JSCE (Journal of System and Computer Engineering)
Vol.6, No.3, July 2025, pp. 227-239
ISSN (online): 2723-1240
DOI : https://doi.org/10.61628/jsce.v6i3.2001

# Graph-Based Fraud Detection with Optimized Features and Class Balance

**Anisa Nur Azizah [1,a,*]; Alven Safik Ritonga [2,b]; Suryo Atmojo [3,c]; Nurwahyudi Widhiyanta [4,d]; Suzana Dewi [5,e]; M. Harist Murdani [6,f]; Mamik Usniyah Sari [7,g]**

*[1,2,3,4,5,6,7] Informatics Engineering, Faculty of Engineering, Universitas Wijaya Putra, Raya Benowo 1 – 3, Surabaya, Indonesia*

*[a] anisanurazizah@uwp.ac.id; [b] alvensafik@uwp.ac.id; [c] suryoatmojo@uwp.ac.id; [d]nurwahyudiwidhiyanta@uwp.ac.id; [e] suzanadewi@uwp.ac.id; [f] muhammadharist@uwp.ac.id; [g]mamikusniyah@uwp.ac.id*

** Corresponding author*

## Abstract

*The increasing use of digital transactions also elevates the risk of fraud, particularly in credit card transactions. Fraud detection poses a challenge due to the highly imbalanced nature of the data and the complexity of relationships among entities. This study proposes a GNN-based approach, integrated with feature selection techniques and class imbalance handling through class weighting based on data distribution. Feature selection was performed using two methods: Correlation-based Feature Selection (CFS) and Random Forest Feature Importance, to obtain the most relevant features. Experimental results show that the combination of Random Forest feature selection and class weighting yielded the highest F1 Score, despite a slight decrease in accuracy. This indicates that feature selection and class weighting strategies can improve the model's ability to detect rare fraudulent transactions. This approach contributes to the development of more accurate and adaptive fraud detection systems in digital transaction environments.*

***Keywords****—Fraud Detection, GNN, Feature Selection, Class Balance*

## 1. Introduction

Cybercrime in the form of credit card fraud is one of the growing financial threats in line with the increasing volume of digital transactions. Global losses due to credit card fraud are estimated to reach billions of dollars annually. Fraud-related losses increased by 140% in just two years, rising from USD 20 billion in 2021 to USD 48 billion in 2023, and are projected to exceed USD 343 billion during the period from 2023 to 2027 (Statista, 2024). All financial institutions are competing to enhance their digital security. The development of security systems must be capable of identifying anomalous transaction patterns to detect fraud in real time.

A study conducted by  (Husnaningtyas & Dewayanto, 2023) revealed that the most popular method used in unsupervised learning is K-Means, for identifying anomalies in credit card transactions. In addition, class imbalance and the complexity of fraudulent activities remain major challenges for researchers in building transaction fraud detection models. A subsequent study by (Billah, 2024) introduced credit card fraud detection using the Random Forest method. The developed model demonstrated good accuracy at approximately 96.53%, with several important features identified in the analysis of fraudulent credit card activities. In addition to machine learning-based methods, rule-based fraud detection systems are still widely used by financial institutions to filter high-risk transactions (Khanum et al., 2024). Research by (Aghware et al., 2024) combined the Random Forest algorithm with the Synthetic Minority

Oversampling Technique (SMOTE) to address the issue of class imbalance in credit card fraud detection. The model produced strong results, achieving an accuracy of 99.19%. Although these models deliver promising results, most of them rely on tabular data without considering the relationships between entities, which are often key characteristics of real-world fraud.

Conventional machine learning methods operate under the assumption that each data sample is an independent and identically distributed entity. In reality, fraudulent activities are often carried out in an organized manner through interconnected networks, such as the use of a single card at multiple locations within a short period of time, repeated transactions at specific merchants by the same perpetrator, or the dissemination of the same identity across different devices and IP addresses. Such relationships cannot be effectively captured by traditional tabular models, resulting in suboptimal detection of collaborative or coordinated fraud. In response to the increasing complexity and scale of sophisticated fraud attacks, analytical approaches capable of modeling relationships between entities within the transaction ecosystem are required. One highly promising approach is the utilization of graph structures. In the context of financial transactions, entities such as users, cards, merchants, and devices can be represented as *nodes*, while interactions or transactions between these entities are represented as *edges*. By using graph-based models, suspicious relational patterns such as the use of the same card by multiple users within a short period of time or unusual connections between accounts can be detected more effectively.

The study by (Li et al., 2022) identified that traditional machine learning models struggle to handle the large scale and structural complexity of financial transaction data. To address this issue, they proposed a graph-learning algorithm, TA-Struc2Vec, for Internet financial fraud detection. (Mao et al., 2022) employed a knowledge graph approach, which represents data by mapping structured and interconnected relationships between entities. A Related-Party Transactions (RPT) knowledge graph was constructed to detect financial fraud. Features such as transaction scale and type were used to enrich the detection model. The study by (Cherif et al., 2024) explored the use of Graph Neural Networks (GNNs) in credit card fraud detection by leveraging the relationships between customers and merchants in the form of graph structures. (Tang & Liang, 2024) proposed a credit card fraud detection (CCFD) model based on federated graph learning, which combines Federated Learning (FL) and GNN. Previous studies have shown that GNNs are effective in fraud detection due to their ability to capture inter-entity relationships. Prior research also emphasized the importance of feature engineering in constructing informative graphs. Therefore, the combination of GNN and feature processing is considered an appropriate approach for credit card fraud detection models.

In the context of financial transactions, feature selection is a crucial aspect that influences the performance of fraud detection models. Transaction data typically contains various features such as transaction time, amount, location, merchant type, device ID, and geolocation information. However, not all of these features are relevant or significantly contribute to the fraud identification process. A study by (Mienye & Sun, 2023) showed that many features were either irrelevant or redundant, which in fact degraded the model's performance. As a result, after performing feature selection, the model achieved a sensitivity of 0.997 and a specificity of 0.994, indicating a very high capability in fraud detection. The study by (Ileberi et al., 2022) emphasized the importance of feature selection in detecting fraud using machine learning. There are numerous feature selection techniques, and this study compared the traditional Correlation-based Feature Selection (CFS) method with the feature importance results obtained from a Random Forest model.

Based on previous studies, the main contributions of this research are to evaluate the Graph Neural Network (GNN) approach in the context of edge classification, to examine the impact of feature selection strategies on model performance, and to assess the effectiveness of handling class imbalance through weighting within the graph. The findings of this study are expected to serve as a foundation for developing fraud detection systems that are more adaptive, accurate, and aligned with the complexity of real-world digital financial transactions.

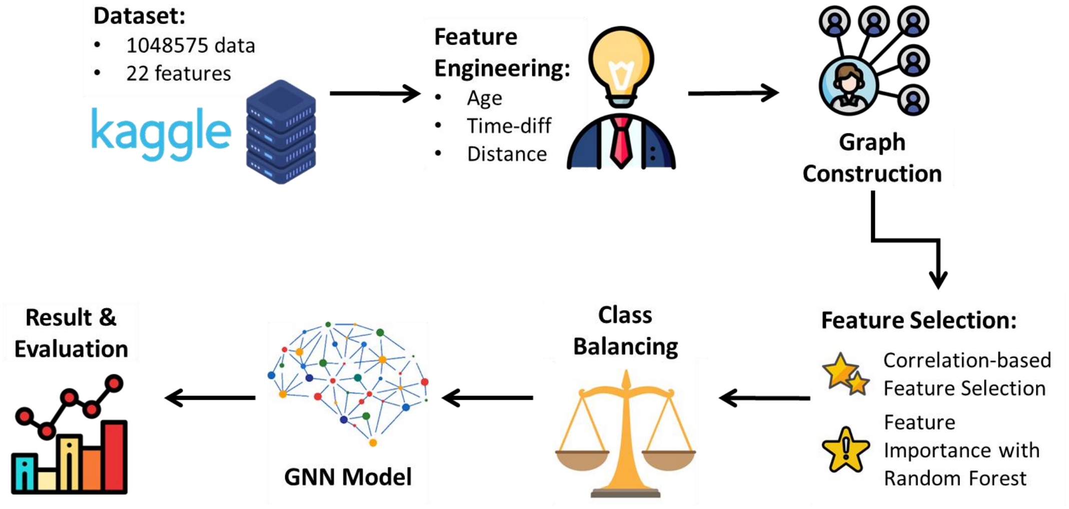


**Figure 1.** Flow Diagram of The Research Process

## 2. Method

This study evaluates the effectiveness of fraud detection on transaction data using a graph-based approach. The use of graphs in this context is considered more representative compared to conventional learning techniques. Graphs are able to capture patterns based on nodes and edges, which better reflect the actual conditions in transaction data. Each node represents the features of the involved parties, in this case, customers and merchants, such as age, gender, occupation, location, and others. Meanwhile, edges represent transaction-related features such as amount, time, category, and other relevant attributes.

In addressing big data challenges in transaction fraud detection, the researchers conducted several experiments to produce a more optimal model. This study applied various feature selection strategies for both nodes and edges in the graph data. Furthermore, the researchers acknowledged the high class imbalance in this problem and therefore implemented a class balancing approach by applying class-based weighting in the loss function during learning. The chosen learning model, Graph Neural Network (GNN), is considered an appropriate choice, as it can capture graph classification patterns in a deep and accurate manner due to its neural network-based architecture. The overall workflow of this study is presented in Figure 1.

### 2.1 Dataset

This study conducts fraud detection on credit card transaction data, with the dataset obtained from a data hosting platform, Kaggle (Kartik Shenoy, 2020). The dataset is a simulated credit card transaction dataset that includes various attributes related to customers, transaction information, and labels indicating whether a transaction is non-fraudulent or potentially fraudulent. The dataset contains a total of 23 features, including “trans_date_trans_time,” “cc_num,” “merchant,” “category,” “amt,” and the class label “is_fraud,” with a total of 1,048,575 transaction records. The recorded transaction times range from January 1, 2019, at 00:00:00 to March 10, 2020, at 16:08:00. The number of unique merchants is 693, and the number of unique customers is 960.

### 2.2 Feature Engineering

Big data analysis in credit card transactions is indeed essential. In reality, a large number of stored features require special handling in order to be transformed into meaningful and useful information to support the problem under study. Not all features in the dataset are utilized in this research; the goal is to ensure efficiency and reduce excessive computational load. Redundant features are eliminated and selected based on the requirements of the fraud detection analysis. Previously, the researchers conducted several studies on credit card fraud patterns (Cherif et al., 2024; Mao et al., 2022), which served as the basis for performing a series of feature engineering steps:

2.2.1 Age. This feature is calculated as the difference between the “trans_date_trans_time” and the cardholder’s date of birth “dob”. The addition of the “age” feature serves as a demographic attribute used to determine risk groups.

2.2.2 Time-diff. This feature represents the time difference between the current transaction and the previous transaction made by the same cardholder. The “time-diff” feature is included as a temporal attribute aimed at identifying automated or bot-generated transaction patterns, which typically exhibit very short time intervals.

2.2.3 Distance. This feature calculates the geographic distance between the user’s location and the merchant’s location using the geopy.distance.geodesic package. The “distance-km” feature is added as a spatial attribute intended to assess the plausibility of transaction locations in relation to the time interval between transactions.

## 2.3 Graph Construction

After performing feature engineering, the next step is to convert the tabular data into graph data. As previously explained, the features are divided into two categories: node_features and edge_features. The node_features consist of “age,” “job,” “city_pop,” “gender,” and “state” for each node representing either a “merchant” or a “cc_num” (credit card holder). Meanwhile, the edge_features include “amt,” “time_diff,” “distance_km,” “transaction_day,” “transaction_month,” “transaction_hour,” “transaction_min,” and “category.” Edges represent transaction events occurring between customers and merchants, with the class label “0” indicating a non-fraudulent transaction and “1” indicating a fraudulent transaction. The graph-based visualization of transaction data can be seen in Figure 2.

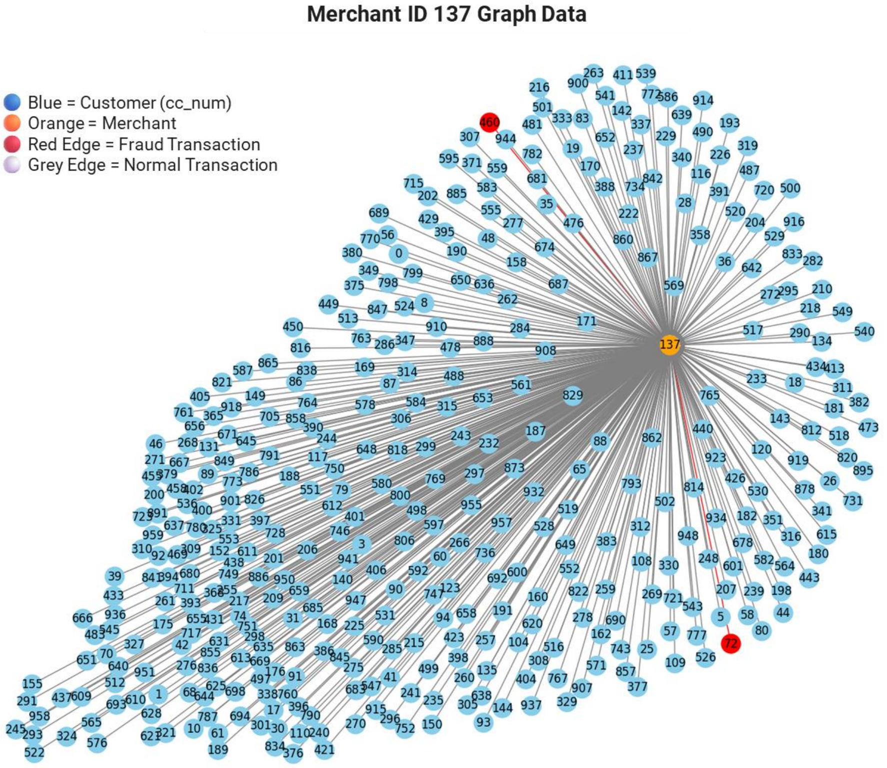

Merchant ID 137 Graph Data
● Blue = Customer (cc_num)
● Orange = Merchant
● Red Edge = Fraud Transaction
● Grey Edge = Normal Transaction
137
460
72

**Figure 2.** Graph Visualization of a Sample Merchant's Data

## 2.4 Feature Selection

This study employs several methods for feature selection, namely Correlation-based Feature Selection (CFS) and Feature Importance using Random Forest. The purpose of this feature selection process is to reduce the number of features to lessen computational burden and to optimize the performance of the learning model by focusing only on the most relevant features. Previous studies have also demonstrated that applying feature selection improves model performance compared to using a large number of features (Ileberi et al., 2022; Mienye & Sun, 2023). This improvement occurs because noisy or irrelevant features can negatively impact the model's learning process.

### 2.4.1 Correlation-based Feature Selection (CFS)

The concept of correlation involves measuring the extent to which two variables exhibit a linear relationship, and in the context of feature selection, it serves as a strategy to identify features with strong associations (Farida & Mustopa, 2023). The Pearson correlation formula is presented in Equation 1 (Gopika & ME, 2018).

$$r = \frac{n\sum X_i Y_i - \sum X_i \sum Y_i}{\sqrt{n\sum {X_i}^2 - (\sum X_i)^2}\sqrt{n\sum {Y_i}^2 - (\sum Y_i)^2}} \tag{1}$$

Where, r = the correlation coefficient between features X and Y, $X_i$ = the i-th value of the independent variable, $Y_i$ = the i-th value of the dependent variable, n = he number of data points. In this context, correlation emerges as a powerful preliminary tool in the feature selection process; however, broader policy considerations and evaluation strategies are necessary to ensure the accuracy and appropriateness of feature selection in more complex data analyses.

### 2.4.2 Feature Important with Random Forest

Random Forest is an ensemble learning algorithm based on decision trees, introduced by (Breiman, 2001). This method constructs multiple decision trees from randomly selected subsets of data and features, and then combines their outputs through voting (for classification) or averaging (for regression) (Istiqamah & Rijal, 2024). Random Forest is widely used due to its high accuracy, robustness against overfitting, ability to handle both numerical and categorical data, and its provision of feature importance metrics that enhance model interpretability.

Feature importance in Random Forest is a method used to evaluate the contribution of each feature to the model's prediction, based on how much the feature reduces impurity in the decision trees. This technique is known as Gini Importance in Equation 2.

$$FI(f) = \sum_{t\in T} \sum_{n\in N_t(f)} \frac{N_n}{N} \cdot \Delta Gini(n) \tag{2}$$

Let $T$ be the set of all trees in the random forest, $N_t(f)$ the set of nodes that use feature $f$, $N_n$ the number of samples reaching node $n$, and $N$ the total number of samples. $\Delta Gini(n)$, refers to the reduction in impurity caused by the split at a given node, which is calculated using Equation 3.

$$\Delta Gini(n) = Gini_{\text{parent}} - \left(\frac{N_L}{N_n} \cdot Gini_L + \frac{N_R}{N_n} \cdot Gini_R\right) \tag{3}$$

The Gini impurity itself is defined in Equation 4.

$$Gini = 1 - \sum_{i=1}^{C} p_i^2 \tag{4}$$

Where $p_i$ is the proportion of class *i* at the node, and *C* is the total number of classes. Features that are frequently used and result in a large impurity reduction will have higher importance scores. This technique is highly useful in feature selection, as it helps identify the most relevant features while also enhancing model interpretability and improving the efficiency of the machine learning process (Scornet, 2023).

## 2.5 Class Balancing

The dataset used in this study exhibits a highly imbalanced class distribution (“non_fraud” > “fraud”). The number of “non_fraud” or normal transaction records is 1,042,569, while the number of “fraud” records is 6,006, resulting in a class imbalance of approximately 98.8%. Therefore, a class balancing approach was applied by using class weighting in the loss function during the GNN model training. The purpose of class weighting is to assign a higher penalty to misclassifications of the minority class (“fraud”). The class weights were calculated proportionally based on the class label distribution in the dataset.

## 2.6 Graph Neural Network (GNN)

This study illustrates a fraud detection case on credit card transaction data represented in the form of a graph. Therefore, the selection of a graph-based model becomes the primary focus of this research. Unlike graph-based machine learning models such as decision trees or random forests, the graph referred to in this context is one that explicitly models the structure of the data through nodes and edges. Hence, the input to the model is a graph that represents the transaction data (S. Zhang et al., 2019). This study adopts a Graph Neural Network (GNN), not only for the reasons mentioned above, but also because GNN is capable of deeply learning data patterns through its neural network architecture.

Graph Neural Networks (GNNs) consist of various architectural variants designed to address different tasks and accommodate the unique structure of graph data. Among these, one of the most foundational and widely utilized models is the Graph Convolutional Network (GCN), introduced by (Kipf & Welling, 2016). The GCN architecture generalizes the convolution operation—commonly applied in Convolutional Neural Networks (CNNs)—to work with non-Euclidean graph structures. Instead of relying on fixed-grid data, GCN aggregates information from a node’s neighbors by employing a normalized adjacency matrix with added self-loops, thereby updating node features accordingly. This architecture typically stacks several graph convolution layers, each followed by nonlinear activation functions, to learn rich representations that capture both the local context and the topological structure of the graph, can be seen in Figure 3.

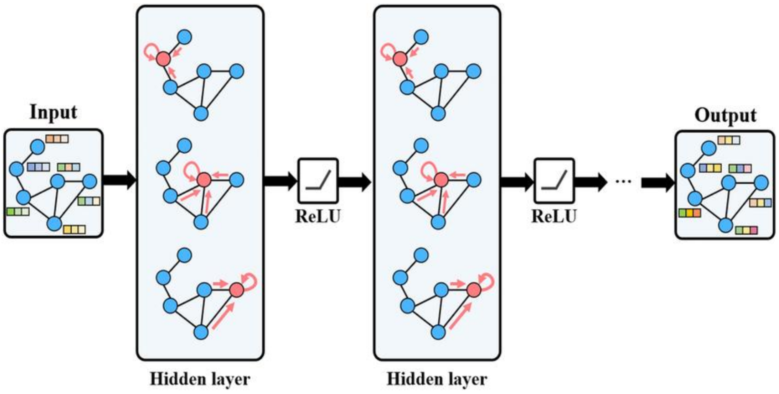


**Figure 3.** GCN Architecture (Kim et al., 2023)

GCN has been widely applied in numerous tasks, including node classification, link prediction, and graph classification (Liu et al., 2023). In the context of fraud detection, GCN demonstrates strong capabilities in identifying relational patterns between entities such as customers and merchants particularly in suspicious transactions that exhibit distinct network structures. Compared to conventional approaches, GCN offers a more robust framework for capturing the intricate structures embedded within transaction data.

## 2.7 Result and Analysis

This study evaluates the performance of the learning model using metrics based on the confusion matrix, namely accuracy, precision, recall, and F1-score. These metrics were chosen because they are appropriate for imbalanced binary classification problems, such as fraud detection, where the number of positive cases (fraud) is significantly lower than the number of negative cases (non-fraud). The accuracy score aims to measure the overall correctness of the model's predictions, although it can be biased in the case of imbalanced data. Precision is intended to assess how well the model avoids false positives, such as labeling a normal transaction as fraud. Recall highlights the main priority of not missing fraudulent cases, as false negatives can be particularly harmful. Meanwhile, the F1-score becomes the main focus, where the highest F1-score is considered the most optimal result because it handles class imbalance and maintains a balance between false positives and false negatives.

# 3. Results And Discussion

## 3.1 Research Experiment: Feature Optimization

At this stage, a feature engineering process was carried out to improve model performance and reduce data complexity. Out of the initial 23 available features, only 15 selected features were used during the model training process. This selection aimed to eliminate features that were redundant, irrelevant, or had low correlation with the target variable. A sample of the data after feature selection and feature engineering can be seen in Table 1.

**Table 1.** Sample Dataset Credit Card Transaction

| cc_ num | merchant | category | amt | gender | state | city pop | job | age | transaction _day | transaction _month | transaction _min | transaction _hour | time _diff | distance _km | is_ fraud |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 293 | 8 | 7.27 | 0 | 50 | 1645 | 246 | 32 | 1 | 1 | 47 | 12 | 0 | 127.6293 | 0 |
| 0 | 43 | 2 | 52.94 | 0 | 50 | 1645 | 246 | 32 | 2 | 1 | 44 | 8 | 71820 | 110.203 | 0 |
| 0 | 399 | 2 | 82.08 | 0 | 50 | 1645 | 246 | 32 | 2 | 1 | 47 | 8 | 180 | 21.84183 | 0 |
| 0 | 407 | 2 | 13.17 | 0 | 50 | 1645 | 246 | 33 | 1 | 3 | 32 | 1 | 23040 | 48.01185 | 1 |
| 0 | 119 | 2 | 11.74 | 0 | 50 | 1645 | 246 | 33 | 1 | 3 | 42 | 2 | 4200 | 88.36797 | 1 |
| 0 | 366 | 10 | 19.16 | 0 | 50 | 1645 | 246 | 33 | 1 | 3 | 6 | 23 | 73440 | 114.9529 | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 495 | 172 | 8 | 53.61 | 1 | 29 | 1517 | 120 | 68 | 18 | 2 | 10 | 9 | 63600 | 117.3829 | 0 |
| 495 | 645 | 4 | 129.74 | 1 | 29 | 1517 | 120 | 68 | 18 | 2 | 46 | 9 | 2160 | 42.51217 | 0 |
| 495 | 177 | 8 | 6.85 | 1 | 29 | 1517 | 120 | 68 | 18 | 2 | 56 | 10 | 4200 | 103.0412 | 0 |
| 495 | 472 | 4 | 326.42 | 1 | 29 | 1517 | 120 | 68 | 19 | 2 | 0 | 0 | 47040 | 84.07129 | 1 |
| 495 | 153 | 2 | 19.04 | 1 | 29 | 1517 | 120 | 68 | 19 | 2 | 17 | 3 | 11820 | 82.0153 | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 959 | 610 | 12 | 3.54 | 1 | 14 | 532 | 42 | 64 | 10 | 3 | 45 | 2 | 6540 | 79.02813 | 0 |
| 959 | 172 | 8 | 16.2 | 1 | 14 | 532 | 42 | 64 | 10 | 3 | 58 | 8 | 22380 | 70.63279 | 0 |

### 3.1.1 Correlation-based Feature Selection (CFS)

The CFS method was used to identify features that have a high correlation with the target variable *"is_fraud"*, but low correlation with other features. This approach aims to preserve the quality of information in the selected features while minimizing redundancy among them. The selection process was conducted by analyzing the Pearson correlation between features and the target label, with the correlation plot results presented in Figure 4.

As shown in Figure 4, the correlation of most features with *is_fraud* is generally low, with correlation values less than 0.1, indicating weak linear relationships. This is expected due to the highly imbalanced nature of the dataset, which weakens the overall correlation with the target class because of the limited number of samples labeled as "1" (fraud). Features such as *"cc_num"*, *"merchant"*, *"state"*, *"city_pop"*, *"distance_km"*, *"job"*, and *"transaction_min"* exhibit very low correlation values (less than |0.01|) with *"is_fraud"*, and were thus excluded from modeling. The final CFS feature selection resulted in 8 features: *"amt"*, *"category"*, *"time_diff"*, *"transaction_month"*, *"transaction_hour"*, *"transaction_day"*, *"age"*, and *"gender"*.

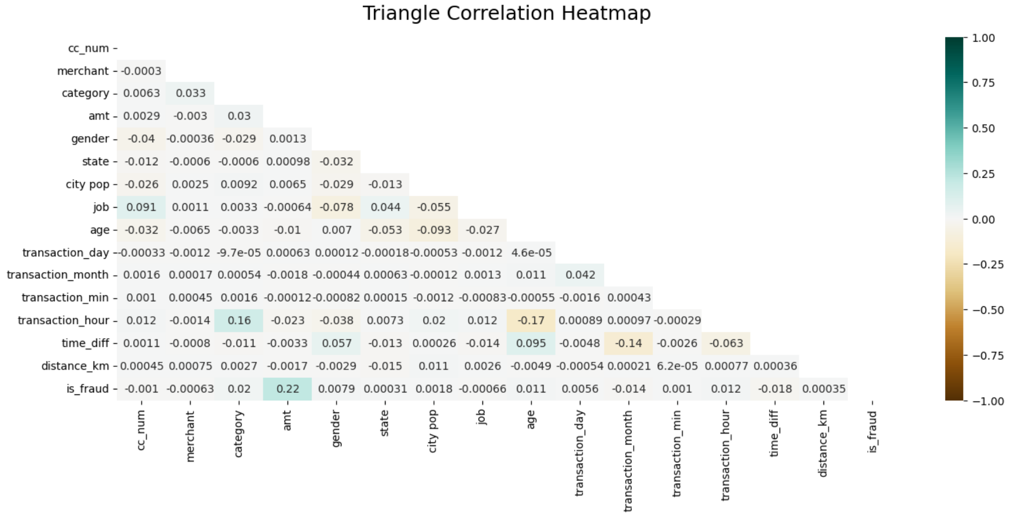


**Figure 4.** Plot of CFS Analysis Results on the Credit Card Transaction Dataset

### 3.1.2 Feature Important with Random Forest

In addition to correlation analysis, feature selection was also carried out using the feature importance approach based on the Random Forest algorithm. Random Forest automatically computes the importance score of each feature based on its contribution to reducing impurity in the decision trees. The feature importance plot generated by Random Forest is presented in Figure 5.

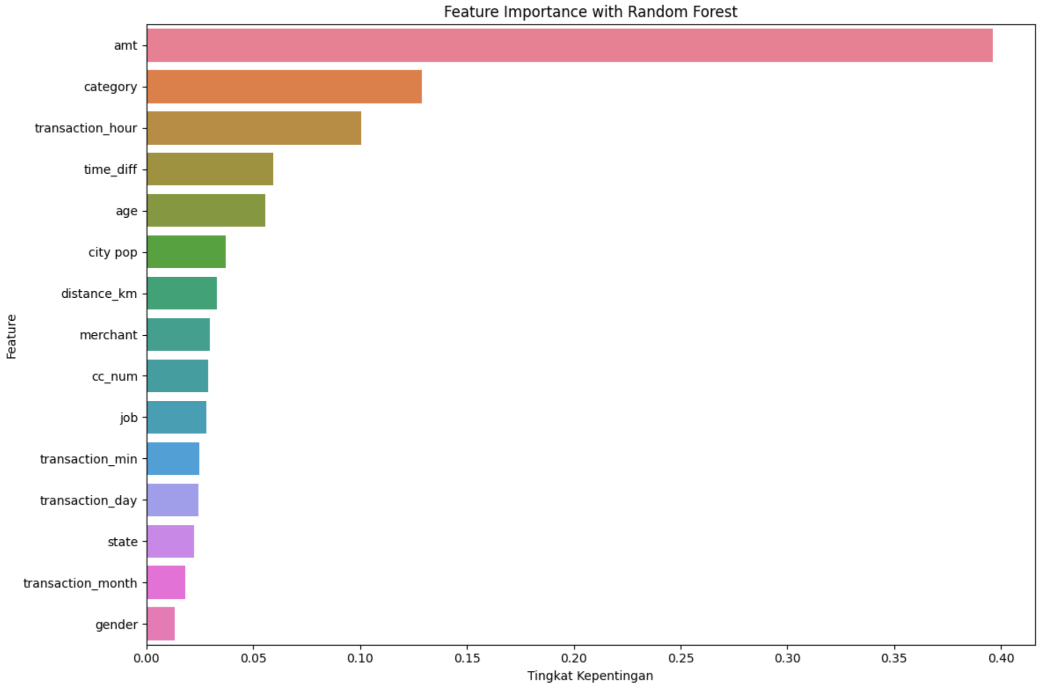


**Figure 5.** Plot of Feature Importance with Random Forest Dataset Credit Card Transaction

According to the results shown in Figure 5, the feature *"amt"* demonstrates a significantly higher influence compared to the other features. However, the researcher did not rely on a single feature alone. Taking several considerations into account, a total of 7 features were selected: *"amt"*, *"category"*, *"transaction_hour"*, *"time_diff"*, *"age"*, *"city_pop"*, and *"distance_km"*.

### 3.2 Result and Analysis

This section presents the results of the fraud detection model experiments based on the selected features. Evaluation was conducted using classification metrics such as accuracy, precision, recall, and F1-score. The model employed is based on the Graph Convolutional Network (GCN) architecture, consisting of two GCN layers and one Multilayer Perceptron (MLP) for edge-level classification. The implementation was carried out using the PyTorch Geometric (PyG) framework. The model was trained for 50 and 100 epochs using the Adam optimizer with a learning rate of 0.01 and the CrossEntropyLoss function. The results of various approaches were compared two training performance evaluations were conducted: one based on the best accuracy and the other on the best F1-score, as presented in Table 2 and Table 3.

**Table 2.** Evaluation Results on Testing Data Using the Model with the Best Accuracy Performance During Training

| Epoch | Features | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| 50 epoch | all_no_balance | 0.994273 | 0 | 0 | 0 |
| 50 epoch | corr_no_balance | 0.994273 | 0 | 0 | 0 |
| 50 epoch | rf_no_balance | 0.990916 | 0.009749 | 0.005828 | 0.007295 |
| 50 epoch | all_balance | 0.994273 | 0 | 0 | 0 |
| 50 epoch | corr_balance | 0.994273 | 0 | 0 | 0 |
| 50 epoch | rf_balance | 0.994273 | 0 | 0 | 0 |
| 100 epoch | all_no_balance | 0.991894 | 0.009823 | 0.004163 | 0.005848 |
| 100 epoch | corr_no_balance | 0.994273 | 0 | 0 | 0 |
| 100 epoch | rf_no_balance | 0.994273 | 0 | 0 | 0 |
| 100 epoch | all_balance | 0.005727 | 0.005727 | 1 | 0.011388 |
| 100 epoch | corr_balance | 0.269671 | 0.005834 | 0.746878 | 0.011578 |
| 100 epoch | rf_balance | 0.005727 | 0.005727 | 1 | 0.011388 |

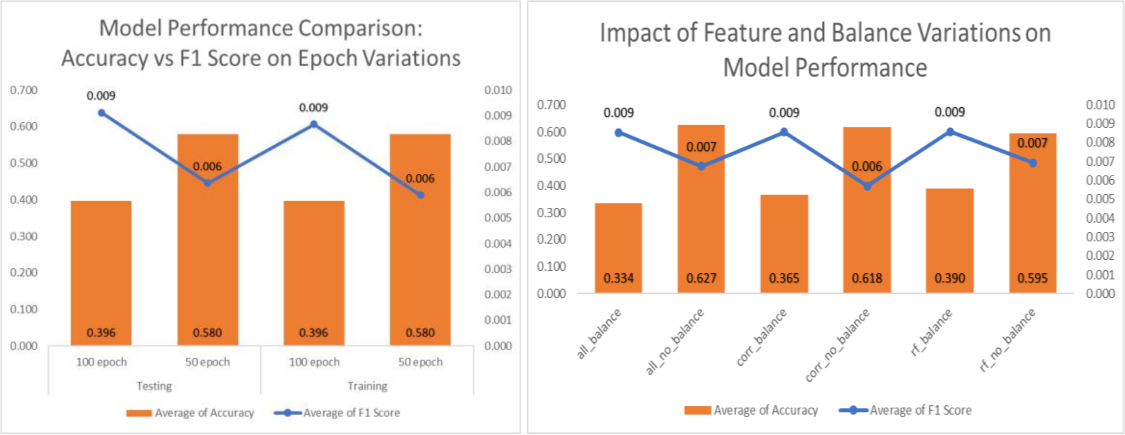
Table 2 presents the evaluation results of model selection based on the highest training (or validation) accuracy during the training process. The results indicate that models with high accuracy tend to exhibit low precision, recall, and F1-score, particularly in detecting fraud cases. On the other hand, the model trained with 100 epochs and class balancing achieved a higher recall despite a decrease in accuracy. This highlights a trade-off between accuracy and fraud detection capability, emphasizing the importance of using more representative evaluation metrics such as recall and F1-score when dealing with imbalanced datasets.

Next, if we look at the model with optimization selection based on the F1-Score value, Table 3, the model with the combination of *"rf_balance"* and 100 epochs was selected due to its highest F1-score (0.011726) and high recall value (0.73189), which are critical in the context of fraud detection with imbalanced data. A high recall indicates that the model is capable of identifying most fraudulent transactions, although the low precision reveals a considerable number of false positive predictions. This phenomenon is common in fraud detection problems, where fraudulent transactions are significantly fewer than legitimate ones, and improvements in recall often come at the expense of precision. In the context of an early warning system, this approach remains valuable, as it is preferable to detect potential fraud even with some false alarms rather than fail to identify actual fraudulent activities. However, in real-time system implementations or financial environments, a high false positive rate can have significant consequences, as legitimate transactions may be rejected. This can reduce customer satisfaction, cause service disruptions, and even damage the institution's reputation. Therefore, in developing a fraud detection system, it is crucial to maintain a balance between recall and precision.

**Table 3.** Evaluation Results on Testing Data Using the Model with the Best F1-Score Performance During Training

DOI : https://doi.org/10.61628/jsce.v6i3.2001

| Epoch | Features | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| 50 epoch | all_no_balance | 0.260501 | 0.005838 | 0.756869 | 0.011587 |
| 50 epoch | corr_no_balance | 0.280748 | 0.005858 | 0.738551 | 0.011624 |
| 50 epoch | rf_no_balance | 0.005727 | 0.005727 | 1 | 0.011388 |
| 50 epoch | all_balance | 0.014648 | 0.005755 | 0.995837 | 0.011443 |
| 50 epoch | corr_balance | 0.173397 | 0.005742 | 0.832639 | 0.011406 |
| 50 epoch | rf_balance | 0.265045 | 0.005861 | 0.755204 | 0.011632 |
| 100 epoch | all_no_balance | 0.261836 | 0.005849 | 0.756869 | 0.011608 |
| 100 epoch | corr_no_balance | 0.204668 | 0.00579 | 0.80766 | 0.011497 |
| 100 epoch | rf_no_balance | 0.388422 | 0.005757 | 0.616153 | 0.011408 |
| 100 epoch | all_balance | 0.323019 | 0.005834 | 0.691923 | 0.011571 |
| 100 epoch | corr_balance | 0.023737 | 0.00575 | 0.985845 | 0.011434 |
| 100 epoch | rf_balance | 0.29347 | 0.00591 | 0.73189 | 0.011726 |



(a) (b)

**Figure 6.** (a) Training vs Testing per Epoch (b) Feature & Balancing Variasi

Based on the visualization results of model performance, in Figure 6, several important findings should be noted. In the Figure 6(a), which compares accuracy and F1-score across different numbers of epochs for both training and testing data, it is observed that model accuracy remains consistently high and stable across all configurations, including at 50 and 100 epochs. However, the F1-score is significantly lower and fluctuates, indicating that the model struggles to detect the minority class (fraud). This suggests a potential issue of overfitting, where the model performs well in predicting the majority class (non-fraud) but fails to capture patterns in the fraud class. Therefore, in the context of imbalanced data, high accuracy should not be the sole metric for evaluating model performance.

Meanwhile, the Figure 6(b) compares model performance across feature variations and class balancing implementations. The results indicate that using all features without class balancing yields the highest accuracy (around 0.62), but still results in a low F1-score. Conversely, the combination of feature selection using Random Forest and the application of class balancing (*rf_balance*) achieves the highest F1-score (0.009), although with a decrease in accuracy (~0.39). This highlights a trade-off between accuracy and the model’s ability to identify fraudulent transactions. However, since the primary objective of this study is to detect rare yet critical fraud cases, the F1-score is considered a more relevant evaluation metric. Therefore, the *rf_balance* configuration is selected as the best-performing setup, as it is more effective in capturing fraudulent patterns compared to other approaches.

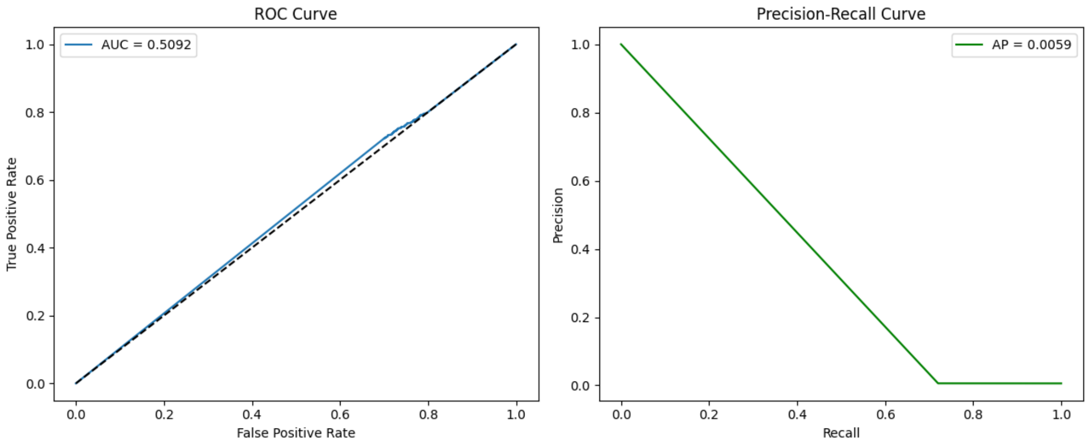


**Figure 7.** ROC Curve and Precision-Recall Curve Graphs of Random Forest Model with Balancing Class

Figure 7 shows that the AUC score of 0.5092 indicates that the model's ability to distinguish between fraud and non-fraud transactions is nearly equivalent to random guessing, as reflected by the ROC curve approaching the diagonal line. Additionally, the very low Average Precision (AP) score of 0.0059 suggests a severe imbalance between precision and recall, with the model generating a large number of false positives. This indicates that the model has not yet successfully identified meaningful patterns in the data and poses a risk of triggering false alarms, which is particularly critical in real-time or financial fraud detection systems.

The t-SNE technique was selected due to its ability to capture local structure, making it suitable for analyzing the distribution of minority classes such as fraud. Figure 8 shows that most edges labeled as fraud (minority class) remain mixed within the majority clusters. However, there are indications that the model has begun to form distinct representations for suspicious transactions. This observation is supported by the clustering evaluation results based on the embeddings, where the low silhouette score (0.0612) and the nearly equal intra-cluster and inter-cluster distances indicate that class separation has not yet been achieved effectively. Nevertheless, the high cluster purity score (0.9940) reflects the dominance of the majority class in the data structure rather than effective class separation.

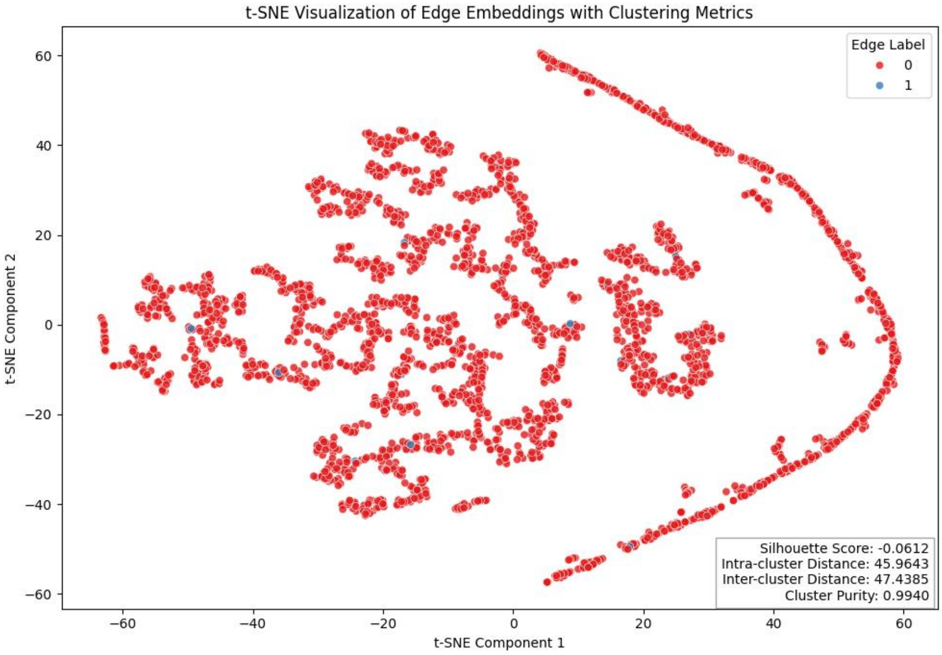


**Figure 8.** t-SNE Visualization of Edge Embeddings on 3,000 Data Samples

### 3.3 Discussion

The results indicate that although the model achieved relatively high accuracy, the F1-score remained low due to class imbalance in the data. Balancing techniques and feature selection particularly using Random Forest proved effective in enhancing the model's ability to detect fraudulent transactions. This highlights the importance of the feature engineering stage in building an effective fraud detection model. Graph-based models such as Graph Neural Networks (GNN) show great potential in mapping the relationships between entities in transaction data. However, the model’s performance heavily depends on the quality and representation of the features used to construct the graph.

For future research, it is recommended to explore more advanced Graph Neural Network (GNN) architectures, such as Graph Attention Network (GAT) (Vrahatis et al., 2024) or GraphSAGE (T. Zhang et al., 2022), which are capable of capturing deeper relationships among entities. Additionally, alternative class balancing techniques use more precise graph-based oversampling approaches such as GraphSMOTE (Zhao et al., 2021), which is specifically designed to address imbalance in graph data.

## 4. Conclusions

This study demonstrates that the combination of appropriate feature selection and effective handling of class imbalance can significantly enhance model performance in detecting fraudulent transactions. Although accuracy is not the primary indicator in imbalanced data scenarios, the application of feature selection using Random Forest and class balancing methods proved to yield better F1-scores. This indicates that the model is more capable of identifying rare but highly impactful fraud patterns. Therefore, a graph-based approach combined with proper feature engineering techniques can serve as a potential solution for developing more accurate and adaptive fraud detection systems.

## References

Aghware, F. O., Ojugo, A. A., Adigwe, W., Odiakaose, C. C., Ojei, E. O., Ashioba, N. C., Okpor, M. D., & Geteloma, V. O. (2024). Enhancing the random forest model via synthetic minority oversampling technique for credit-card fraud detection. *Journal of Computing Theories and Applications*, *1*(4), 407–420.

Bank Indonesia. (2024). Blueprint Sistem Pembayaran Indonesia 2030 Bank Indonesia: Mengakselerasi Ekonomi Digital Nasional untuk Generasi Mendatang. *Bspi 2030*. https://www.bi.go.id/id/publikasi/kajian/Documents/Blueprint-Sistem-Pembayaran-Indonesia-2030.pdf

Billah, K. S. (2024). DETEKSI PENIPUAN KARTU KREDIT MENGGUNAKAN METODE RANDOM FOREST. *JOISIE (Journal Of Information Systems And Informatics Engineering)*, *8*(2), 200–208.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32.

Cherif, A., Ammar, H., Kalkatawi, M., Alshehri, S., & Imine, A. (2024). Encoder-decoder graph neural network for credit card fraud detection. *Journal of King Saud University-Computer and Information Sciences*, *36*(3), 102003.

Dublin. (2025). *Credit Card Issuance Services Market Trends, Strategies and Growth Opportunities 2025-2029 & 2034 - Global Revenues to Exceed $787 Billion by 2029*. GLOBE NEWSWIRE. https://www.globenewswire.com/news-release/2025/03/24/3047674/28124/en/Credit-Card-Issuance-Services-Market-Trends-Strategies-and-Growth-Opportunities-2025-2029-2034-Global-Revenues-to-Exceed-787-Billion-by-2029.html

Farida, F., & Mustopa, A. (2023). Comparison of logistic regression and random forest using correlation-based feature selection for phishing website detection. *Sistemasi: Jurnal Sistem Informasi*, *12*(1), 13–20.

*DOI : https://doi.org/10.61628/jsce.v6i3.2001*

Gopika, N., & ME, A. M. K. (2018). Correlation based feature selection algorithm for machine learning. *2018 3rd International Conference on Communication and Electronics Systems (ICCES)*, 692–695.

Husnaningtyas, N., & Dewayanto, T. (2023). FINANCIAL FRAUD DETECTION AND MACHINE LEARNING ALGORITHM (UNSUPERVISED LEARNING): SYSTEMATIC LITERATURE REVIEW. *Jurnal Riset Akuntansi Dan Bisnis Airlangga (JRABA)*, *8*(2).

Ileberi, E., Sun, Y., & Wang, Z. (2022). A machine learning based credit card fraud detection using the GA algorithm for feature selection. *Journal of Big Data*, *9*(1), 24.

Istiqamah, N., & Rijal, M. (2024). Klasifikasi Ulasan Konsumen Menggunakan Random Forest dan SMOTE. *Journal of System and Computer Engineering*, *5*(1), 66–77.

Kartik Shenoy. (2020). *Credit Card Transactions Fraud Detection Dataset*. Kaggle Dataset. https://www.kaggle.com/datasets/kartik2112/fraud-detection/data?select=fraudTrain.csv

Khanum, A., Chaitra, K. S., Singh, B., & Gomathi, C. (2024). Fraud Detection in Financial Transactions: A Machine Learning Approach vs. Rule-Based Systems. *2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*, 1–5.

Kim, M., Lee, J., & Kim, J. (2023). GMR-Net: GCN-based mesh refinement framework for elliptic PDE problems. *Engineering with Computers*, *39*(5), 3721–3737.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *ArXiv Preprint ArXiv:1609.02907*.

Li, R., Liu, Z., Ma, Y., Yang, D., & Sun, S. (2022). Internet financial fraud detection based on graph learning. *IEEE Transactions on Computational Social Systems*, *10*(3), 1394–1401.

Liu, X., Chen, J., & Wen, Q. (2023). A survey on graph classification and link prediction based on gnn. *ArXiv Preprint ArXiv:2307.00865*.

Mao, X., Sun, H., Zhu, X., & Li, J. (2022). Financial fraud detection using the related-party transaction knowledge graph. *Procedia Computer Science*, *199*, 733–740.

Mienye, I. D., & Sun, Y. (2023). A machine learning method with hybrid feature selection for improved credit card fraud detection. *Applied Sciences*, *13*(12), 7254.

Scornet, E. (2023). Trees, forests, and impurity-based variable importance in regression. *Annales de l'Institut Henri Poincare (B) Probabilites et Statistiques*, *59*(1), 21–52.

Statista. (2024). *Value of e-commerce losses to online payment fraud worldwide in 2023 and 2024, with forecasts for 2029*. Statista 2025. https://www.statista.com/statistics/1273177/ecommerce-payment-fraud-losses-globally/

Subramaniam, D. N., Jeyananthan, P., & Sathiparan, N. (2024). Soft computing techniques to predict the electrical resistivity of pervious concrete. *Asian Journal of Civil Engineering*, *25*(1), 711–722.

Tang, Y., & Liang, Y. (2024). Credit card fraud detection based on federated graph learning. *Expert Systems with Applications*, *256*, 124979.

Vrahatis, A. G., Lazaros, K., & Kotsiantis, S. (2024). Graph attention networks: a comprehensive review of methods and applications. *Future Internet*, *16*(9), 318.

Zhang, S., Tong, H., Xu, J., & Maciejewski, R. (2019). Graph convolutional networks: a comprehensive review. *Computational Social Networks*, *6*(1), 1–23.

Zhang, T., Shan, H.-R., & Little, M. A. (2022). Causal GraphSAGE: A robust graph method for classification based on causal sampling. *Pattern Recognition*, *128*, 108696.

Zhao, T., Zhang, X., & Wang, S. (2021). Graphsmote: Imbalanced node classification on graphs with graph neural networks. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 833–841.