Application of Smote and Decision Tree Classification in Detecting Fraudulent Transactions

Nora Mina [1], Eka Prakarsa Mandyarth[2], Agung Mustika Rizki[3]
[1,2,3] Informatics Master of Informatics, Faculty of Computer Science, UPN "Veteran" East Java,
Jl. Rungkut Madya No.1, Mt. Anyar, Surabaya
noramina261@gmail.com

ABSTRACT

Fraud detection in online transactions is critical to protecting consumers and maintaining the integrity of the online business ecosystem. Dataset imbalance can affect the classification prediction performance. To overcome data imbalance, this research uses an oversampling approach with the SMOTE method. The aim of this research is to analyze the performance of the SMOTE algorithm and decision tree classification in dealing with data imbalance problems in fraudulent transactions. The dataset used is online payments taken from Kaggle. The dataset shows that there are unbalanced classes, and it was found that using the SMOTE method increased the performance value better than using it without the SMOTE method. Using SMOTE gets very high metric values, up to a recall value of 100%. This shows that the model used in classifying fraudulent transactions is very effective.

Keywords: Transactions, Cheating, Classification, SMOTE, Decition Tree

INTRODUCTION

The development of technological advances in making transactions brings serious challenges, namely fraud cases as a form of fraud that can provide financial losses from both customers and merchants. Fraud can be committed in the form of identity card theft or the use of fake credit cards. Therefore, to overcome this, fraud detection is carried out to protect customers and maintain the integrity of the payment system (Armiani & Agustini, 2022). Efforts made in detecting online transaction fraud with the Knowledge Discovery in Database (KDD) process aim to identify significant patterns, trends, or relationships in data that may have been previously unknown. With the application of the KDD process helps extract patterns from data using algorithms for classification, which predict the characteristics of the data based on different datasets (Guo et al., 2018). However, the challenge in classification is often the problem of data imbalance, which can affect the accuracy of the results. The use of classification algorithms without considering the balance of data between classes can lead to accurate predictions only for the majority class, while the minority class is ignored. Direct implementation of classification algorithms on unbalanced datasets will result in overall performance degradation (Selfiani et al., 2022).

Data imbalance refers to the difference in the ratio between the amount of data in one class and another. In general, an unbalanced dataset is characterized by an unbalanced ratio of majority data to minority data. There are 2 ways to deal with the problem of data imbalance, namely oversampling and undersampling. Oversampling is one of the strategies by taking or adding samples from minority classes. Meanwhile, undersampling is a strategy with the elimination of samples from the majority class (Cahyaningtyas et al., 2021).

In this study, the researcher will overcome the class imbalance by focusing on minority classes using a resampling technique called oversampling. The decision to use oversampling was based on its ability to augment samples in minority classes without reducing the dataset. One of the oversampling algorithms used is the Synthetic Minority Over-sampling Technique (SMOTE), chosen because of its superiority in producing high accuracy and its effectiveness in dealing with class imbalances by reducing the risk of overfitting. The goal of this is to achieve a balance between classes in the dataset, thereby improving the performance of the classification method.

The classification used in this study is the Decision Tree method. Decision Tree, commonly known as Decision Tree, has become a very popular classification method and is recognized for its effectiveness(Franseda et al., 2020). This method is efficient in processing large datasets into a set of rules that are represented in the form of decision trees. One of its main advantages is its ability to generate rules that are easy to understand in everyday language, making it an easy-to-learn and frequently used classification technique. The main purpose of this method is to group objects or data by utilizing the structure of the tree as its representation.

The background in this study is to compare 2 test scenarios, namely, the test scenario without the SMOTE algorithm and assisted by the SMOTE algorithm. This aims to show the influence of the use of the SMOTE algorithm on the Decision Tree classification in detecting online transaction fraud.

LITERATURE REVIEW

Online Transactions

Online transactions refer to the process of buying and selling goods or services that are carried out through the internet or electronic platforms. In online transactions, there are basic principles that support the operation, security, and effectiveness of electronic commerce.

Security is the main cornerstone, which is achieved through the use of encryption technology to maintain the integrity and confidentiality of data during transmission. User authorization and authentication are essential to ensure that only authorized parties can access and conduct transactions, online transaction theory includes strategies to ensure system reliability, such as establishing a high level of service(Dian Fitri Mellina & Ainul Yaqin, 2024).

Cheating

Cheating is behavior that involves dishonest or manipulative actions with the aim of gaining advantage or deceiving others. Theories about cheating include several aspects, including ethics, the psychology of human behavior, and legal considerations (Sari et al., 2022). Fraud can be a variety of actions, such as fraud, manipulation, or violations of business ethics. To tackle fraud, a holistic approach is needed that includes ethics education, increased surveillance, and effective law enforcement. By understanding the theory behind cheating, society can develop better strategies to prevent and deal with dishonest behavior, maintain integrity, and create a fair and ethical environment.

Classification

Classification is a data analysis process that aims to build a model that can describe and differentiate different classes of data, so that the model can be used to predict classes from data that has not been labeled. The classification process consists of two main stages: the learning stage (training phase) and the testing stage (testing phase). In the learning stage, classification algorithms are used to analyze training data and form classification models. Meanwhile, at the testing stage, the classification model that has been formed is tested on the test data to assess the accuracy of the model (Trisnanto, 2023)

Data Imbalance

Data imbalance occurs when the amount of data in one class is much larger or smaller than another class (Fauziningrum, M.Pd & Sulistyaningsih, 2021). This imbalance is a challenge in the application of machine learning because it can lead to classification errors, reduce model accuracy, and often ignore minority classes by considering them as outliers. To overcome this problem, various classification strategies and algorithms can be applied. There are two common approaches to dealing with data imbalances:

1. Undersampling

A method that reduces the number of samples from the majority class to achieve balance with the minority class.

2. Oversampling

A technique that increases the number of samples in minority classes. One of the well-known oversampling methods is the Synthetic Minority Oversampling Technique (SMOTE).

SMOTE (Synthetic Minority Oversampling Technique)

Synthetic Minority Oversampling Technique, is a technique used to overcome the Class Imbalance Problem (CIP) problem in the dataset. In contrast to traditional oversampling methods that only duplicate minority data randomly, SMOTE generates new synthetic data to improve the representation of minority classes (Chawla et al., 2002). The main goal of SMOTE is to balance the amount of data between minority classes and majority classes by creating new data examples that synthetically expand minority classes, rather than just doubling them.

Decision Tree

Decision trees, or decision trees, are a very popular classification method and are well-known for their effectiveness. This method is able to convert large datasets into an easy-to-understand set of rules, represented in the form of tree structures. One of the main advantages of decision trees is that the resulting rules can be easily explained in natural language (Maulidah et al., 2020) Decision trees are used to group data or objects based on the structure of the tree that represents that data.

The decision tree consists of three main components, as described by (Franseda et al., 2020):

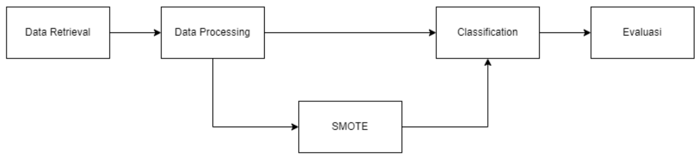
1. The root node is located at the very top of the decision tree and usually represents a variable or feature that starts the data sharing process.
2. Internal nodes that are branching points in the tree, receive one input and produce at least two outputs, which indicate different decision paths based on certain criteria.
3. Leaf Node The final node where the classification or decision process stops. Leaf nodes have only one input and do not produce any further outputs.

k Fold Cross Validation

k fold Cross Validation is a method used to assess the performance of a classifier by dividing the data into several parts or folds, as well as a technique used to measure the quality of the model (Zamachsari & Puspitasari, 2021) . In this method, the number of folds, represented by the value k, plays an important role because the dataset will be divided into k equal parts. This procedure involves dividing the dataset into two main groups: training data and test data. Training data is used to train the model, while test data is used to evaluate the model's performance. This method allows for a more comprehensive evaluation of the model as each piece of data is used for testing, which helps in understanding how the model can be expected to perform on data that has never been seen before.

METHODS

Figure 1 Research Stages



Data Retrieval

The data used in the study was taken from Kaggle's official platform, namely the "Online Payments Fraud Detection Dataset". The total data used amounted to 6,32,620 data. It consists of 11 attributes including step, type, amount, nameOrig, oldbalanceOrg, newbalanceOrg, nameDest, oldbalanceDest, newbalanceDest, isFraud, isFlaggedFraud. And it has 2 classes, namely the Cheating transaction class and marked Cheating, where a value of 0 is considered a non-cheating class and class 1 is a cheating class.

Data Processing

The data to be used is processed first to ensure that the data used is representative, clean, and suitable for use in training the model. There are several things that are done in the data processing process, including;

a. Conducting analysis of inappropriate transactions

The dataset used is transaction data, so that transactions must certainly be adjusted to transaction logic. Where an examination is carried out between the amount paid and the final balance. Table 1 shows the percentage of data results that do not conform to the supposed transaction logic. 85% of the transaction logic data from the sender does not match as it should. Meanwhile, from the recipients, there are 75% of the data that does not match. The high percentage of non-compliant transaction logic deteriorates the quality of the data. Therefore, it is necessary to update the value of the 'final balance of the sender' column with the calculation result (initial balance of the sender – amount) and update the value of the 'final balance of the recipient' column with the result of the calculation (initial balance of the sender + amount). Figure 2 shows some data that did not match before it was updated, while Figure 3 shows changes in the updated data

Figure 2 Transaction data does not match.

Incorrect Sender Transaction Line			
No.	**oldbalanceOrg**	**Amout**	**newbalanceOrg**
8	2671.0	4024.36	0.0
9	41720.0	5337.77	0.0
10	4465.0	9644.94	0.0
Incorrect Receiver Transaction Line			
No.	**oldbalanceDest**	**Amout**	**newbalanceDest**
0	0.0	9839.64	0.0
1	0.0	1864.28	0.0
2	0.0	181.00	0.0

Figure 3 Transaction data matches

Incorrect Sender Transaction Line			
No.	**oldbalanceOrg**	**Amout**	**newbalanceOrg**
8	2671.0	4024.36	-1353.36
9	41720.0	5337.77	36382.23
10	4465.0	9644.94	-5179.94
Incorrect Receiver Transaction Line			
No.	**oldbalanceDest**	**Amout**	**newbalanceDest**
0	0.0	9839.64	9839.64
1	0.0	1864.28	1864.28
2	0.0	181.00	181.00

b. Calculate the number of fraudulent and non-fraudulent transactions

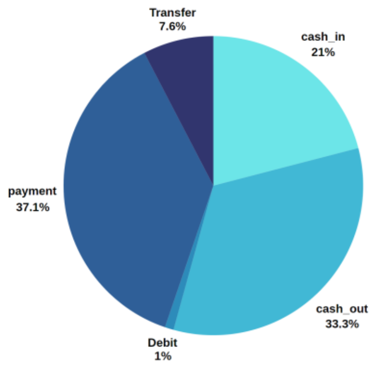
Table 1. Number of Classes

Class	**Sum**	**Data**
0	6354407	*Not Cheating*
1	8213	*Cheating*

c. Remove multiple payment types

There are 5 types of payment types in the data. Figure 3 shows the percentage of the number of payment types that are made.

Figure 4. Payment Type



From the results of the analysis of the type of payment of fraudulent transactions, there are only cash_out *and transfer* payment types. So that the other 3 types of payments were removed to overcome the problem of data imbalance.

d. Removing a class 'marked as Cheating'

The removal of the 'marked as fraudulent' class was carried out because the research focused on transactions where there was absolute fraud taken from the 'fraudulent transactions' column.

e. Remove some attributes

This is to focus the data on the features that are important, so that it will help in improving performance. Figure 4 shows the results of the display of clean data after data processing.

Figure 5. Data

No	type	amount	Oldbalance Orig	Newbalance Orig	Oldbalance Dest	Newbalance Dest	isFraud
0	1	181.00	181.00	0.00	0.0	181.00	1
1	0	181.00	181.00	0.00	21182.0	21363.00	1
2	0	229133.94	15325.0	-213808.94	5083.0	234216.94	0
3	1	215310.30	705.0	-214605.30	22425.0	237735.30	0
4	1	311685.89	10835.0	-300850.89	6267.0	317952.89	0

Dataset Sharing

The training dataset is classified into two, namely cheating and non-cheating. The dataset is divided into 2 parts. The first part is used as a training dataset to develop a classification model, while the second part is used as a test dataset to test and evaluate the performance of the model that has been created. The research focused on the sharing of 60% of the training data and 40% of the test data. The test scenario model of the dataset division is described in Table 2.

Table 2. Split Data

No.	Training Data	Test Data	Algorithm
1.	60%	40%	non-SMOTE + Decision Tree
2.	60%	40%	SMOTE (ratio: 0.5) + Decision Tree
3.	60%	40%	SMOTE (ratio: 1.0) + Decision Tree

Modelling

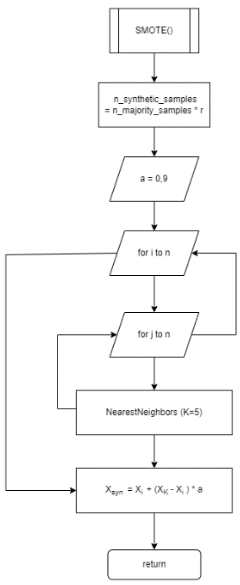
The model used in the problem for data imbalance is the Synthetic Minority Over-sampling Technique (SMOTE). SMOTE as one of the oversampling methods aims to increase the amount of data in minority classes by adding random data replicas so that the number is balanced or close to the amount of data in the majority class. The SMOTE algorithm operates by searching for K-Nearest Neighbors, where data is grouped based on proximity to the nearest neighbors. The selection of the nearest neighbor is done by calculating the Euclidean distance between two data, e.g. *pi* = [*p1*,*p2*...,*pi*], and *qi* = [*q1*,*q2*...,*qi*], so that the Euclidean distance from (p, q) is calculated by equation 1.

$$d(p, q) = \sqrt{(p_1 - q_1) + (p_2 - q_2) \ldots (p_i - q_i)}$$

Meanwhile, for synthetic data, it is built with equation 2.

Xsyn = Xi + (Xk – Xi) * a

Figure 6 SMOTE Algorithm



Information:
n is the number of samples added

K is the number of nearest neighbors

A is a random number (alpha) between 0 – 1 with a determined value of 0.9

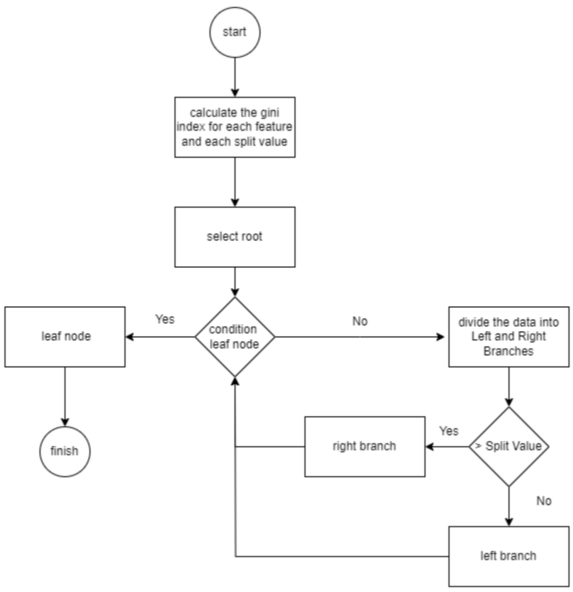
Xsyn is the data of replication (synthesis data)

Xi is the ith data of the minority class

Xk is the data of the minority class that has the closest distance from Xi

The stage flow performed by the SMOTE algorithm is shown in figure 5. After the SMOTE process is carried out, it is continued with the Decision Tree classification. The first step to modeling using a decision tree is to separate features from labels. To perform this separation, calculations using the Gini Index are required. The Gini Index is a popular method for measuring the homogeneity or cleanliness of a node in a decision tree, which plays an important role in machine learning algorithms for classification. The Gini Index measures how well it determines the best separation point on a data feature during decision tree construction (*decision tree*). Figure 6 shows the algorithm used in building the decision tree (Syukron et al., 2023).

Figure 7 Decision Tree Algorithm



- Formula for calculating Gini Index

$$G = 1 - \sum_{i=1}^{k} p_i^2$$

Information:

- G is the Gini Index

- K is the total number of classes or categories in the dataset
- *pi* is the proportion or probability of class i in the dataset. This is calculated as the number of items in class i divided by the total number of items.

- Formula for calculating the Gini Split

$$G_{split} = \frac{n\,left}{n}\,G_{left} + \frac{n\,right}{n}\,G_{right}$$

Information:

- n is the total number of samples on the node before the split.
- nleft and n_{right} are the number of samples on the left and right of the separation.
- Gleft and G_{right} are the Gini values for the left group of the right fund.

Evaluation

To evaluate the overall performance of the model, *K-Fold Cross Validation is implemented*. In this method, the value of k has a significant impact because the data will be divided into k parts. The procedure of this method involves dividing the dataset into two main parts, namely training data and test data. To help ensure the performance of the model, a confusion matrix analysis was also carried out as shown in Table 3.

Table 3. Confusion Matrix

True Negative (TN)	False Positive (FP)
False Negative (FN)	True Positive (TP)

RESULTS

The number of data samples used with and without smote from all three tests is shown in table 4. From the results of the tests carried out, the model performance values with confusion matrix from the first to the third scenario were produced and grouped in table 5. The test results in table 5 show that the use of SMOTE improves the performance value of the model, while the ratio difference scenario of the SMOTE algorithm has the same performance value. Table 6 shows five data samples (minority data and selected closest neighbor data) used in the SMOTE method, while table 7 shows the five results of the synthesis data that were successfully created.

Table 4. Number of Test Samples

Data	NonSmote	SMOTE (0.5 ratio)	SMOTE (1.0 ratio)
0 (Majority Class)	2762196	2762196	2762196

1 (Minority Class	8213	1381098	2762196

Table 5. Performance Result Values

Testing	Accuracy	Precision	Recall	F1 Score
Scenario 1	99%	85%	32%	47%
Scenario 2	99%	99%	100%	99%
Scenario 3	99%	99%	100%	99%

Table 6. Sample Data Used by SMOTE

Selected Minority Data	Nearest Neighbor Data	Results of Data Discrepancy between Minorities and Nearest Neighbors
5925th Data	-3629th Data	[0 -400.09 -400.09 0 0 -400.09]
-500th data	Data from 1990	[0 542.56 542.56 0 0 542.56]
919th Data	-5770th data	[0 -501.90 501.90 0 -46382.20 -46884.11]
-4859th Data	-4860th data	[-1 0 0 0 0 0]
263rd Dara	Data -7013	[0 -31.369 -31.369 0 0 -31.369]

Table 7. SMOTE Data Results

No.	Data Synthesis (SMOTE)
1.	[1 136125.36 136125.36 0 0 136125.36]
2.	[1 350952.36 350952.36 0 0 350952.36]
3.	[0 3284047.307 3284047.307 0 0 187027.417 3284047.307]
4.	[1 673722.73 673722.73 0 0 673722.73]
5.	[1 146117.689 146117.689 0 0 146117.689]

The decision tree is built by calculating the lowest Gini index as the *root node*. Then divide the *root node* into 2 branches until the *leaf node* condition is met. As for the parameters used in building the decision tree by limiting the tree depth value = 3, the minimum number of samples used for separation is 1000 samples and the minimum *leaf node* sample has 500 samples. A low Gini value in many nodes indicates that the separation is good enough at isolating one of the classes, while a high Gini value indicates uncertainty or mixing between classes. The results of *the tree* in all three test scenarios showed that darker colors showed strong dominance by one class, while lighter or mixed colors showed imbalance or impurity.

Figure 8 Tree Scenario 1

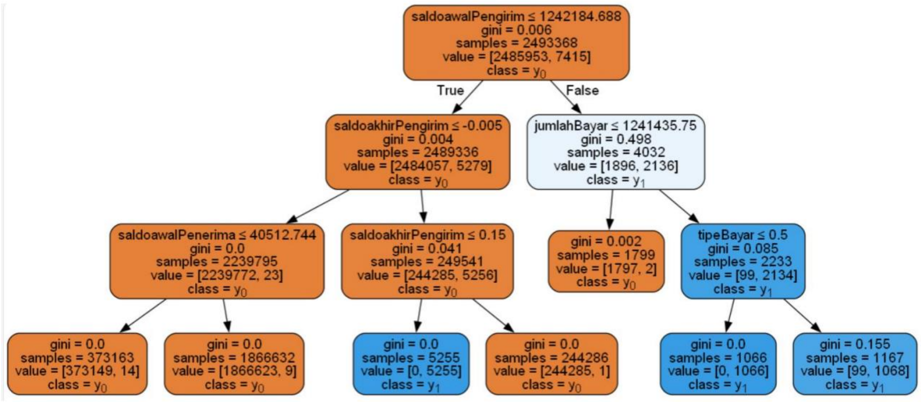


Figure 9 Tree Scenario 2

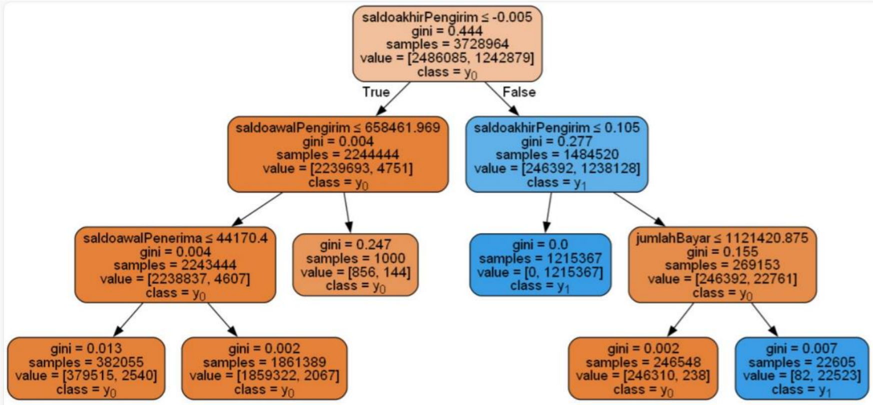
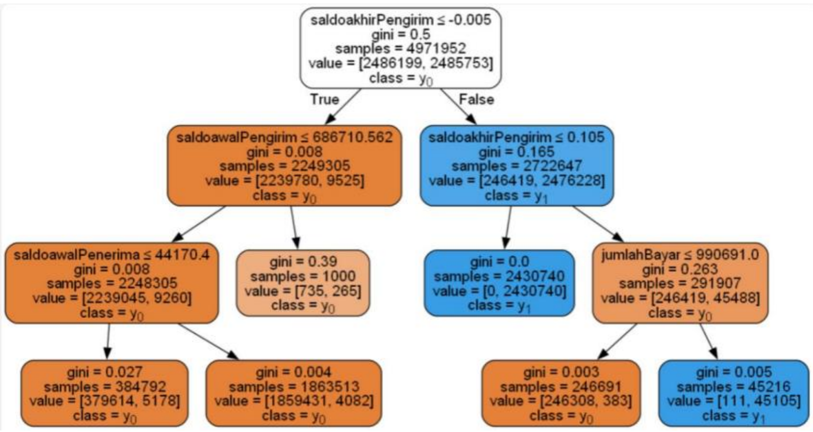


Figure 10 Tree Scenario 3



In measuring the performance evaluation of the model that has been carried out, *k-fold cross validation* is applied with a value of k=5, then calculating the matrix confusion analysis from the average results obtained is shown in table 7.

Table 7. Results of k fold cross validation

Test Scenarios	Average yield	Confusion Matrix
Test Scenario 1	0.99995	[[2762096 100] [32 8181]]
Test Scenario 2	0.99864	[[2762092 104] [5525 1375573]]
Test Scenario 3	0.99798	[[2762075 121] [11012 2751184]]

DISCUSSION

Based on the results of the research carried out, the test scenario carried out without the use of the SMOTE method affects the value of the evaluation metrics. Where the number of class 0 is much higher than that of class 1, the model tends to predict the dominant class, which results in errors in predicting minority classes (class 1). Although the classification report shows high accuracy, this number can be misleading because it does not describe the model's performance in handling minority class predictions correctly. Meanwhile, the use of scenarios with the addition of synthetic data from SMOTE increases the accuracy value to perfection until it gets *a 100%* recall value.

CONCLUSION

Based on the results obtained from the research, the results of data that have been processed and trained using a model obtained a stable metric value or accuracy with a value of 99%. This shows that the model is very suitable and reliable in classifying fraudulent transactions from the dataset used. However, the use of SMOTE in balancing data can correct classification errors and improve the value of other metrics. The evaluation results of *k fold cross validation* also show that the model has good performance so that it also gets a high accuracy value of 99%. The results of *k fold cross validation* show that the use of SMOTE adds an increase in *false negatives*. This shows that the creation of SMOTE synthesis data is insufficient in maintaining the original datset pattern.

REFERENCES

Arifiyanti, A. A., & Wahyuni, E. D. (2020). Smote: Metode Penyeimbang Kelas Pada Klasifikasi Data Mining. *SCAN - Jurnal Teknologi Informasi Dan Komunikasi*, *15*(1). https://doi.org/10.33005/scan.v15i1.1850

Armiani, R., & Agustini, E. P. (2022). Analisa Fraud Pada Transaksi Kartu Kredit Menggunakan Algoritma Random Forest. *Jurnal Teknologi Informasi Dan Terapan*, *9*(2), 118–126. https://doi.org/10.25047/jtit.v9i2.297

Cahyaningtyas, C., Nataliani, Y., & Widiasari, I. R. (2021). Analisis Sentimen Pada Rating Aplikasi Shopee Menggunakan Metode Decision Tree Berbasis SMOTE. *Aiti*, *18*(2), 173–184. https://doi.org/10.24246/aiti.v18i2.173-184

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). snopes.com: Two-Striped Telamonia Spider. *Journal of Artificial Intelligence Research*, *16*(Sept. 28), 321–357. https://arxiv.org/pdf/1106.1813.pdf%0Ahttp://www.snopes.com/horrors/insects/telamonia.asp

Dian Fitri Mellina, A., & Ainul Yaqin, M. (2024). Algoritma Decision Tree untuk Prediksi Deteksi Penyakit Kanker Payudara. *Jurnal Informatika Sunan Kalijaga)*, *9*(1), 70–78.

Fauziningrum, M.Pd, E., & Sulistyaningsih, E. I. (2021). Penerapan Data Mining Metode Decision Tree Untuk Mengukur Penguasaan Bahasa Inggris Maritim (Studi Kasus Di Universitas Maritim Amni). *Jurnal Sains Dan Teknologi Maritim*, *22*(1), 41. https://doi.org/10.33556/jstm.v22i1.285

Franseda, A., Kurniawan, W., Anggraeni, S., & Gata, W. (2020). Integrasi Metode Decision Tree dan SMOTE untuk Klasifikasi Data Kecelakaan Lalu Lintas. *Jurnal Sistem Dan Teknologi Informasi (Justin)*, *8*(3), 282. https://doi.org/10.26418/justin.v8i3.40982

Guo, Y., Han, S., Li, Y., Zhang, C., & Bai, Y. (2018). K-Nearest Neighbor combined with guided filter for hyperspectral image classification. *Procedia Computer Science*, *129*, 159–165. https://doi.org/10.1016/j.procs.2018.03.066

Maulidah, M., Windu Gata, Rizki Aulianita, & Cucu Ika Agustyaningrum. (2020). Algoritma Klasifikasi Decision Tree Untuk Rekomendasi Buku Berdasarkan Kategori Buku. *E-Bisnis : Jurnal Ilmiah Ekonomi Dan Bisnis*, *13*(2), 89–96. https://doi.org/10.51903/e-bisnis.v13i2.251

Samuel, Y. T., & Nahuway, C. B. A. (2020). Prediksi Indeks Prestasi Mahasiswa Yang Berkuliah Sambil Bekerja Di Universitas Advent Indonesia Dengan Menggunakan Metode Decision Tree C4.5 Dan Smote. *TeIKa*, *10*(01), 69–77. https://doi.org/10.36342/teika.v10i01.2281

Sari, E. P., Febrianti, D. A., & Fauziah, R. H. (2022). Fenomena Penipuan Transaksi Jual Beli Online Melalui Media Baru Berdasarkan Kajian Space Transition Theory. *Deviance Jurnal Kriminologi*, *6*(2), 153. https://doi.org/10.36080/djk.1882

Selfiani, S., Prihanto, H., Yulaeli, T., & Moestopo, H. J. (2022). Analisa Potensi Kecurangan Pada Praktik Belanja Online. *Jurnal Manajemen Dan Bisnis*, *2*(1), 88–98. https://doi.org/10.32509/jmb.v2i1.2004

Syukron, A., Sardiarinto, S., Saputro, E., & Widodo, P. (2023). Penerapan Metode Smote Untuk Mengatasi Ketidakseimbangan Kelas Pada Prediksi Gagal Jantung. *Jurnal Teknologi Informasi Dan Terapan*, *10*(1), 47–50. https://doi.org/10.25047/jtit.v10i1.313

Trisnanto, P. (2023). Konseptual Desain Alat Sensor Map Dokumen Rekam Medis: Konseptual Desain Alat Sensor Map Dokumen Rekam Medis. *Jurnal Teknologi …*, *September*. https://doi.org/10.1980/jurnalteknologikonseptualdesign.v1i1

Zamachsari, F., & Puspitasari, N. (2021). Penerapan Deep Learning dalam Deteksi Penipuan Transaksi Keuangan Secara Elektronik. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *5*(2), 203–212. https://doi.org/10.29207/resti.v5i2.2952