JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i4.6420

APPLICATION OF MACHINE LEARNING MODELS FOR FRAUD DETECTION IN SYNTHETIC MOBILE FINANCIAL TRANSACTIONS

Imam Mulyana[1*]; Muhamad Bahrul Ulum[2]

Informatics Engineering[1,2]
Universitas Esa Unggul Jakarta, Indonesia[1,2]
https://www.esaunggul.ac.id[1,2]
mulyanaimam6@student.esaunggul.ac.id[1*], m.bahrul_ulum@esaunggul.ac.id[2]

(*) Corresponding Author
(Responsible for the Quality of Paper Content)

Abstract— *The financial industry faces challenges in detecting fraud. The 2023 Basel Anti-Money Laundering (AML) Index report shows a worsening money laundering risk trend over the last five years in 107 countries. And according to the Financial Action Task Force (FATF) in 2023, this is exacerbated by financial institutions which have problems with low reporting of suspicious financial transactions (Suspicious Transaction Report). Limited access to confidential financial transaction data is an obstacle in developing machine learning-based fraud detection models. To overcome this challenge, the research uses PaySim synthetic datasets that mimic real financial transaction patterns. The CRISP-DM approach is used, including the Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Deployment stages. The algorithms used are Decision Tree, Random Forest, and XGBoost. Model evaluation is carried out using accuracy, precision, recall, F1-score, specificity, cross-validation and ROC-AUC metrics. The results show that the Random Forest algorithm has the best performance with 99% accuracy, followed by XGBoost (98.9%) and Decision Tree (97%). Data analysis shows that cash-out and transfer transactions have the highest risk of fraud. This model has proven effective in detecting suspicious financial transactions with a high level of accuracy. This research makes a significant contribution to mitigating financial risks, supporting anti-fraud policies, and encouraging innovation in fraud detection using synthetic data.*

Keywords: *anti-money laundering, fraud detection, machine learning, paysim, random forest, synthetic datasets.*

Intisari— *Industri keuangan menghadapi tantangan dalam mendeteksi fraud. Laporan Indeks Basel Anti-Money Laundering (AML), pada tahun 2023 menunjukkan tren risiko pencucian uang yang memburuk selama lima tahun terakhir pada 107 negara. Dan menurut Financial Action Task Force (FATF) pada tahun 2023, hal ini diperparah oleh lembaga keuangan yang memiliki masalah dalam rendahnya pelaporan transaksi keuangan mencurigakan (Suspicious Transaction Report). Keterbatasan akses terhadap data transaksi keuangan yang bersifat rahasia menjadi hambatan dalam pengembangan model fraud detection berbasis machine learning. Untuk mengatasi tantangan ini, penelitian menggunakan dataset sintetis PaySim yang meniru pola transaksi keuangan asli. Pendekatan CRISP-DM digunakan, mencakup tahap Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, dan Deployment. Algoritma yang digunakan adalah Decision Tree, Random Forest, dan XGBoost. Evaluasi model dilakukan menggunakan metrik accuracy, precision, recall, F1-score, specifity, cross-validation dan ROC-AUC. Hasil menunjukkan bahwa algoritma Random Forest memiliki performa terbaik dengan akurasi 99%, diikuti oleh XGBoost (98,9%) dan Decision Tree (97%). Analisis data menunjukkan bahwa jenis transaksi cash-out dan transfer memiliki risiko fraud tertinggi. Model ini terbukti efektif dalam mendeteksi transaksi keuangan mencurigakan dengan tingkat akurasi tinggi. Penelitian ini memberikan kontribusi signifikan dalam mitigasi risiko keuangan, mendukung kebijakan anti-fraud, dan mendorong inovasi deteksi fraud menggunakan data sintetis.*

Kata Kunci: *anti pencucian uang, deteksi penipuan, pembelajaran mesin, paysim, hutan acak.*

INTRODUCTION

Threats to the financial industry have become increasingly complex in recent years, especially in the form of organized fraudulent activities. Crimes such as money laundering and illegal transactions not only harm financial institutions but also threaten global economic stability. The Basel Anti-Money Laundering (AML) Index Report 2023 shows a worsening trend of money laundering risks in 107 countries [1], reflecting significant challenges in maintaining the integrity of the financial system. In addition, the low level of suspicious transaction reporting by financial institutions further exacerbates the situation, as highlighted in the Financial Action Task Force (FATF) report 2023.

Amid the urgency to detect suspicious financial activity faster and more accurately, conventional methods often fail to address the complexity of modern fraud patterns. Machine learning technologies have emerged as a promising approach [2], offering the ability to recognize hidden patterns in financial transaction data. However, these efforts face major obstacles related to limited access to the original transaction data, which is confidential and closely monitored by regulatory authorities [3]. Alternatively, synthetic data such as that generated by the PaySim simulator has provided a new opportunity to overcome these limitations. The data mimics the characteristics of mobile financial transactions, allowing for the testing of fraud detection models without violating data privacy [3], [4], [5].

In this study, the main focus is on utilizing synthetic data sets to develop machine learning models that can detect suspicious transactions quickly and accurately [6]. Using a quantitative approach, this study aims to design, test, and evaluate algorithm-based fraud detection models such as Decision Tree, Random Forest, and XGBoost [7], [8][9]. To ensure the effectiveness of the developed model, this study applies the Random Undersampling technique to overcome class imbalance in transaction data.

This method works by randomly reducing the number of samples from the majority class so that the class distribution becomes more balanced, without adding synthetic data. Although this approach can reduce the total amount of training data, this technique is effective in avoiding overfitting that often occurs in the oversampling method [10]. This approach not only offers a practical solution for financial institutions in identifying fraud risks, but also opens up opportunities for further development in technology-based fraud detection research. By integrating technological innovation and in-depth understanding of financial transaction patterns, this research is expected to provide real contributions in overcoming the challenges of modern financial crime.

In addition, hyperparameter optimization is performed using the Grid Search method to adjust key parameters such as the number of decision trees, maximum depth (max_depth), and learning rate. With this approach, the model is expected to work optimally and more accurately in detecting fraudulent transactions. It is important to note that fraud patterns continue to evolve, causing concept drift in financial transaction data.

Therefore, this fraud detection model is designed to be regularly updated with the latest data to remain relevant in identifying new modus operandi used by financial criminals. As a comparison, this study also uses a literature study of the Logistic Regression model as a baseline approach to measure the effectiveness of the machine learning-based approach against conventional statistical methods. The test results show that the Random Forest model achieves 99% accuracy, superior to XGBoost (98.9%) and Decision Tree (97%), and much better than the baseline model which only achieves 92% accuracy.

This proves that the machine learning-based approach can provide significant improvements in fraud detection accuracy compared to traditional methods. Furthermore, this study contributes to the development of technology-based fraud detection systems by exploring the use of synthetic data in real-world scenarios. The results obtained show that synthetic data can be used effectively as an alternative when original data is difficult to access due to regulatory limitations.

Different from previous studies, this study not only focuses on model accuracy but also considers practical implementation aspects. The developed model can be integrated into a real-time transaction monitoring system in financial institutions, where suspicious transactions will be automatically flagged and sent to the investigation team for further analysis. This approach not only offers a practical solution for financial institutions in identifying fraud risks, but also opens up opportunities for further development in technology-based fraud detection research. By integrating technological innovation and in-depth understanding of financial transaction patterns, this research is expected to provide real contributions in overcoming the challenges of modern financial crime.

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i4.6420

MATERIALS AND METHODS

The data used for model training is secondary data obtained from simulation results developed in previous research by Lopez-Rojas, Edgar Alonsom Elmir, Ahmad Axelsson, and Stefan. The research produces synthetic data through the PaySim simulator designed to mimic the patterns and characteristics of mobile financial transactions based on real data. This synthetic data was created to overcome the limitations of access to real financial transaction data protected by regulations. Thus, the simulation data is used as an alternative to support the development of a fraud detection model using machine learning.

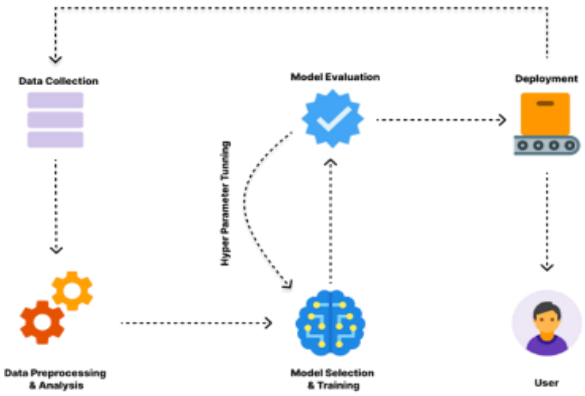
This synthetic dataset is derived from one month of financial records from a mobile financial service used in Africa. The data was provided by a multinational company providing mobile financial services and operating in more than 14 countries worldwide. For the purposes of the analysis, the original dataset was then reduced to a quarter of its original size and created as a synthetic dataset. This dataset includes a total of 6,362,620 mobile financial transaction samples, of which 8,213 were identified as fraudulent transactions, while the remaining 6,354,407 transactions were classified as non-fraudulent transactions [5]. The features contained in the dataset are as follows:

Table 1 PaySim Variable Data

Feature Name	Description
step	Time unit scale
type	Transaction type
amount	Number of transactions
nameOrig	Sender name
oldbalanceOrig	Sender's initial balance
newbalaceOrig	Sender's final balance
nameDest	Recipient's name
oldbalanceDest	Recipient's initial balance
newbalanceDest	Recipient's final balance
isFraud	Fraud and non-fraud categories
isFlaggedFraud	Fraud and non-fraud indication categories

Source: (Research Results, 2025)

A research plan is a structured guide that outlines the steps to be taken to conduct a study. It explains the purpose of the study, the methods used, and the steps required to achieve the desired results. A research plan serves as a roadmap that helps researchers to make the research process systematic, efficient, and focused on the main objectives.



Source: (Research Results, 2025)
Figure 1 Machine Learning Development Process

In general, in Figure 1, the development of a machine learning model will go through data collection to collect data to be used, data preprocessing and analysis to understand or explore data and clean up data problems, conduct model training based on the selected algorithm, evaluate model performance to consider the model's ability to be applied to the system, and deployment into the system or simply in explaining the results of model creation. The following is a research plan that will be carried out in the development of a machine learning model using the CRISP-DM method [11]:

1. Business Understanding

The data used for model training is secondary data obtained from simulation results developed in previous research by Lopez-Rojas, Edgar Alonso, Elmir, Ahmad, Axelsson, and Stefan. The research produces synthetic data through the PaySim simulator, designed to mimic the patterns and characteristics of mobile financial transactions based on real data. This synthetic data was created to overcome the limitations of access to real financial transaction data protected by regulations. Thus, the simulation data is used as an alternative to support the development of a fraud detection model using machine learning.

This synthetic dataset is derived from one month of financial records from a mobile financial service used in Africa. The data was provided by a multinational company offering mobile financial services and operating in more than 14 countries in Africa. For analysis purposes, the original dataset was reduced to a quarter of its original size, maintaining fraud distribution to avoid skewed representations. This dataset includes a total of 6,362,620 mobile financial transaction samples, of which 8,213 were identified as fraudulent transactions, while the remaining 6,354,407 transactions were classified as non-fraudulent transactions.

VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480 /jitk.v10i4.6420

JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

2. Data Preparation

After understanding the data structure, transaction patterns, and general characteristics of fraudulent transactions, the previously identified problems are handled by cleaning the data, handling missing values or outliers, handling imbalanced data, and performing feature engineering to produce relevant features for the model. Data will also be divided into training data and test data.Data Modeling

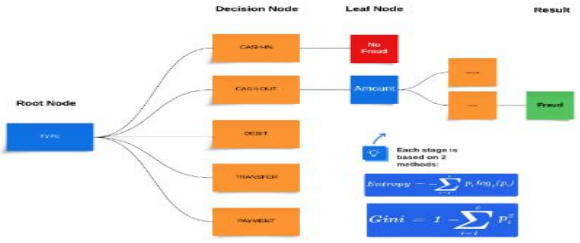
In this study, to handle data imbalance, where fraudulent transactions are only 0.13% of the total dataset, random undersampling technique is used to reduce bias that may occur due to the dominance of the majority class. In addition, feature engineering is carried out by adding new features so that it is expected to increase the accuracy in detecting fraudulent transactions. After the preprocessing stage is complete, the dataset is then divided into training data (80%) and testing data (20%) using stratified sampling technique to ensure that the class distribution is maintained, so that the model can learn with a more balanced data representation and remain accurate in detecting suspicious transactions.

Building several machine learning models and training them on training data. The algorithms used such as Decision Tree, Random Forest, and XGBoost will be compared to choose the best model. The description of how the Decision Tree algorithm works is as follows:

Source: (Research Results, 2025)

Figure 2 Dataset Sharing

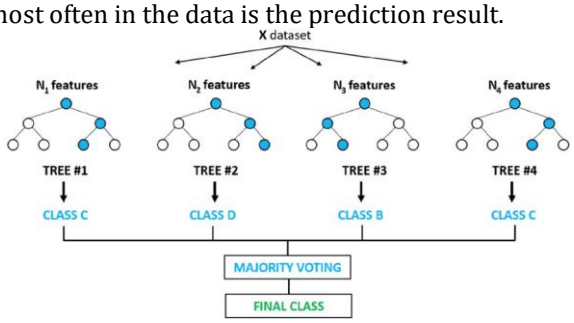
Figure 2 explains how the process begins by importing the entire dataset that will be used to train the model. This dataset contains various attributes or characteristics that are used as input or predictors, as well as labels or targets that are the results to be predicted. Decision Tree will divide this dataset gradually based on the most relevant attributes to form a decision tree structure.



Source: (Research Results, 2025)

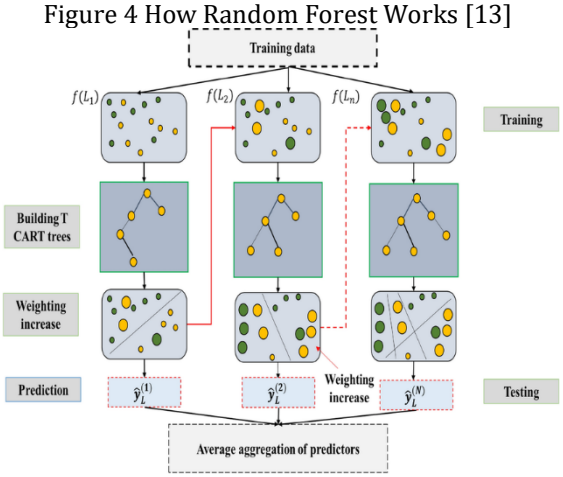
Figure 3 How Decision Trees Work

In figure 3 shows each node in the Decision Tree, the algorithm will select the most relevant features to separate the data into smaller subsets. [12]. The selection of these features is based on certain criteria such as Gini Impurity and Entropy which are used to measure how well the features separate the data. [12]. In the Scikit-Learn library, the decision tree algorithm version used is the CART (Classification and Regression Tree) version, where the Gini Impurity feature selection criteria are used as mathematical calculations in the performance of the decision tree model. The purpose of this process is to maximize clean data separation based on class or target value. After the most relevant features are selected, the data will be divided based on the value of the feature. This process is repeated continuously on each subset of the resulting decision tree until the specified stopping condition has been met. If the condition has been met, the last node in the tree (leaf node) will provide the final result of the prediction or decision. Where the class that appears most often in the data is the prediction result.



Source: (Research Results, 2025)

Figure 4 How Random Forest Works [13]



Source: (Research Results, 2025)

Figure 5 How XGBoost Works [14]

Figures 4 and 5 show how the overall concept of the Random Forest and XGBoost algorithms based on ensemble learning work, namely combining several basic Decision Tree models to maximize prediction results. However, the difference lies in the approach used to achieve the

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i4.6420

final result, if Random Forest uses an approach based on bagging, namely creating many Decision Trees independently, then combining the results using voting to select the majority results of each tree as the final result. [15], Meanwhile, XGBoost uses an approach based on boosting, namely building decision trees sequentially, where each tree tries to improve the prediction error (residual) of the previous tree, this step is repeated several times until the error is minimum. [15].

3. Modeling

This study has applied hyperparameter optimization using Grid Search CV on three machine learning models, namely Decision Tree, Random Forest, and XGBoost, to improve fraud detection performance. Decision Tree was tested with various combinations of splitter, max_depth, min_samples_split, min_samples_leaf, ccp_alpha, and max_features and a wider variation of max_depth. Random Forest was tested with n_estimators, max_depth, min_samples_split, min_samples_leaf, and bootstrap, with additional recommendations to include max_features and expand the variation of the number of decision trees (n_estimators) to improve model stability. XGBoost was optimized with n_estimators, learning_rate, max_depth, gamma, reg_alpha, and reg_lambda, with suggestions to add subsample, colsample_bytree to avoid overfitting. This was done to find the optimal parameter combination with better computational efficiency, as well as to evaluate model performance so that the results obtained are more representative in detecting fraudulent transactions.

4. Data Evaluation

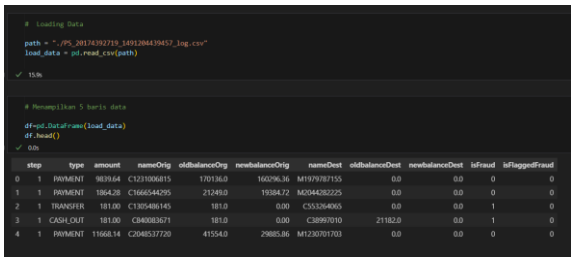
Evaluate the model using test data with specified metrics. To conduct the evaluation, this study uses evaluation metrics such as Accuracy, Precision, Recall, F1-Score, cross-validation and ROC-AUC. If the evaluation results are inadequate, iterations are carried out with parameter adjustments or other techniques. [16][17][18][19].

5. Deployment

Once the best model is selected, the model will be prepared to be implemented in a fraud detection system or in the form of a model performance report. [20].

RESULTS AND DISCUSSION

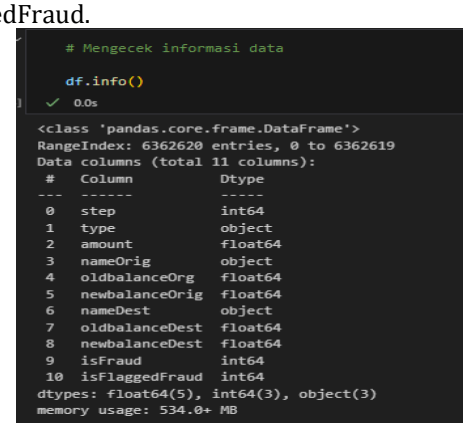
In this section, the results of the machine learning model used for fraud detection are presented in the form of tables, graphs, or other visualizations, accompanied by an interpretation of the effectiveness of the model in identifying suspicious transactions. In addition, the discussion relates the results of the model evaluation to machine learning theory and previous research in the field of fraud detection, to understand the extent to which this approach improves accuracy compared to traditional methods. The analysis also includes factors that affect model performance, such as data imbalance, feature selection, and the possibility of improving the model through optimization techniques or additional data to deal with evolving fraud patterns.



Source: (Research Results, 2025)

Figure 6 Load Dataset

The fraud-related dataset is loaded into the program using the read_csv function from the Pandas library and converted into a Pandas DataFrame to facilitate manipulation and analysis as shown in Figure 6. The initial exploration step is done by displaying the first five rows to understand the structure and type of data. This dataset has 11 columns, namely step, type, amount, nameOrig, oldbalanceOrg, newbalanceOrig, nameDest, oldbalanceDest, newbalanceDest, isFraud, and isFlaggedFraud.



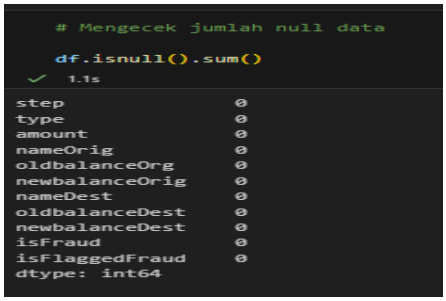
Source: (Research Results, 2025)

Figure 7 Check Data Information

Figure 7 performs a dataset information check using the info() function to find out the number of rows, columns, data types, and memory usage. The dataset consists of 6,362,620 rows with a total memory of 534.0 MB, covering three types of data types: int64 for integer numeric values, object for text data, and float64 for decimal numeric values.

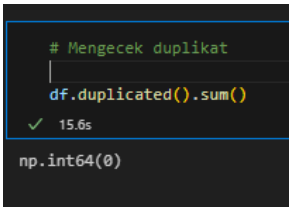
VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480 /jitk.v10i4.6420

JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)



Source: (Research Results, 2025)

Figure 8 Checking Null Values



Source: (Research Results, 2025)

Figure 9 Checking Duplicate Data

Source: (Research Results, 2025)

Figure 10 Rename Columns

The preprocessing stage is carried out to improve data quality before further analysis. This process includes checking and handling empty data, removing duplicates, changing column names to be more descriptive, and standardizing data formats as in figures 8, 9, and 10.

Source: (Research Results, 2025)

Figure 11 Variable Correlation Analysis

This study contributes to the field of financial fraud detection by introducing an innovative feature engineering approach to improve the predictive power of machine learning models. Unlike previous studies that mostly focus on raw transaction data and conventional statistical features—such as transaction amount, balance change, or frequency-based metrics—this study incorporates contextual information about the relationship between the sender and receiver of a transaction by creating a "type2" feature. Figure 11 shows how this feature classifies transactions into four main categories: Customer to Customer (CC), Customer to Merchant (CM), Merchant to Customer (MC), and Merchant to Merchant (MM) based on the interactions between the sender and receiver, which has not been widely explored in previous studies.

Previous studies, such as those by Hajek et al. (2023) and Lokanan (2023), have focused more on traditional financial attributes, such as transaction frequency, transaction velocity, and aggregate account activity. While these features are effective, they often fail to account for the context of the relationships between entities in a transaction, which can be key to detecting fraud patterns in the digital payments ecosystem. This study addresses these limitations by integrating the sender-receiver relationship in the form of additional features, which provide a deeper understanding of suspicious behavioral patterns. Unlike rule-based fraud detection systems that rely on manual heuristics, this study shows that feature engineering can be systematically optimized through machine learning algorithms.

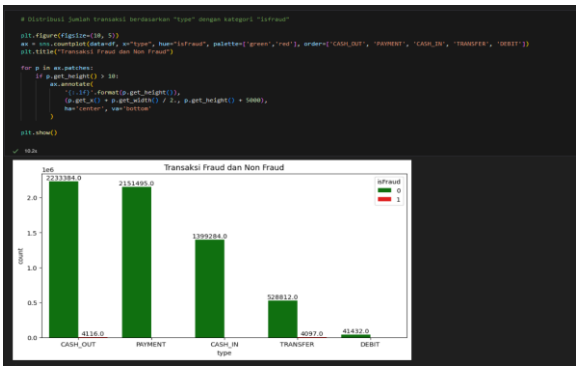


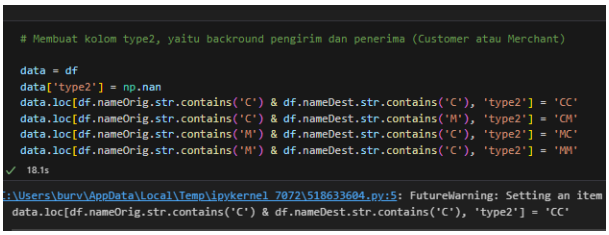
Figure 12 Distribution of Fraud and Non-Fraud Transactions

Figure 12 shows the distribution of transaction types that fraud cases occur more often in cash-out and transfer transactions, while payment, cash-in, and debit transactions have a very low or no number of fraud cases. This finding

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i4.6420

highlights the importance of focusing fraud analysis on higher risk transaction types.



Source: (Research Results, 2025)

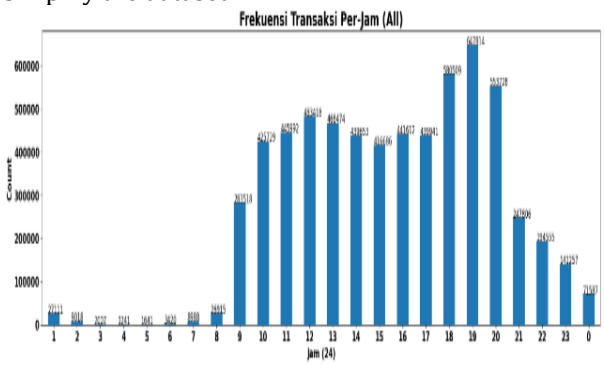
Figure 13 Feature Engineering

Figure 13 shows how the data is also processed to create a new column (feature engineering) named type2, which reflects the background of the sender and recipient of the transaction based on the combination of strings in the nameOrig and nameDest columns. This column is divided into four categories: CC (customer to customer), CM (customer to merchant), MC (merchant to customer), and MM (merchant to merchant), which help in analyzing transaction patterns.

Source: (Research Results, 2025)

Figure 14 Dropping Kolom

Figure 14 shows further data processing involving the removal of less relevant columns, such as nameOrig, nameDest, and isFlaggedFraud, to simplify the dataset.



Source: (Research Results, 2025)

Figure 15 Transaction Frequency in 24 Hours

Figure 15 is the transaction activity analyzed based on hourly frequency, which shows the peak of transactions occurs at 7 pm and the lowest in the early morning. This pattern illustrates the habits of users who are more active in transacting during working hours until early evening. The decrease in activity in the early morning reflects a period with a lower risk of fraud due to minimal transaction volume.

Source: (Research Results, 2025)

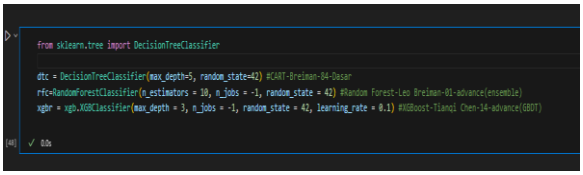
Figure 16 Performing Resampling and Label Encoding

The dataset is then prepared for the machine learning model by randomly sampling the data and encoding the categories into numbers using LabelEncoder as shown in Figure 16. The type and type2 columns are converted into numeric values, resulting in a data representation that is compatible for machine learning algorithms.

Source: (Research Results, 2025)

Figure 17 Splitting Data

The final data is split into training data (70%) and testing data (30%) using train_test_split, with stratification to maintain a proportional label distribution as in figure 17. The use of the random_state parameter ensures that the data split results are consistent across executions, thus supporting the development of reliable models.



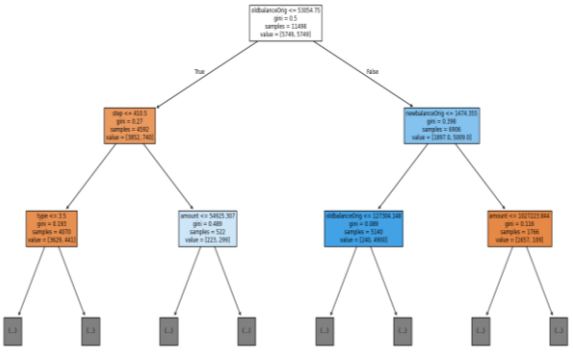
Source: (Research Results, 2025)

Figure 18 Modelling

Figure 18 shows three machine learning models, namely Decision Tree Classifier, Random Forest Classifier, and XGBoost Classifier, initialized with certain configurations to ensure optimal performance. Decision Tree Classifier uses a maximum depth of 5, Random Forest Classifier has 10 classifier trees, and XGBoost Classifier is set to a maximum depth of 3 and a learning rate of 0.1. To maintain consistency of results, all models are initialized with the parameter random_state=42. This configuration allows for structured model testing and evaluation with reproducible results.

VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480 /jitk.v10i4.6420

JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)



Source: (Research Results, 2025)
Figure 19 Decision Tree Model Tree Visualization

The Decision Tree Classifier model is visualized through a graphical representation that shows how features such as oldbalanceOrig, newbalanceOrig, and amount are used to separate the data into different categories as shown in Figure 19. This visualization involves important information at each node, including the Gini value, number of samples, and label distribution. This process provides an overview of how the model utilizes features to make decisions, explaining the internal logic used for classification or prediction.

```
# k-Fold Cross-Validation
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold

kf = KFold(n_splits=5, shuffle=True, random_state=42)  # 5-Fold
cv_scores_dtc = cross_val_score(model_dtc, X, y, cv=kf, scoring='accuracy')
cv_scores_rfc = cross_val_score(model_rfc, X, y, cv=kf, scoring='accuracy')
cv_scores_xgbr = cross_val_score(model_xgbr, X, y, cv=kf, scoring='accuracy')

# Output hasil
print(f"Accuracy per fold dtc: {cv_scores_dtc}")
print(f"Mean accuracy dtc: {cv_scores_dtc.mean():.4f}")
print(f"Accuracy per fold rfc: {cv_scores_rfc}")
print(f"Mean accuracy rfc: {cv_scores_rfc.mean():.4f}")
print(f"Accuracy per fold xgbr: {cv_scores_xgbr}")
print(f"Mean accuracy xgbr: {cv_scores_xgbr.mean():.4f}")
```

✓ 1.0s

```
Accuracy per fold dtc: [0.97534997 0.96468798 0.97108067 0.9695586  0.96803653]
Mean accuracy dtc: 0.9697
Accuracy per fold rfc: [0.98874011 0.98873668 0.9890411  0.98812785 0.99299848]
Mean accuracy rfc: 0.9895
Accuracy per fold xgbr: [0.98874011 0.98630137 0.98995434 0.9890411  0.98843227]
Mean accuracy xgbr: 0.9885
```

Source: (Research Results, 2025)
Figure 20 Cross-Validation

Figure 20 shows how to evaluate the model performance using k-fold cross-validation (5-fold) on the three models. This process divides the data into 5 subsets, using 4 subsets for training and 1 for testing alternately. The evaluation results include the accuracy of each fold as well as the average accuracy for each model, providing a comprehensive picture of the model's reliability. This approach helps ensure that the model is able to handle variations in the data well and is not overfitting.

```
accuracy_clf = accuracy_score(y_test, y_pred_dtc)
precision_clf = precision_score(y_test, y_pred_dtc)
recall_clf = recall_score(y_test, y_pred_dtc)
f1_score_clf = f1_score(y_test, y_pred_dtc)
auc_clf = roc_auc_score(y_test, y_pred_dtc)

cm = confusion_matrix(y_test, y_pred_dtc)
true_negative = cm[0, 0]
false_positive = cm[0, 1]
specificity_clf = true_negative / (true_negative + false_positive)

print(f'Accuracy Decision Tree: {accuracy_clf}')
print(f'Precision Decision Tree: {precision_clf}')
print(f'Recall Decision Tree: {recall_clf}')
print(f'Specificity Decision Tree: {specificity_clf}')
print(f'F1 Score Decision Tree: {f1_score_clf}')
print(f'AUC Decision Tree: {auc_clf}')
```

```
Accuracy Decision Tree: 0.9699675324675324
Precision Decision Tree: 0.9841137123745819
Recall Decision Tree: 0.9553571428571429
Specificity Decision Tree: 0.9845779220779922
F1 Score Decision Tree: 0.9695224052718828
AUC Decision Tree: 0.9699675324675323
```

Source: (Research Results, 2025)
Figure 21 Evaluation of Decision Tree Model

Figure 21 shows the performance of the Decision Tree Classifier analyzed using evaluation metrics such as accuracy, precision, recall, specificity, F1 score, cross-validation and AUC (Area Under Curve). The evaluation results show that the model has very good performance, with accuracy and AUC values of around 0.97, precision 0.98, recall 0.95, specificity 0.98, and F1 score 0.97. These numbers reflect a good balance between positive and negative predictions, while also demonstrating the model's ability to effectively distinguish classes.

```
accuracy_rfc = accuracy_score(y_test, y_pred_rfc)
precision_rfc = precision_score(y_test, y_pred_rfc)
recall_rfc = recall_score(y_test, y_pred_rfc)
f1_score_rfc = f1_score(y_test, y_pred_rfc)
auc_rfc = roc_auc_score(y_test, y_pred_rfc)

cm = confusion_matrix(y_test, y_pred_rfc)
true_negative = cm[0, 0]
false_positive = cm[0, 1]
specificity_rfc = true_negative / (true_negative + false_positive)

print(f'Accuracy Random Forest: {accuracy_rfc}')
print(f'Precision Random Forest: {precision_rfc}')
print(f'Recall Random Forest: {recall_rfc}')
print(f'Specificity Random Forest: {specificity_rfc}')
print(f'F1 Score Random Forest: {f1_score_rfc}')
print(f'AUC Random Forest: {auc_rfc}')
```

✓ 0.0s

```
Accuracy Random Forest: 0.9906655844155844
Precision Random Forest: 0.9867149758454107
Recall Random Forest: 0.994724025974026
Specificity Random Forest: 0.9866071428571429
F1 Score Random Forest: 0.9907033144704931
AUC Random Forest: 0.9906655844155845
```

Source: (Research Results, 2025)
Figure 22 Random Forest Model Evaluation

```
accuracy_xgbr = accuracy_score(y_test, y_pred_xgbr)
precision_xgbr = precision_score(y_test, y_pred_xgbr)
recall_xgbr = recall_score(y_test, y_pred_xgbr)
f1_score_xgbr = f1_score(y_test, y_pred_xgbr)
auc_xgbr = roc_auc_score(y_test, y_pred_xgbr)

cm = confusion_matrix(y_test, y_pred_xgbr)
true_negative = cm[0, 0]
false_positive = cm[0, 1]
specificity_xgbr = true_negative / (true_negative + false_positive)

print(f'Accuracy Random Forest: {accuracy_xgbr}')
print(f'Precision Random Forest: {precision_xgbr}')
print(f'Recall Random Forest: {recall_xgbr}')
print(f'Specificity Random Forest: {specificity_xgbr}')
print(f'F1 Score Random Forest: {f1_score_xgbr}')
print(f'AUC Random Forest: {auc_xgbr}')
```

✓ 0.0s

```
Accuracy Random Forest: 0.9890422077922078
Precision Random Forest: 0.9816147082334132
Recall Random Forest: 0.996753246753247
Specificity Random Forest: 0.9813311688311688
F1 Score Random Forest: 0.9891260571888845
AUC Random Forest: 0.9890422077922078
```

Source: (Research Results, 2025)
Figure 23 XGBoost Model Evaluation

The Random Forest Classifier and XGBoost Classifier models were also evaluated using the same metrics, with the results showing higher performance compared to the Decision Tree Classifier as shown in Figures 22 and 23. The

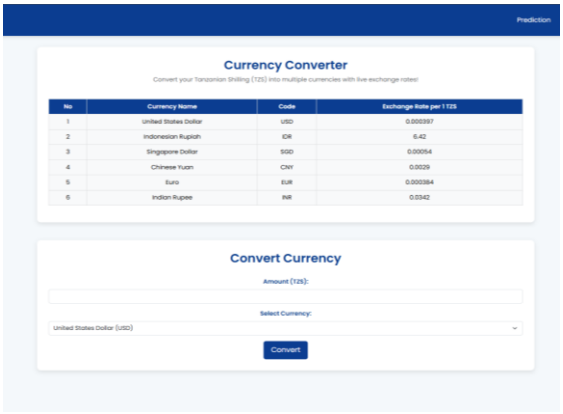
VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i4.6420

Random Forest Classifier achieved accuracy and AUC values of around 0.99, precision 0.98, recall 0.99, specificity 0.98, and F1 score 0.99. Meanwhile, the XGBoost Classifier has accuracy and AUC values of around 0.989, precision 0.981, recall 0.997, specificity 0.981, and F1 score 0.989. Both models show excellent ability in predicting accurately and efficiently, making them a very reliable solution for detecting fraud cases.

Source: (Research Results, 2025)
Figure 24 Prediction Form and Batch Prediction Form pages

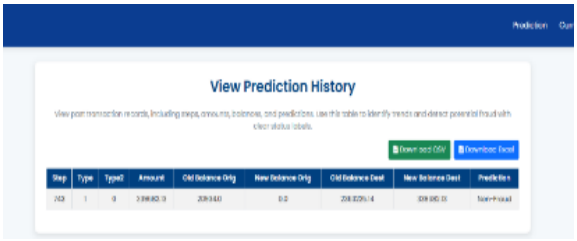
The Transaction Prediction Form in Figure 24 allows users to enter transaction details that include several important elements. Step (Time step) refers to the step or stage in time of the transaction. Transaction Type indicates the type of transaction being made, while Sub-Transaction Type describes a more specific subtype of transaction. Transaction Amount refers to the amount of the transaction being made. Next, details related to balances include Sender's Old Balance, which is the sender's initial balance before the transaction, and Sender's New Balance, which is the sender's final balance after the transaction. In addition, there is Recipient's Old Balance, which describes the recipient's initial balance before the transaction, and Recipient's New Balance, which is the recipient's final balance after the transaction is completed.

In addition, the Transaction Batch Prediction Form allows users to upload a CSV file containing data from multiple transactions to perform batch predictions. The file must follow the CSV format structure as in the Transaction Prediction Form. The prediction process can be done by pressing the Batch Predict button.



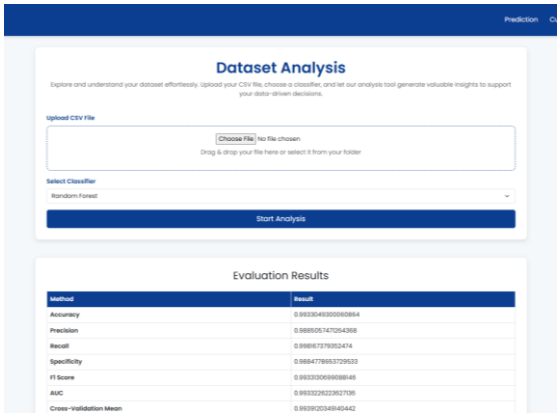
Source: (Research Results, 2025)
Figure 25 Currency Exchange page

Figure 25 shows the currency conversion page with two main features. The first section is a table that displays the exchange rates of Tanzanian Shilling (TZS) to several other currencies such as United States Dollar (USD), Indonesian Rupiah (IDR), Singapore Dollar (SGD), Chinese Yuan (CNY), Euro (EUR), and Indian Rupee (INR). The second section is a form to calculate the conversion rate of TZS to a specific currency based on the amount entered by the user. The second section is a form to calculate the currency conversion rate that allows users to calculate the conversion rate from TZS to a specific currency. Users simply enter the amount of TZS they want to convert and select the destination currency, then the system will immediately display the results based on the latest exchange rates.



Source: (Research Results, 2025)
Figure 26 History page

Figure 26 displays the transaction prediction history. The table contains information such as step, transaction type, sub-transaction type (type2), amount, sender's initial and final balance (old balance orig, new balance orig), recipient's initial and final balance (old balance dest, new balance dest), and the prediction result whether it is fraud or non-fraud. There is an option to download this history in CSV or Excel format.



Source: (Research Results, 2025)

Figure 27 Training Page

Figure 27 is used for dataset analysis, where users can upload a CSV file and select a classification model such as Random Forest or XGBoost. The model evaluation results, including metrics such as accuracy, precision, recall, and others are displayed in a table, along with a preview of the dataset and visualization of the confusion matrix and distribution of fraud and non-fraud data.

CONCLUSION

This study successfully developed a machine learning-based fraud detection model using the synthetic PaySim dataset that mimics real financial transaction patterns. By applying the Decision Tree, Random Forest, and XGBoost algorithms, this study shows that the developed model is able to detect suspicious transactions with a high level of accuracy, where Random Forest achieves the highest accuracy of 99%, followed by the XGBost algorithm with an accuracy value of 98.9%, and Decision Tree with an accuracy of 97%. The analysis shows that cash-out and transfer transactions have the highest risk of fraud, making them the main focus in financial risk mitigation. In addition, this study highlights the importance of feature engineering in improving model accuracy and the use of Random Undersampling techniques to overcome class imbalance in data. Although this study is still in the prototype stage with synthetic data, the results provide valuable insights into the application of fraud detection models in the real world. This model has the potential to be integrated into real-time bank transaction monitoring systems or in batch analysis, and can be used as an additional layer in existing rule-based detection systems. The implementation of this model is expected to help financial institutions reduce the risk of fraud, increase compliance with Anti-Money Laundering (AML) regulations, and encourage innovation in technology-based fraud detection systems.

REFERENCE

[1] Basel, "Basel AML Index 2023 : 12th Public Edition Ranking money laundering and terrorist financing risks around the world," *Basel*, vol. 10, pp. 1–42, 2023, [Online]. Available: https://index.baselgovernance.org/api/uploads/Basel_AML_Index_2023_12th_Edition_879b07b7b2.pdf

[2] P. M. A. C. Rakesh Pandit, Sheetal Bawane, Jayesh Surana, "Credit Risk Assessment and Fraud Detection in Financial Transactions Using Machine Learning," *J. Electr. Syst.*, vol. 20, no. 3s, pp. 2061–2069, 2024, doi: 10.52783/jes.1807.

[3] N. Naveed, S. Munawar, and A. Usman, "Intelligent Anti-Money Laundering Fraud Control Using Graph-Based Machine Learning Model for the Financial Domain," *J. Cases Inf. Technol.*, vol. 25, no. 1, pp. 1–21, 2023, doi: 10.4018/JCIT.316665.

[4] M. S. Gal and O. Lynskey, "Synthetic Data: Legal Implications of the Data-Generation Revolution," *Iowa Law Rev.*, vol. 109, no. 3, pp. 1087–1156, 2024, doi: 10.2139/ssrn.4414385.

[5] T. Wiehn, "Synthetic Data: From Data Scarcity to Data Pollution," *Surveill. Soc.* , vol. 22, no. 4, pp. 472–476, 2024, doi: 10.24908/ss.v22i4.18327.

[6] M. E. Lokanan, "Predicting mobile money transaction fraud using machine learning algorithms," *Appl. AI Lett.*, vol. 4, no. 2, pp. 1–16, 2023, doi: 10.1002/ail2.85.

[7] S. M. N. Nobel *et al.*, "Unmasking Banking Fraud: Unleashing the Power of Machine Learning and Explainable AI (XAI) on Imbalanced Data," *Inf.*, vol. 15, no. 6, pp. 1–23, 2024, doi: 10.3390/info15060298.

[8] O. I. Alex, "Development and Implementation of a Machine Learning-Based Framework for Credit Card Fraud Detection : A Comparative Study of Random Forest and Logistic Regression Models," vol. 9, no. 3, pp. 52–66, 2025.

[9] P. Hajek, M. Z. Abedin, and U. Sivarajah, "Fraud Detection in Mobile Payment Systems using an XGBoost-based Framework," *Inf. Syst. Front.*, vol. 25, no. 5, pp. 1985–2003, 2023, doi: 10.1007/s10796-022-10346-6.

[10] A. Alwadain, R. F. Ali, and A. Muneer, "Estimating Financial Fraud through Transaction-Level Features and Machine Learning," *Mathematics*, vol. 11, no. 5, 2023, doi: 10.3390/math11051184.

VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i4.6420

[11] M. Elkabalawy, A. Al-Sakkaf, E. Mohammed Abdelkader, and G. Alfalah, "CRISP-DM-Based Data-Driven Approach for Building Energy Prediction Utilizing Indoor and Environmental Factors," *Sustainability*, vol. 16, no. 17, p. 7249, 2024, doi: 10.3390/su16177249.

[12] A. Ostonov and M. Moshkov, "Comparative Analysis of Deterministic and Nondeterministic Decision Trees for Decision Tables from Closed Classes," *Entropy*, vol. 26, no. 6, 2024, doi: 10.3390/e26060519.

[13] A. Abdulla, G. Baryannis, and I. Badi, "An Integrated Machine Learning and MARCOS Method for Supplier Evaluation and Selection," *Decis. Anal. J.*, vol. 9, Oct. 2023, doi: 10.1016/j.dajour.2023.100342.

[14] Z. Ali and A. Burhan, "Hybrid machine learning approach for construction cost estimation: an evaluation of extreme gradient boosting model," *Asian J. Civ. Eng.*, vol. 24, pp. 1–16, Apr. 2023, doi: 10.1007/s42107-023-00651-z.

[15] J. H. Chen, X. L. Wang, and F. Lei, "Data-driven multinomial random forest: a new random forest variant with strong consistency," *J. Big Data*, vol. 11, no. 1, 2024, doi: 10.1186/s40537-023-00874-6.

[16] P. Handayani and A. Charis Fauzan, "KLIK: Kajian Ilmiah Informatika dan Komputer Machine Learning Klasifikasi Status Gizi Balita Menggunakan Algoritma Random Forest," *Media Online)*, vol. 4, no. 6, pp. 3064–3072, 2024, doi: 10.30865/klik.v4i6.1909.

[17] M. E. Hasan and A. Wagler, "New Convolutional Neural Network and Graph Convolutional Network-Based Architecture for AI Applications in Alzheimer's Disease and Dementia-Stage Classification," *AI*, vol. 5, no. 1, pp. 342–363, 2024, doi: 10.3390/ai5010017.

[18] J. de Pedro-Carracedo, J. Clemente, D. Fuentes-Jimenez, M. F. Cabrera-Umpiérrez, and A. P. Gonzalez-Marcos, "Photoplethysmographic Signal-Diffusive Dynamics as a Mental-Stress Physiological Indicator Using Convolutional Neural Networks," *Appl. Sci.*, vol. 13, no. 15, 2023, doi: 10.3390/app13158902.

[19] S. M. Mbiva and F. M. Correa, "Machine Learning to Enhance the Detection of Terrorist Financing and Suspicious Transactions in Migrant Remittances," *J. Risk Financ. Manag.*, vol. 17, no. 5, 2024, doi: 10.3390/jrfm17050181.

[20] V. Plotnikova, M. Dumas, and F. P. Milani, "Applying the CRISP-DM data mining process in the financial services industry: Elicitation of adaptation requirements," *Data Knowl. Eng.*, vol. 139, p. 102013, 2022, doi: https://doi.org/10.1016/j.datak.2022.102013.