Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th October 2025. Vol. 5. No. 1; e-ISSN: 2808-4519

Implementation of the Random Forest Algorithm for Financial Transaction Fraud Detection

David[1*], Robet[2], Hendri[3]

[1,2,3]*Informatics Engineering, STMIK Time, Medan, Indonesia*
zxtzhuang@gmail.com[1*], robertdetime@gmail.com[2], h4ndr7@hotmail.com[3]

Abstract

Fraud detection in financial transactions is a major challenge for the banking and fintech industries, especially with the increasing volume of digital transactions. This study aims to implement the Random Forest algorithm in machine learning to detect suspicious financial transactions. The Random Forest algorithm was chosen due to its ability to handle complex data and produce accurate predictions. This research uses a financial transaction dataset consisting of various features such as transaction amount, location, payment method, and user activity patterns. The data undergoes preprocessing stages, including handling missing values, normalization, and oversampling techniques to address data imbalance. The Random Forest model is then developed and evaluated using accuracy, precision, recall, and F1-Score metrics to assess its fraud detection performance. The results show that the Random Forest model performs well in detecting fraudulent transactions with a high level of accuracy. Analysis of the confusion matrix also indicates that the model is able to reduce the number of false negatives, which is a crucial aspect in preventing losses due to illegal transactions. Additionally, a feature correlation heatmap is used to identify the most influential variables in fraud prediction. With the implementation of this Random Forest-based system, it is expected that the financial industry can enhance early detection of suspicious activities and strengthen security in digital transactions.

Keywords: *Fraud Detection, Random Forest, Machine Learning, Financial Transactions, Digital Security*

1. Introduction

In this increasingly advanced digital era, electronic financial transactions have become an integral part of everyday life. Cashless payment methods, such as credit cards and online transactions, provide convenience and speed that encourage society to shift from traditional to digital transactions. According to [1], consumer behavior in the use of banking credit cards shows an increasing trend in line with technological developments. However, this growth is also accompanied by new challenges, particularly in terms of transaction security. Fraud cases in the financial sector, especially those related to credit card usage, are becoming more frequent. According to [2], credit card fraud has caused financial losses amounting to billions of rupiah for customers. This condition drives the need for a reliable fraud detection system to protect both consumers and financial institutions from economic losses.

One of the common approaches used to identify fraud is through the application of Machine Learning techniques. Machine Learning technology has become an important research topic in improving the security and reliability of electronic payment systems, especially for detecting fraud in online financial transactions. This research aims to evaluate the effectiveness of Machine Learning algorithms in identifying fraudulent activities in financial transactions, which includes collecting datasets covering various types of online financial transactions, such as user data, transaction amounts, transaction types, and a number of other relevant features. The dataset is then processed and cleaned to ensure optimal data quality before being analyzed using Machine Learning techniques [3]. Ensemble learning algorithms, such as Random Forest, are known to be highly effective in detecting anomalies in financial transactions. Research conducted by [2] revealed that ensemble learning methods, particularly Random Forest, can significantly improve fraud detection performance compared to simpler models. In addition, a study conducted by [4] proved that the Random Forest method can classify fraud in credit card transactions with a high level of accuracy.

However, the implementation of the Random Forest algorithm also faces several challenges, especially in dealing with data imbalance between normal transactions and those classified as fraud. According to [3], the use of oversampling techniques is an important step to overcome this obstacle and improve detection accuracy. Furthermore, computational efficiency becomes a crucial factor that must be considered in designing a fraud detection system that can operate in real-time.Based on this background, this research aims to implement the Random Forest algorithm in detecting fraud in financial transactions, utilizing existing Machine Learning frameworks such as Scikit-learn. The focus of this study is on the implementation and performance analysis of the Random Forest method, in order to contribute to efforts in preventing financial transaction fraud in Indonesia.

2. Literature Study

Machine Learning (ML) is a branch of Artificial Intelligence (AI) that enables systems to learn from data without being explicitly programmed. With Machine Learning, computer systems can continuously adapt and improve their performance as they gain more "experience" [5]. Thus, the performance of such systems can be enhanced by providing larger and more diverse datasets for processing. In the context of financial transaction fraud detection, ML is used to build predictive models based on patterns found in historical transaction data [6]. ML algorithms can be classified into three main categories: supervised learning, unsupervised learning, and reinforcement learning. The Random Forest method used in this study falls under supervised learning [7].Random Forest is an ensemble learning algorithm based on decision trees. This algorithm works by constructing a collection of decision trees during the training process, then combining the results of each tree to improve prediction accuracy and reduce overfitting. Random Forest is highly suitable for both classification and regression problems [10].

Fraud detection is the process of identifying suspicious or unusual activities in financial transactions. It involves recognizing unauthorized or irregular transaction activities. Fraud can occur in various forms, such as unauthorized credit card usage, embezzlement, and identity theft. Fraud detection systems are designed to protect financial institutions and customers from financial losses. In developing such systems, transaction data is analyzed to detect unusual patterns [15].Although Random Forest offers many advantages in detecting financial transaction fraud, there are also several limitations and challenges in its application. These challenges include computational aspects, interpretability, and data imbalance, which may affect the model's effectiveness under certain conditions [16].

3. Method

The development of a fraud detection system is carried out through several main stages, which include:

1. Data Collection: obtaining a financial transaction dataset containing both legitimate and fraudulent transactions.
2. Data Preprocessing: cleaning the data, handling missing values, and performing normalization or feature transformation.
3. Model Building: using the Random Forest algorithm to build the fraud detection model.
4. Model Training: training the model using historical data to identify suspicious transaction patterns.
5. Testing and Evaluation: using test data to measure the model's performance with evaluation metrics.
6. System Implementation: integrating the model into a web-based system for real-world application.

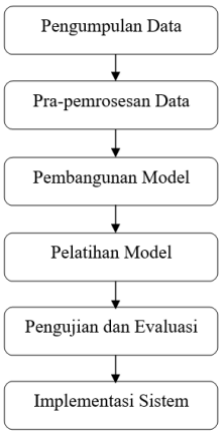


Fig. 1: Fraud Detection System Development Flow Diagram

The dataset used in this research is obtained from relevant sources, such as the Credit Card Fraud Detection dataset from Kaggle. This dataset contains credit card transactions with labels indicating whether a transaction is fraudulent (fraud) or not (non-fraud). Each transaction in the dataset includes several features, such as:

1. Amount: the transaction amount.
2. Time: the transaction time relative to the first transaction.
3. V1–V28: features derived from dimensionality reduction using Principal Component Analysis (PCA).
4. Class: the target label (1 for fraud, 0 for non-fraud).

Below is an example of the dataset structure used in this research:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	0.090794	-0.551600	-0.617801	-0.991390	-0.311169
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	-0.166974	1.612727	1.065235	0.489095	-0.143772
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	0.207643	0.624501	0.066084	0.717293	-0.165946
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	-0.054952	-0.226487	0.178228	0.507757	-0.287924
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	0.753074	-0.822843	0.538196	1.345852	-1.119670

Fig. 2: Example of Credit Card Transaction Dataset Structure

The dataset used in this research consists of 284,807 transactions with 31 features, including Time, Amount, and the PCA-transformed features (V1–V28). It can be observed that the average transaction amount (Amount) is 88.35, with a maximum value reaching 25,691.16. Furthermore, the Class value shows that only 0.17% of the total transactions are classified as fraud, which indicates a significant data imbalance (imbalanced dataset). Therefore, data balancing techniques must be applied before the model training process.

```
(284807, 31)
                Time            V1  ...         Amount         Class
count  284807.000000  2.848070e+05  ...  284807.000000  284807.000000
mean    94813.859575  3.919560e-15  ...      88.349619       0.001727
std     47488.145955  1.958696e+00  ...     250.120109       0.041527
min         0.000000 -5.640751e+01  ...       0.000000       0.000000
25%     54201.500000 -9.203734e-01  ...       5.600000       0.000000
50%     84692.000000  1.810880e-02  ...      22.000000       0.000000
75%    139320.500000  1.315642e+00  ...      77.165000       0.000000
max    172792.000000  2.454930e+00  ...   25691.160000       1.000000

[8 rows x 31 columns]
```

Fig. 3: Descriptive Statistics of the Dataset

This imbalance may cause the model to ignore fraudulent transactions due to their small proportion compared to legitimate transactions. Hence, in the next stage, techniques such as resampling (oversampling or undersampling), synthetic data generation (SMOTE), or the use of algorithms that are more robust to imbalanced data are required.

```
0.0017304750013189597
Fraud Cases: 492
Valid Transactions: 284315
```

Fig. 4: Class Distribution in the Dataset

Based on Figure 3.5, the following explanation can be made:
1. 492 fraudulent transactions (label 1).
2. 284,315 legitimate transactions (label 0).
3. The ratio of fraudulent transactions is only 0.17% of the total data, indicating that the dataset is highly imbalanced.

This imbalance may lead the Machine Learning model to predict all transactions as valid, due to the dominance of the majority class. Therefore, specific techniques such as oversampling, undersampling, or other methods must be applied to handle this imbalance before model training is conducted.

```
Amount details of the fraudulent transaction
count     492.000000
mean      122.211321
std       256.683288
min         0.000000
25%         1.000000
50%         9.250000
75%       105.890000
max      2125.870000
Name: Amount, dtype: float64
```

Fig. 5: Descriptive Statistics of Fraudulent Transaction Amounts

Based on Figure 3.6, the following analysis of fraudulent transaction statistics can be made:
1. Average fraud transaction amount: 122.21
2. Standard deviation: 256.68 (indicating large variation in fraud amounts)
3. Minimum value: 0.00 (some fraudulent transactions have very small amounts)
4. 25th percentile: 1.00 (25% of fraudulent transactions are less than or equal to 1.00)
5. 50th percentile (median): 9.25 (half of fraudulent transactions are below 9.25)
6. 75th percentile: 105.89 (75% of fraudulent transactions are below this value)
7. Maximum value: 2,125.87 (some fraudulent transactions involve very large amounts)

Conclusion:
1. Most fraudulent transactions involve small amounts (median 9.25), but there are some with very large amounts (up to 2,125.87).
2. Further analysis can be conducted to observe patterns of high-value fraudulent transactions.
3. Histogram or boxplot visualizations can be used to better understand data distribution.

Below is the source code used for data collection (Data Collecting):

```
data = pd.read_csv("creditcard.csv")  # Ganti dengan lokasi dataset
print(data.head())  # Menampilkan 5 baris pertama dataset
```

Fig 6: Source Code for Data Collecting

4. Result

In this stage, the system results will be discussed, covering the implementation and application of the analysis and system design presented in the previous chapter, as well as the devices required to run the application.

1. Login Screen
This is the first screen displayed when the user opens the application, as shown in Figure 7.

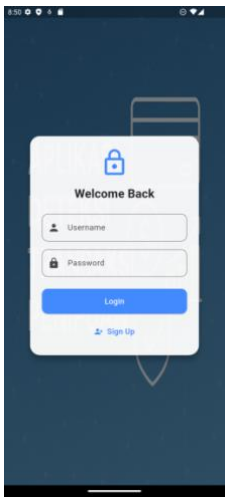


Fig. 7: Login Screen

The login screen contains two input elements and two main buttons:
a. Input Elements:
Username Input
Used to enter the registered username. The data entered is validated with the database to verify the user's identity.
Password Input
Used to enter the user's password. The password is validated to ensure secure access.
b. Buttons:
Login Button
Performs verification of the entered username and password. If valid, the user is directed to the main page of the application.
Sign Up Button
Directs the user to the account registration page. Suitable for users who do not yet have an account.

2. Sign Up Screen
This screen appears when the user selects the Sign Up button on the login screen. It allows new users to register for access to the system, as shown in Figure 8.

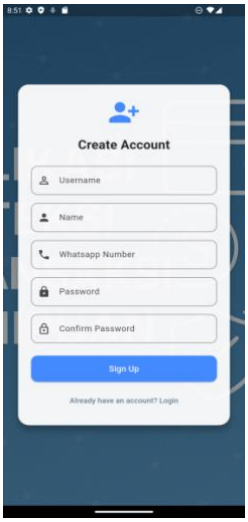


Fig. 8: Sign Up Screen

The sign up screen contains five input fields and two buttons:
a. Input Elements:
Username Input – to set a unique username for login.
Name Input – to enter the user's full name.
Phone Number Input – to enter an active phone number for verification or communication.
Password Input – to set a secure account password.
Confirm Password Input – to confirm the entered password. The system checks for consistency.
b. Buttons:
Sign Up Button – submits the registration data. The system validates before saving to the database.
Have an Account? Login Button – redirects the user back to the login page if they already have an account.

3. Admin Home Screen
After successfully logging in as an admin, the user is directed to the Admin Home page, which displays a summary of key transaction data analyzed by the system, as shown in Figure 9.

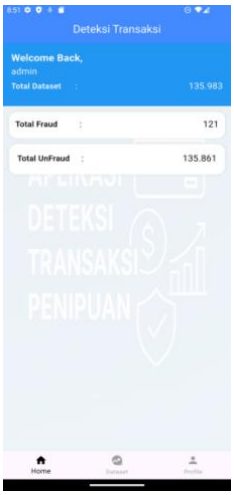


Fig. 9: Admin Home Screen

a. Displayed Information:
Total Dataset – total number of transactions processed.
Total Fraud – number of transactions identified as fraud by the Random Forest algorithm.
Total Unfraud – number of transactions classified as normal.
b. Bottom Navigation Bar:
Home – displays the summary screen.
Dataset – navigates to transaction dataset management.
Profile – displays admin account info and provides account settings or logout.

4. Dataset Screen
This page is used to display and manage financial transaction datasets for fraud detection (Figure 10).

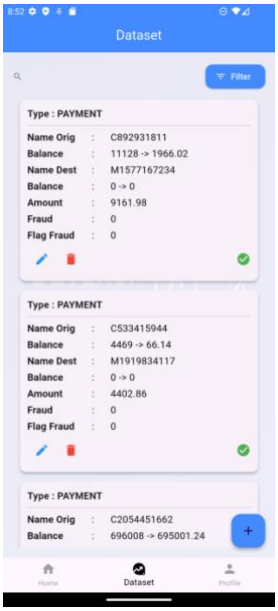


Fig. 10: Dataset Screen

Each dataset item shows attributes such as:
Type (e.g., TRANSFER, CASH_OUT)
Name Orig (origin account)
Balance Orig (origin account balance before transaction)
Name Dest (destination account)
Balance Dest (destination account balance after transaction)
Amount (transaction amount)
Fraud (1 = fraud, 0 = non-fraud)
Flag Fraud (fraud suspicion indicator before analysis)
Buttons:
Filter – search data by keyword.
Add – add a new transaction entry.
Edit – modify transaction data.

Delete – permanently remove a transaction entry.

5. Filter Dialog Display
This display is a pop-up dialog that appears when the user presses the Filter button on the dataset page.

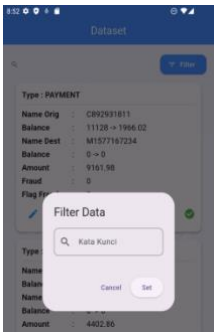


Fig.11: Filter Dialog Display

Its function is to simplify the search for data based on specific keywords, such as source account name, destination account name, or transaction type.

a. Input Element:
Keyword Input: Used to enter the search keyword. The system will filter and display the dataset containing the keyword in any of the data columns.

b. Buttons:
Set Button
Functions to apply the filter based on the entered keyword. Once pressed, the dataset display will be updated according to the search results.
Cancel Button
Used to close the filter dialog without applying any search, and return to the original dataset display.

6. Add Dataset Page
This page is used by the admin to add new transaction data into the system. The added data will become part of the dataset used in the fraud detection process with the Random Forest algorithm.

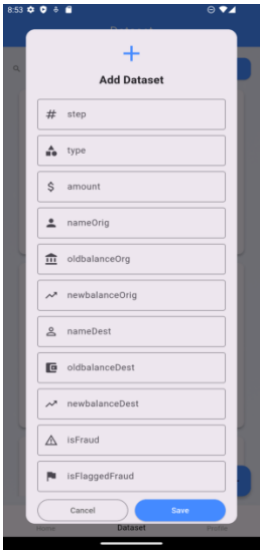


Fig. 12: Add Dataset Page

The input form consists of several transaction data fields, namely:

a. Step
Input for the sequence step or simulation time when the transaction took place.

b. Type
The type of transaction, such as TRANSFER, CASH_OUT, and others.

c. Amount
The nominal amount of money being transacted.

d. Name Orig
The source account name or sender of the funds.

e. Old Balance Orig
The initial balance of the source account before the transaction.

f. New Balance Orig
The balance of the source account after the transaction.

g. Name Dest
The destination account name or receiver of the funds.
h. Old Balance Dest
The initial balance of the destination account before receiving the funds.
i. New Balance Dest
The balance of the destination account after receiving the transaction.
j. Is Fraud
A binary value (0 or 1) indicating whether the transaction is fraudulent (1) or not (0).
k. Is Flagged Fraud

An additional binary value to mark whether the transaction is suspected of fraud prior to verification (1 for suspected, 0 for not).
Buttons:
a. Save Button
Used to save the new transaction data into the database once all fields are completed and validation succeeds.
b. Cancel Button
Used to cancel the dataset addition process and return to the previous page without saving changes.

7. Edit Dataset Page
This page is used by the admin to modify or update existing transaction information within the dataset.

Fig. 13: Edit Dataset Page

This page contains input elements similar to the Add Dataset Page, but with an additional special field, ID, which serves as the unique identifier of the transaction being edited.
Input Elements:
ID – A unique identifier for each transaction in the dataset. This field is read-only and cannot be changed, used to specify which data will be updated.
Step – The sequence step or simulation time of the transaction.
Type – The type of transaction, such as TRANSFER, CASH_OUT, etc.
Amount – The amount of money transacted.
Name Orig – The sender's account name.
Old Balance Orig – The initial balance of the source account before the transaction.
New Balance Orig – The final balance of the source account after the transaction.
Name Dest – The recipient's account name.
Old Balance Dest – The initial balance of the destination account before the transaction.
New Balance Dest – The final balance of the destination account after the transaction.
Is Fraud – Status indicating whether the transaction is fraudulent (1) or not (0).
Is Flagged Fraud – Indicator whether the transaction is suspected of fraud (1) or not (0) before further analysis.
Buttons:
Update Button – Saves the changes made to the existing transaction data.
Cancel Button – Cancels the editing process and returns to the previous page without saving changes.

8. Delete Dialog
This dialog appears when the admin presses the Delete button on one of the dataset entries. Its function is to confirm whether the data should indeed be permanently deleted from the system. This confirmation is important to prevent accidental deletions.

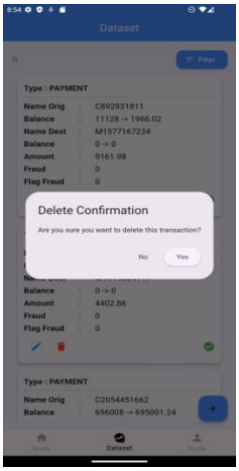
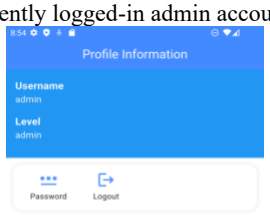


Fig.14: Delete Dialog

Dialog Components:
Confirmation Message
Displays a question such as:
"Are you sure you want to delete this transaction?"
Buttons:
a. Yes Button – Permanently deletes the transaction from the dataset.
b. No Button – Closes the dialog without deleting anything.

9. Admin Profile Page
This page is used to display information about the currently logged-in admin account.



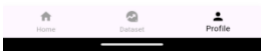


Fig.15: Admin Profile Page

The admin can view basic account details and perform security actions such as changing the password or logging out of the application.
Displayed Information:
a. Username – Displays the currently logged-in user's name.
b. Level – Displays the role or access rights of the user, e.g., admin.
Buttons:
a. Password Button – Opens the password change page or dialog. The admin can update the old password with a new one for account security.
b. Logout Button – Logs out of the session and returns to the login page.

10. User Home Page
The User Home page is visually and functionally similar to the Admin Home page but intended for non-admin users. It presents a summary of fraud detection results accessible to general users.

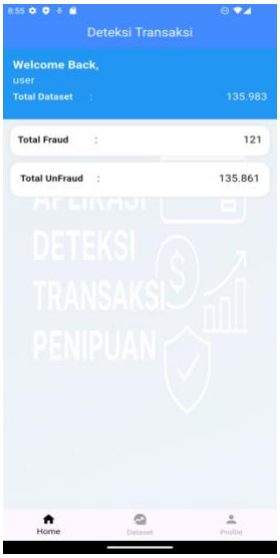


Fig. 16: User Home Page

Displayed Information:
a. Total Dataset – Displays the total number of available transaction records.
b. Total Fraud – Displays the number of transactions detected as fraudulent.
c. Total Unfraud – Displays the number of transactions detected as non-fraudulent.

Bottom Navigation:
a. Home – To display this summary page.
b. Dataset – To access the list of transaction datasets.
c. Profile – To access the user account information page.

11. Dataset Page
This page is used to display a list of financial transaction datasets utilized in the fraud detection process.

Fig. 17: Dataset Page

Each item in the dataset list displays several important transaction attributes, namely:
a. Type – The type of transaction, e.g., TRANSFER, CASH_OUT, etc.
b. Name Orig – The source account name or identifier.
c. Balance Orig – The balance of the source account before the transaction.
d. Name Dest – The destination account name or identifier.
e. Balance Dest – The balance of the destination account after the transaction.
f. Amount – The amount of money transacted.
g. Fraud – Status indicating whether the transaction is identified as fraudulent (1) or not (0).
h. Flag Fraud – An additional indicator showing whether the transaction is suspected of being fraudulent.
Buttons:
a. Filter – Used to search for data based on specific keywords such as account names or transaction types, making it easier to find specific records within large datasets.
b. Process – Used to run fraud detection on transactions.

12. Detection Process Page

This page allows users to directly detect fraud based on input transaction data. The system processes the data using the Random Forest algorithm to predict whether the transaction is fraudulent or not.

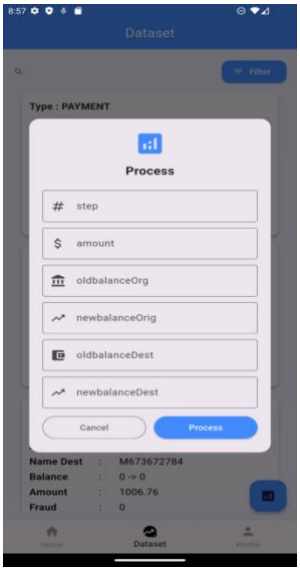


Fig. 18: Detection Process Page

Input Elements:
Users are asked to fill in several transaction parameters as input:
a. Step – The simulation step time in numerical format (e.g., minute 1, 2, etc.).
b. Amount – The transaction amount.
c. Old Balance Orig – The initial balance of the source account before the transaction.
d. New Balance Orig – The final balance of the source account after the transaction.
e. Old Balance Dest – The initial balance of the destination account before receiving the funds.
f. New Balance Dest – The final balance of the destination account after receiving the funds.

Buttons:
a. Process Button – Runs the fraud detection process based on the input data and displays the prediction result (fraud or non-fraud).
b. Cancel Button – Cancels the detection process and returns to the previous page without processing the input.

13. User Profile Page
This page is used to display information about the currently logged-in user account.

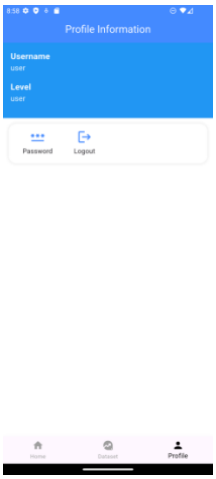


Fig. 19: User Profile Page

Users can view their basic account details and perform security-related actions such as changing the password or logging out of the application.
Displayed Information:
a. Username – Displays the currently logged-in user's name.
b. Level – Displays the user's role or access rights, e.g., regular user.
Buttons:
a. Password Button – Opens the password change page or dialog. Users can update their password for account security.
b. Logout Button – Logs out and returns to the login page.
14. Detection Result Notification Page

This page appears after the user presses the Process button on the Detection Process page. The system provides a notification showing the analysis results from the Random Forest algorithm based on the previously entered data.

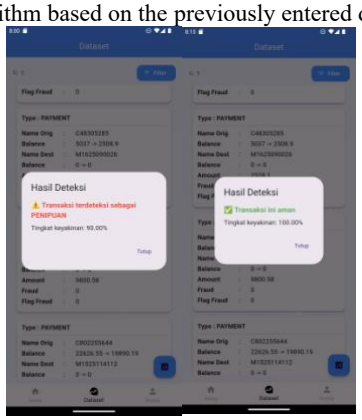


Fig.20: Detection Result Notification Page

Notification Content:
If the transaction is detected as fraudulent, the following text will appear:
"The transaction is detected as fraudulent."
If the transaction is not detected as fraud, the following text will appear:
"The transaction is safe."
Button:
Close Button – Closes the notification and returns to the previous page.

5. Conclusion

The conclusions of this study are as follows: the Random Forest model has been proven capable of delivering high accuracy and robustness against overfitting, making it effective in identifying fraud patterns in financial transaction data. The Flutter interface ensures a consistent and interactive cross-platform experience, facilitating both administrators and users in maintaining datasets and performing independent detection. The real-time addition, editing, and deletion of datasets provide administrators with flexibility to maintain data quality and periodically retrain the model. However, the model's performance is significantly affected by class imbalance, training time, and limited interpretability. In addition, reliance on API connections for inference may cause latency that disrupts the user experience.

References

[1] K. Rizkiyah, L. Nurmayanti, R. D. N. Macdhy, and A. Yusuf, "PENGARUH DIGITAL PAYMENT TERHADAP PERILAKU KONSUMEN DI ERA REVOLUSI INDUSTRI 4.0 (Studi Kasus Pengguna Platform Digital Payment OVO)," *Manag. Insight J. Ilm. Manaj.*, vol. 16, no. 1, pp. 107–126, 2021.
[2] T. S. Lestari and D. A. N. Sirodj, "Klasifikasi Penipuan Transaksi Kartu Kredit Menggunakan Metode Random Forest," *J. Ris. Stat.*, vol. 1, no. 2, pp. 160–167, 2022.
[3] P. T. S. Ningsih, M. Gusvarizon, and R. Hermawan, "Analisis Sistem Pendeteksi Penipuan Transaksi Kartu Kredit dengan Algoritma Machine Learning," *J. Teknol. Inform. dan Komput.*, vol. 8, no. 2, pp. 386–401, 2022.
[4] R. A. Billah, K. S., & Saputra, "Pendeteksian Penipuan Menggunakan Pendekatan Metode Random Forest," 2024.
[5] D. R. M. Mandyara, "Dampak Kecerdasan Buatan Dan Pembelajaran Mesin Terhadap Pendidikan: Tinjauan Bibliometrik," *J. PenKoMi Kaji. Pendidik. dan Ekon.*, vol. 7, no. 1, pp. 291–298, 2024.
[6] R. A. Putra, "Penerapan Machine Learning Dalam Deteksi Kecurangan Pada Transaksi Keuangan Online," *J. Dunia Data*, vol. 1, no. 4, pp. 1–16, 2024.
[7] R. S. Nurhalizah, R. Ardianto, and P. Purwono, "Analisis Supervised dan Unsupervised Learning pada Machine Learning: Systematic Literature Review," *J. Ilmu Komput. dan Inform.*, vol. 4, no. 1, pp. 61–72, 2024.
[8] E. Mardiani *et al.*, "Penerapan Algoritma Supervised Learning untuk Klasifikasi Data Music Listening," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 3, no. 2, pp. 115–124, 2023.
[9] E. Fitri, "Analisis Perbandingan Metode Regresi Linier, Random Forest Regression dan Gradient Boosted Trees Regression Method untuk Prediksi Harga Rumah," *J. Appl. Comput. Sci. Technol.*, vol. 4, no. 1, pp. 58–64, 2023.
[10] D. Mualfah, W. Fadila, and R. Firdaus, "Teknik SMOTE untuk Mengatasi Imbalance Data pada Deteksi Penyakit Stroke Menggunakan Algoritma Random Forest," *J. CoSciTech (Computer Sci. Inf. Technol.*, vol. 3, no. 2, pp. 107–113, 2022.
[11] I. Werdiningsih *et al.*, "Identifikasi Penipuan Kartu Kredit Pada Transaksi Ilegal Menggunakan Algoritma Random Forest dan Decision Tree," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 12, no. 3, pp. 477–484, 2023.
[12] R. Armiani and E. P. Agustini, "Analisa Fraud Pada Transaksi Kartu Kredit Menggunakan Algoritma Random Forest," *J. Teknol. Inf. dan Terap.*, vol. 9, no. 2, pp. 118–126, 2022.
[13] M. R. Givari, M. R. Sulaeman, and Y. Umaidah, "Perbandingan Algoritma SVM, Random Forest Dan XGBoost Untuk Penentuan Persetujuan Pengajuan Kredit," *Nuansa Inform.*, vol. 16, no. 1, pp. 141–149, 2022.
[14] D. R. K. Saputra, Y. V. Via, and A. N. Sihananto, "Deteksi Anomali Menggunakan Ensemble Learning Dan Random Oversampling Pada Penipuan Transaksi Keuangan," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 3, 2024.
[15] F. Zamachsari and N. Puspitasari, "Penerapan Deep Learning dalam Deteksi Penipuan Transaksi Keuangan Secara Elektronik," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 2, pp. 203–212, 2021.
[16] G. W. M. Kurniawan, "Pendeteksian Penipuan Menggunakan Pendekatan Metode Klasifikasi Random Forest," 2025.