and "random_state" which is the value for reproducing random situations in order to create the same situation every time the program is run. “test_size” used in this research is 0.3 or 30% of the total dataset and "random_state" is the seed. The result of splitting the data is 199364 instances for training data and a total of 85443 instances for data testing.

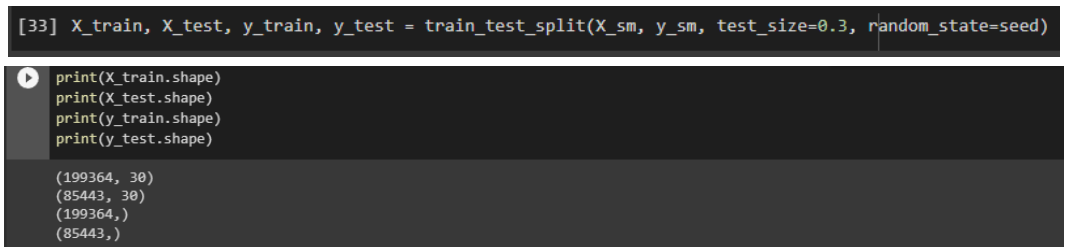

Figure 5. Results of the data splitting process

4.2 Results of Application Undersampling Technique

In this study, researchers used the undersampling technique. The best algorithm for undersampling is random undersampling. Undersampling stage used to randomly select a sample in the majority class and add it to the minority class, forming a new training dataset. The algorithm used is random undersampling which is implemented on the RandomUnderSampler object which has the function “undersample.fit_resample”. The use of these functions can be seen in figure 6. As shown in figure 7, before the process undersampling of the fraudulent minority class "1" (fraud transactions) has a total of 85443 while the number of normal majority classes "0" (normal transactions) is 199364 in the training data. After an undersampling process, the majority class has the same number as the minority class which is 85443.

```
X_over, y_over = undersample.fit_resample(X, y)
```

Figure 6. Use of function “undersample.fit_resample”

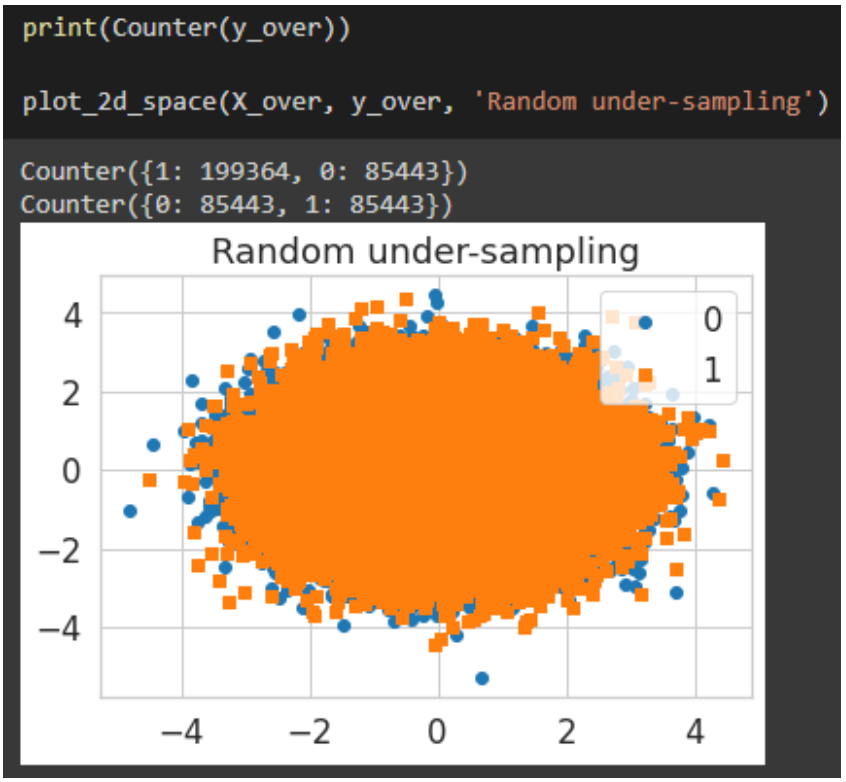

Figure 7. Undersampling results using random undersampling

4.3 Results of Application Oversampling Technique

The oversampling stage is used to produce a new sample on the training data so that the data in the minority class can have the same amount as the majority class. The algorithm used is random oversampling which is implemented on the RandomOverSampler object which has the "ros.fit_resample" function. The use of this function is shown in figure 8. As seen in figure 9, before the oversampling process the fraudulent minority class "1" (fraud transactions) had a total of 85443 while the number of normal majority classes "0" (normal transactions) had a total of 199364 in the training data. After the oversampling process, the minority class has the same number as the majority class, which is 199364.

```
X_ros, y_ros = ros.fit_resample(X, y)
```

Figure 8. Use of function "ros.fit_resample"

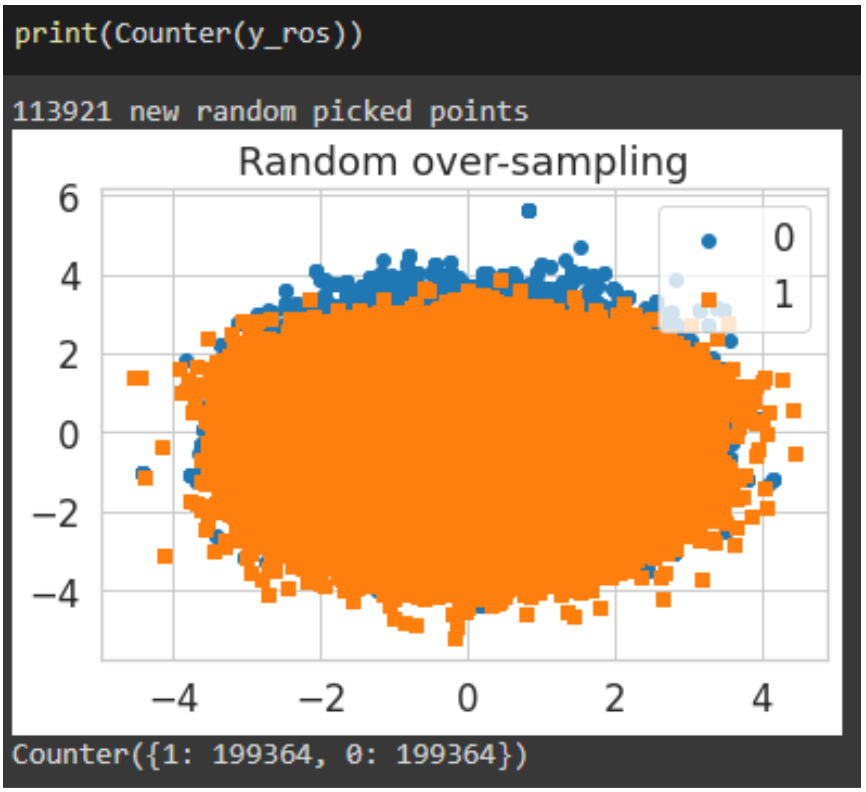

Figure 9. Oversampling results using random oversampling

4.4 Results of Application Data Augmentation

In this study, not only applying the resampling technique but also applying the data augmentation method. Data augmentation is a technique that allows the process of new training data from existing training data. The process carried out in the data augmentation stage is the same as that carried out in the oversampling stage. The data-level focus approach on data augmentation is to generate a new sample for the minority class. The data augmentation method generates multiple data points or feature points that fit into the minority class to rebalance the unbalanced original data set without losing data information. With this process, it makes the data more balanced. The algorithm used as data augmentation is the synthetic minority oversampling technique which is implemented on the SMOTE object which has the "smote.fit_resample" function. The use of this function can be seen in figure 10. As in figure 11, after the data augmentation process the number of fraudulent minority class data "1" (fraud transactions) and normal majority class data "0" (normal transactions) has the same amount, which is 199364.

```
X_sm, y_sm = smote.fit_resample(X, y)
```

Figure 10. Use of function "smote.fit_resample"

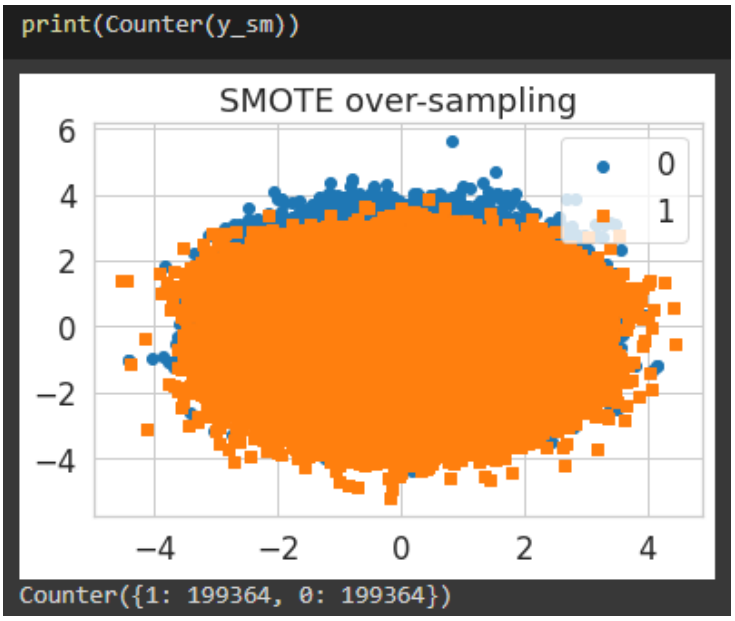

Figure 11. Data augmentation results using SMOTE

4.5 Results of Application Random Forest Classification Algorithm

The first process that was carried out was the classification of the random forest algorithm on the credit card fraud detection dataset. The number of features used are 30 features and 1 class attribute. The average time required for the classification process is 180 seconds. Evaluation of the performance of the random forest algorithm is measured using a confusion matrix. Table 3 shows the results of the random forest classification calculated using the confusion matrix.

Table 3. Confusion matrix random forest

	Positive	Negative	Amount
Positive	57012	2653	59665
Negative	6223	53731	59954
Amount	63235	56384	119619

$$Accuracy = \frac{TP+TN}{P+N} x\ 100\% \quad (2)$$

$$Accuracy = \frac{110743}{119619} x\ 100\% = 92,57\%$$

The accuracy results obtained from the random forest algorithm are 92.57%. Based on these results, it can be said that the random forest algorithm can detect credit card transactions well because the accuracy results are greater than the error value. The accuracy results may be increased if the data preprocessing process is carried out using several algorithms.

4.6 Results of Application of Random Forest Classification with Random Undersampling

The second process is the classification of the random forest algorithm on the credit card fraud detection dataset. The number of features used are 30 features and 1 class attribute. The average time required for the classification process is 210 seconds. Evaluation of the performance of the random forest algorithm is measured using a confusion matrix. Table 4 shows the results of the random forest classification with random undersampling calculated using the confusion matrix.

Table 4. Confusion Matrix Random Forest Algorithm with Random Undersampling

	True Positive	True Negative	Amount
Pred Positive	25896	0	25896
Pred Negative	2802	56745	59547
Amount	28698	56745	85443

$$Accuracy = \frac{TP+TN}{P+N} x\ 100\% \quad (3)$$

$$Accuracy = \frac{82641}{85443}\ x\ 100\% = 96,72\%$$

The accuracy results obtained from the random forest algorithm with random undersampling are 96.72%. The results of this accuracy may be increased if the data preprocessing process is carried out using other resampling techniques.

4.7 Results of Application of Random Forest Classification with Random Oversampling

The third process is classification using the random forest algorithm on the credit card fraud detection dataset with random oversampling. From the preprocessing process that has been carried out, 30 features were obtained as input values. A total of 30 features are then used for the classification process using the random forest algorithm. The average time required for the classification process is 250 seconds. The results of random forest classification by oversampling are then calculated using a confusion matrix as shown in table 5.

Table 5. Confusion matrix random forest algorithm with random oversampling

	True Positive	True Negative	Amount
Pred Positive	25668	0	25668
Pred Negative	2	59773	59775
Amount	25670	59773	85443

$$Accuracy = \frac{TP+TN}{P+N} x\ 100\% \quad (4)$$

$$Accuracy = \frac{85441}{85443}\ x\ 100\% = 99,9976\%$$

The accuracy results obtained from the preprocessing process carried out using random oversampling are 99.9976%. Based on these results, it can be said that the combination of the random forest algorithm with random oversampling on the credit card fraud detection dataset can improve the classification accuracy results by the random forest algorithm.

4.8 Results of Application of Random Forest Classification with SMOTE

The fourth process is classification with the random forest algorithm on the credit card fraud detection dataset with SMOTE. From the preprocessing process that has been carried out, 30 features were obtained as input values. A total of 30 features are then used for the classification process using the random forest algorithm. The average time required for the classification process is 300 seconds. The results of the random forest classification with SMOTE are then calculated using a confusion matrix as shown in table 6.

Table 6. Confusion matrix random forest algorithm with SMOTE

	True Positive	True Negative	Amount
Pred Positive	25668	0	25668
Pred Negative	2	59773	59775
Amount	25670	59773	85443

$$Accuracy = \frac{TP+TN}{P+N} x\ 100\% \quad (5)$$

$$Accuracy = \frac{85441}{85443}\ x\ 100\% = 99,9976\%$$

The accuracy results obtained from the preprocessing process carried out using SMOTE is 99.9976%. Based on these results, it can be said that the combination of the random forest algorithm with SMOTE obtains the same accuracy results as the random forest accuracy results with random oversampling.

4.9 Classification

In scenario I, the algorithm used provides a lower accuracy than scenario II, which is 92.57%. And the acquisition of the confusion matrix scenario I in table 3 shows that the model succeeded in classifying almost all negative classes of 53731 instances and was only wrong in 2653 instances, while in the positive class the model managed to classify as many as 57012 instances and failed to classify others as many as 6223 instances.

In scenario II, the combination of algorithms used gives a higher accuracy than scenario I, which is 96.72%. This shows that optimization using random undersampling has succeeded in increasing the performance of the random forest algorithm. With the acquisition of the confusion matrix scenario II, it shows that this model has succeeded in classifying almost all negative classes of 56745 instances and only 0 instances is wrong, while in the positive class the model has succeeded in classifying as many as 25896 instances and failed to classify others as many as 2802 instances.

In scenario III and scenario IV, the combination of algorithms used provides a higher accuracy than scenario II, which is 99.9976%. This shows that optimization using random oversampling and SMOTE has succeeded in increasing the performance of the random forest algorithm. With the acquisition of the confusion matrix scenario III and scenario IV, it shows that this model has succeeded in classifying almost all negative classes of 59773 instances and is only wrong in 0 instances, while in the positive class the model has succeeded in classifying 25668 instances and failed to classify the others as much as 2 instances.

To find out that the method used in this study is better than the previous research method, a comparison of the accuracy results obtained with previous classification studies using the same dataset, namely the credit card fraud detection dataset, was compared. Comparison of the accuracy results obtained with previous research can be seen in figure 12.

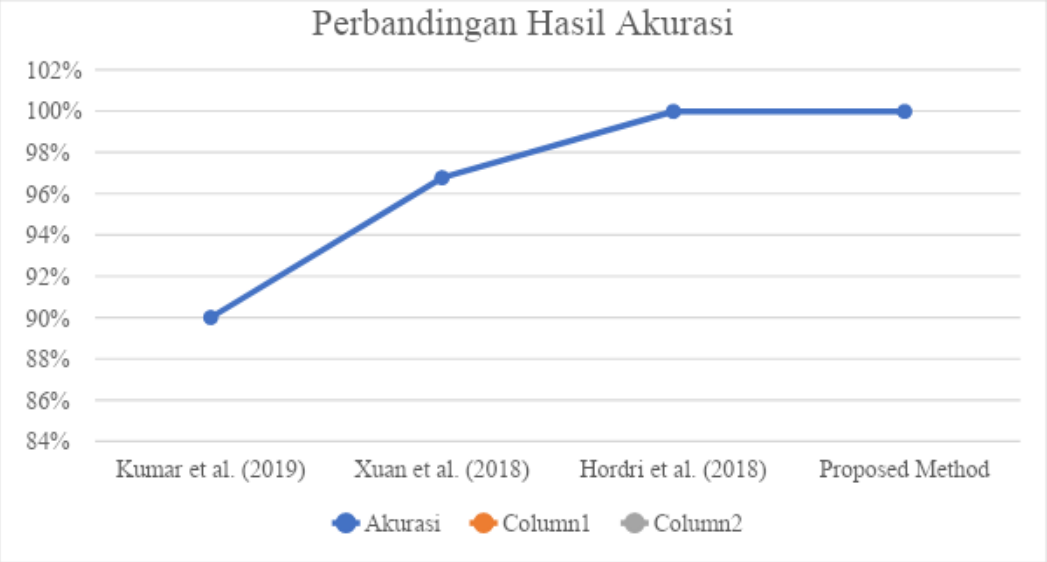

Figure 12. Comparison of accuracy results

5 Conclusion

The random forest algorithm is applied to the data classification process that has passed the preprocessing stage, namely the data transformation stage performed by PCA and data resampling performed by resampling and data augmentation techniques. Furthermore, the process of predicting the qualification of the type of transaction by the random forest algorithm is carried out with the final process, namely the calculation of the accuracy value using the confusion matrix. The workings of the PCA method in optimizing the random forest algorithm is to perform preprocessing (data transformation) on the dataset by converting features into numeric values. The resampling technique (random undersampling and random oversampling) with data augmentation (SMOTE) works on the training set to optimize the random forest algorithm by resampling data. The application of using PCA and resampling techniques with data augmentation is carried out in several stages, namely preprocessing which consists of data transformation and data resampling, splitting, training, and testing. The ratio of the distribution of datasets for training data and testing data is 70% and 30%. The highest accuracy results obtained in this study are found in the use of a combination of random forest algorithms using PCA and resampling techniques and data augmentation, which is 99.9976% with a balanced dataset. The accuracy result is higher than using the random forest algorithm alone, which is 92.57%.

6 References

AKKI. (2022). *Credit Card Growth*. Diakses pada tanggal 20 April 2022, dari https://www.akki.or.id/index.php/credit-card-growth

Alhakim, A., & Sofia. (2021). Kajian Normatif Penanganan Cyber Crime di Sektor Perbankan di Indonesia. *E-Journal Komunitas Yustisia*, *4*(2), 377–385.

Andrian, D., Soleh, A. M., & Wijayanto, H. (2018). Penerapan Metode Resampling dan K-Nearest Neighbor dalam Memprediksi Keberhasilan Studi Mahasiswa Program Magister IPB. *Xplore: Journal of Statistics*, *2*(1), 49–60.

Carneiro, N., Figueira, G., & Costa, M. (2017). A data mining based system for credit-card fraud detection in e-tail. *Decision Support Systems*, 1–36.

Chao, X., & Zhang, L. (2021). Few-shot imbalanced classification based on data augmentation. *Multimedia Systems*, 1–9.

Cheng, L., Chen, X., De Vos, J., Lai, X., & Witlox, F. (2019). Applying a random forest method

approach to model travel mode choice behavior. *Travel Behaviour and Society*, *14*, 1–10.

Devi, D., Biswas, S. K., & Purkayastha, B. (2019). A Cost-sensitive weighted Random Forest Technique for Credit Card Fraud Detection. *International Conference on Computing, Communication and Networking Technologies*, 1–6.

Devi, V., & Kavitha, K. S. (2017). Fraud Detection in Credit Card Transactions by using Classification Algorithms. *International Conference on Current Trends in Computer, Electrical, Electronics and Communication (ICCTCEEC)*, 125–131.

Finogeev, A. G., Parygin, D. S., & Finogeev, A. A. (2017). The convergence computing model for big sensor data mining and knowledge discovery. *Human-Centric Computing and Information Sciences*, *7*(11), 1–16.

Gao, R., Peng, J., Nguyen, L., Liang, Y., Thng, S., & Lin, Z. (2019). Classification of Non-Tumorous Facial Pigmentation Disorders Using Deep Learning and SMOTE. *IEEE Access*, 1–5.

Iwana, B. K., & Uchida, S. (2021). An empirical survey of data augmentation for time series classification with neural networks. *PLoS ONE*, *16*(7), 1–32.

Jaiswal, J. K., & Samikannu, R. (2017). Application of Random Forest Algorithm on Feature Subset Selection and Classification and Regression. *World Congress on Computing and Communication Technologies*, 65–68.

Jeihouni, M., Toomanian, A., & Mansourian, A. (2020). Decision Tree-Based Data Mining and Rule Induction for Identifying High Quality Groundwater Zones to Water Supply Management: a Novel Hybrid Use of Data Mining and GIS. *Water Resources Management*, *34*, 139–154.

Klapwijk, E. T., van de Kamp, F., van der Meulen, M., Peters, S., & Wierenga, L. M. (2019). Qoala-T: A supervised-learning tool for quality control of FreeSurfer segmented MRI data. *NeuroImage*, *189*, 116–129.

Kumar, M. S., Soundarya, V., Kavitha, S., Keerthika, E. S., & Aswini, E. (2019). Credit Card Fraud Detection Using Random Forest Algorithm. *3rd International Conference on Computing and Communication Technologies ICCCT*, 149–153.

Lebichot, B., Paldino, G. M., Siblini, W., He-Guelton, L., Oblé, F., & Bontempi, G. (2021). Incremental Learning Strategies for Credit Cards Fraud Detection. *International Journal of Data Science and Analytics*, *12*(2), 165–174.

Lebichot, Bertrand, Le Borgne, Y.-A., He-Guelton, L., Oblé, F., & Bontempi, G. (2020). Deep-Learning Domain Adaptation Techniques for Credit Cards Fraud Detection. *International Journal of Data Science and Analytics*, 78–88.

Lee, C. H., & Yoon, H. J. (2017). Medical big data: Promise and Challenges. *Kidney Research and Clinical Practice*, *36*, 3–11.

Lin, W., Wu, Z., Lin, L., Wen, A., & Li, J. (2017). An Ensemble Random Forest Algorithm for Insurance Big Data Analysis. *IEEE Access*, 2–8.

Nadi, A., & Moradi, H. (2019). Increasing the views and reducing the depth in random forest. *Expert Systems with Applications*, *138*, 1–13.

Schonlau, M., & Zou, R. Y. (2020). The random forest algorithm for statistical learning. *The Stata Journal*, *20*(1), 3–29.

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, *6*(60), 1–48.

Siringoringo, R. (2017). Integrasi Metode Resampling Dan K-Nearest Neighbor Pada Prediksi Cacat Softwar Aplikasi Android. *Jurnal ISD*, *2*(1), 47–58.

Sisodia, Di. S., Reddy, N. K., & Bhandari, S. (2017). Performance Evaluation of Class Balancing Techniques for Credit Card Fraud Detection. *IEEE International Conference on Power, Control, Signals and Instrumentation Engineering*, 2747–2752.

Utomo, D. P., & Mesran, M. (2020). Analisis Komparasi Metode Klasifikasi Data Mining dan Reduksi Atribut Pada Data Set Penyakit Jantung. *Jurnal Media Informatika Budidarma*, *4*(2), 437–444.

Xuan, S., Zheng, L., Liu, G., Wang, S., Li, Z., & Jiang, C. (2018). Random Forest for Credit Card Fraud Detection. *IEEE Access*, 1–6.

Yee, O. S., Sagadevan, S., & Malim, N. H. A. H. (2018). Credit Card Fraud Detection using Machine Learning as Data Mining Technique. *Journal of Telecommunication, Electronic and Computer Engineering*, *10*(1–4), 23–27.