Tamjeed Journal of Healthcare Engineering and Science Technology (TJHEST) ISSN: 2583-7281
***Vol.3, issue* (2),2025, *Page* 42-54**

Enhancing Fraud Detection in Banking: A Hybrid Approach Combining Graph Algorithms and Machine Learning

Yi Zhu[1], Shihab A. Shawkat [2,*] Shafeeq Kanaan Shakir AlDoori [3]

[1] Department of Artificial Intelligence, Nanjing University, Nanjing, Jiangsu 210003, China; 194160330@smail.cczu.edu.cn
[2] Department of Quality Assurance and Academic Performance, University of Samarra, Salah al-Din, Iraq; shahab84ahmed@gmail.com
[3] Department of Physics, College of Education, University of Samarra, Salah-AL-Din, Iraq; shafiq.k.shaker@uosamarra.edu.iq
* Correspondence: shahab84ahmed@gmail.com.

Received: 22.10.2025

Revised: 12.11.2025

Accepted: 21.12.2025

Published:27.12.2025

Abstract: Financial institutions face significant losses from increasingly sophisticated fraud attacks that evade traditional detection methods. This study proposes a hybrid approach combining machine learning (ML) (XGBoost, Random Forest, SVM, k-NN) with graph algorithms (PageRank, Community Detection, Degree Centrality) to enhance fraud detection accuracy in banking transactions. Using the BankSim dataset containing 587,443 legitimate and 7,200 fraudulent transactions we first address class imbalance through under sampling, then integrate graph-based features extracted via Neo4j to capture complex transactional relationships. Our methodology demonstrates that combining graph analytics with machine learning yields superior performance compared to standalone models, achieving precision scores up to 0.93 (k-NN) and recall rates of 0.87 (XGBoost). The hybrid approach also reduces training and prediction times by 2.9% and 6.8%, respectively, validated through 5-fold cross-validation. Key findings highlight that graph-augmented features improve F1-scores by 4–7% over conventional methods, with Random Forest and k-NN showing the most significant gains. This work contributes a practical framework for financial institutions to leverage interconnected transaction data, balancing detection accuracy (minimizing false negatives) and operational efficiency (reducing false positives). Future directions include testing this approach on real-time transaction streams and expanding to multi-modal fraud detection.

Keywords: Fraud detection; Graph algorithms; Machine learning; Neo4j; BankSim dataset; Hybrid modeling.

1. Introduction

Financial institutions and insurance firms incur annual losses amounting to millions of dollars as a result of fraudulent activities. Although conventional fraud detection techniques are crucial in mitigating these losses, increasingly advanced fraud schemes have become challenging to identify, both through collaboration and by employing diverse methods to create counterfeit identities [1]. Although no fraud prevention method works perfectly, significant improvement opportunities can be found by looking beyond individual data points to the connections that connect them [2]. Graph data mining is derived from frequentist pattern mining, which focuses on subgraphs, and graph subgraph mining is a popular extension of graph mining [3-5]. Standard machine learning methods are inadequate in processing and visualizing data from various data sources and increasing data

size [6,7]. In addition, in the face of evolving fraud attacks, standard machine learning methods may not be able to detect some of the fraudulent transactions in this attack. As a solution to these problems, this study aims to increase the accuracy of machine learning methods currently used in fraud detection and to facilitate the applicability of the study by using graph databases and graph algorithms. In this study, normal payments and fraudulent payments on the BankSim data set are classified with RF, SVM, XGB, k-NN classification algorithms. To perform fraud detection, it is aimed to detect fraudulent transactions that are difficult to detect with traditional fraud detection methods by using classical machine learning algorithms and graph databases and to express the relationship between the data better compared to other methods.

In this study, unlike other studies in the literature, the effect of machine learning techniques and graph mining algorithms on the performance of fraud detection processes has been examined in detail. The presented study contributes to the literature in showing that better results can be obtained by using graph mining and ML algorithms together in this field. The literature review section of the study presents current studies in the literature on fraud detection for credit and debit card transactions. The methodology section includes a general overview of the dataset, classification algorithms, data preprocessing methods and tools used in the study. The Neo4j and dataset section includes the details of the dataset, the definition of the Neo4j tool and its use in visualizing the dataset. The data preprocessing section includes the details of the data preprocessing stages. The ML and graph algorithms section includes the details of the ML classification algorithms and graph algorithms used. Finally, the experimental results are given in the experimental results section and the results obtained in this study are listed together with the results of ML methods using the same preprocessing and dataset in the literature.

2. Literature Review

The remarkable expansion of digital payments in recent years has instigated substantial alterations in fraud and financial crimes. In this novel environment, conventional detection methods, such as rule-based systems, have largely proven ineffective, while artificial intelligence and ML solutions have garnered significant interest. Doğan [8] provided an overview of the common application issues and comprehensive implementation challenges faced by graph-based solutions in fraud and financial crime detection. The remarkable expansion of digital payments in recent years has instigated substantial alterations in fraud and financial crimes. In this novel environment, conventional detection methods, such as rule-based systems, have largely proven ineffective, while artificial intelligence and ML solutions have garnered significant interest. The use of manual methods in fraud detection requires domain knowledge in the feature extraction phase. This requires focusing on the fraud behaviour models closest to the fraud behaviours in the online fraud detection system.

Wang, *et al.* [16] employs a knowledge graph that comprehensively delineates the co-occurrence relationships of transaction attributes related to the problem. The efficacy of the proposed method is validated through experiments conducted on an actual dataset from a commercial bank. This research is the inaugural investigation to apply data for varied behavioural models utilizing network embedding algorithms at the feature level. The proposed method demonstrates superior performance compared to leading classifiers. Ogundokun, *et al.* [17] used supervised ML techniques for fraud detection. Using domain-related constraints, a method was proposed to create a probabilistic graphical model for fraud detection. Bayesian Network algorithms such as K2 search, Hill-Climbing, and Simulated Annealing were used on the BankSim dataset in the study. Lopez, *et al.* [18] detected fraud on the Banksim dataset using k-NN, XGB, and RF ML algorithms. In this study, the unbalanced dataset was balanced with the under-sampling method. The k-NN precision value was found as 0.83, the recall value as 0.61, and the f1 score value as 0.70. The same values were found as 0.89, 0.76, 0.82 for XGB, and 0.24, 0.98, 0.82 for RF, respectively Gorton*, et al.* [19] discussed the most effective methods and ML algorithms for fraud detection in credit payments. After balancing the dataset with the under-sampling method on the Banksim dataset, the precision value for k-NN was 0.80, the precision value for SVM was 0.77,

and the precision value for RF was 0.93. For K-NN, precision, recall, and f-score values were obtained as 0.80, 0.80, 079; for SVM, the same values were obtained as 0.76, 0.77, 0.76; for RF, the same values were obtained as 0.93, 0.93, and 0.93, respectively. Table 1 presents a comparison of the literature studies. It was observed that probabilistic graphical models performed better than other basic techniques with 99.272% accuracy.

Table 1. Comparative analysis of ML methods in the literature review.

Ref.	Methods	Algorithms	Dataset	Metrics
[9]	Spatio-Temporal Attention Graph Network (STAGN)	Logistic Regression, Gradient Boosting, Multilayer Perceptron, AdaBoost, STAGN	Credit card transactions	AUC, Recall
[10]	Text-Associated Deep Walk (TADW)	Deep Walk, Probabilistic Latent Semantic Analysis (PLSA), Transductive SVM, Bipartite Graph Embedding	Cora, Wiki, Citeseer datasets	Matrix Factorization
[11]	Point of Interest (POI) Refinement	Bipartite Graph Embedding Optimization, Co-Location Learning	Foursquare, Gowalla location data	Precision, Recall, Accuracy, Rank
[12]	Deep Learning	ANN, RNN, LSTM, GRU	Retail banking transaction data	Precision, Recall, Accuracy
[15]	Deep Transfer Learning	Deep Neural Networks	5 months of e-commerce and in-person transaction data	Precision, Recall, AUC-PR
[16]	Graph Representation, Network Embedding	Multi-Agent Behavioral Models, Logistic Regression (LR), RF, XGBoost, CNN, Naive Bayes	3 months of B2C/C2C bank transactions from China	Precision, Recall, AUC-ROC
[17]	Machine Learning	Naive Bayes, Hill-Climbing, Simulated Annealing, K2 Search	Banksim dataset	Precision, Recall, Accuracy
[18]	Machine Learning	RF, XGBoost, k-NN)	Banksim dataset	Precision, Recall, F1-Score
[19]	Machine Learning	k-NN, SVM, RF	Banksim dataset	Precision, Recall, F1-Score, Accuracy

Considering the literature studies [13,14] examined, it was observed that graph-based methods provide higher accuracy rates and faster returns compared to ML methods. In this study, unlike the studies in the literature, ML algorithms were used together with graph algorithms to optimize performance results. This study contributes to the literature in showing that ML algorithms can be used together to obtain better results.

3. Methodology

Graph data science enables the discovery and analysis of network structures using searches, queries, and graph algorithms. In addition, by utilizing graph theory, a subfield of discrete mathematics, it increases fraud prediction accuracy and can save financial services firms millions of dollars even if it provides small percentage accuracy increases [20].

Even if the data to be used for fraud detection is collected from various places in various formats, they do not provide any value on their own. In order to make sense of the data, communication between the data and the data should be established and information production should be provided by organizing the data. This information and the relationships between them are quite complex and the transformation process requires a lot of effort. At the end of a long transformation process, the data finds its true value and obtaining a learning that can be turned into action by obtaining value from this information requires intelligence. This intelligence is machine learning [21]. The graph model helps manage well-analysed data. Querying graphs allows you to combine data from multiple sources and easily find and extract the variable list for training, which is much easier than querying relational or NoSQL databases. It speeds up the model creation process, easily combines data with external information sources and data can be exported in the desired format. Due to all these advantages, ML methods were combined with graph mining methods while performing fraud analysis in this study, and the accuracy of the analysis was increased. Neo4j tool was used for graph database and graph mining. Neo4j is the largest graph community, provides high-performance reading and writing scalability, provides high performance in graph storage and processing, is easy to learn and use, and is reliable, which are the reasons why Neo4j was chosen in this study [22]. ML methodologies encompass six phases of the CRISP-DM framework [23]. SVM, RF, XGB, k-NN algorithms from ML classification algorithms were used for fraud detection and performance evaluations were performed using 5-fold cross-validation.

3.1. Dataset and neo4j

In this study, BankSim dataset based on a sample of mass bank payments belonging to a bank in Spain was used [24]. The dataset comprises legitimate payment records and fraudulent data. BankSim was executed for roughly six months in 180 iterations, and the parameters were adjusted to achieve a dependable distribution for the assessment. 594643 records were generated, including 587443 normal payments and 7200 fraudulent payments [25]. Neo4j is a graph database tool used by thousands of companies and organizations to improve products and services in almost all sectors including financial services, energy, management, technology, retail and manufacturing [26]. In this study, Neo4j database and CypherQL language were used to visualize the dataset that passed the pre-processing stage. The visualized dataset was connected to Neo4j database with Python 3 and retrieved from the database with CypherQL query language. The basic structure of the graph dataset visualized with CypherQL is shown in Figure 1.

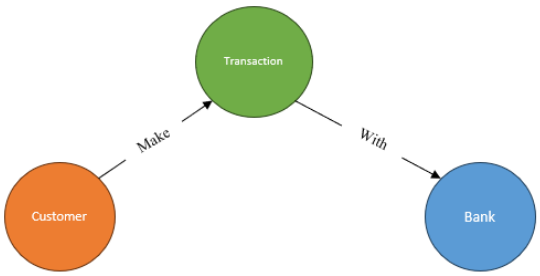


Figure 1. Neo4j graph dataset structure.

The graph dataset created with CypherQL in Figure 1 was later expanded by adding customer and bank indexes connected to the Placeholder node. Here, Customer and Bank are labelled as Constraint (main nodes). The connection between the two is made with the transaction node. Graph algorithms determine nodes with a high value in the network, were performed on the customer and bank indexes labelled as Placeholder. The purpose of this process is to determine how much of the network it will affect when a node is wanted to be disrupted. For example, if it is assumed that there is a transformation process in a network that only converts one of the end customers with the highest value, there are not many paths through the process. Therefore, the intermediate centrality of the node is quite low, but the node's value is high because it affects an important element in the chain. The placeholder node is seen in Figure 2. Each customer and bank index connected to the Placeholder node contains the values of Community, Degree, and PageRank. Graph algorithms are implemented on the Placeholder tag and the Placeholder's connection is made to itself with the

payments link. The dataset indexes extended with Cypherql and the Cypherql code are as shown in Figure 3.

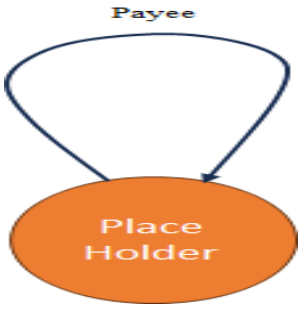


Figure 2. Placeholder node.

Figure 3. Expanding the placeholder node.

3.2. Data pre-processing

In the data pre-processing stage, firstly the number of empty values in each column was determined. Then the if attributes were taken from the data frame. Since the zip codes and zip Merchant columns had the same value for all rows, these columns were deleted. After the nodes with high values in the data set feature network were determined, the step, age, gender, customer, fraud columns were deleted and the data was categorized with One Hot Encoding. Then the feature standardization process was performed. Supervised learning models were trained using internal features and graph-based features. After these processes, since there was a big difference between fake nodes and real nodes, the under-sampling method was applied to the data set in order to balance the data set. After the under-sampling process, the number of fake and real nodes was determined as 7200. In the pre-processing stage, dimension reduction was performed using PCA (Principal Component Analysis) and limiting the number of components to explain 95% of the variance.

3.3. Machine learning and graph algorithms

- Support vector machine (SVM): has gained importance in machine learning and pattern classification [27]. In the case of binary classification, the central idea of the algorithm is to find a hyperplane capable of separating the categories specified by the labels of the dataset. It is assumed that the data are linearly separable and that there may be more than one feasible hyperplane. Based on these assumptions, the algorithm first requires finding the optimal hyperplane. The optimal hyperplane is defined as the one that preserves the most significant distance to the observations in both categories. In this sense, the procedure is as follows [28-29]:
  1. Detect the two closest observations of different categories (support vectors).
  2. Calculate the Euclidean distance between both observations as well as the imaginary line connecting them.

3. Calculate a perpendicular line equidistant from both observations (optimal hyperplane).
4. Calculate the hyperplanes parallel to the optimal hyperplane touching the support vectors. These hyperplanes are defined as the margins.

- Random Forest: algorithm is a collection of tree estimators. In the expression $H(x;\ \theta k), k = 1, \ldots, K, x$ is the associated random vector, θk and k are independent and identically distributed (id) random vectors. When we have a numerical result in the algorithm, we focus on regression tuning, but some special touchpoints are established with classification (categorical result) problems [30].
- XGBoost algorithm: is an optimized version of Gradient Boosting algorithm with various adjustments. It shows good performance with its features such as being able to obtain high predictive power, preventing over-learning, and handling empty data [31]. In the XGBoost algorithm, the first step was to make an initial guess. This value was assumed to be 0.5. Errors were obtained by subtracting the predicted values from the observed value. The aim was to approach the correct guess by learning the errors.
- The K-Nearest Neighbors algorithm: is a nonparametric machine learning technique. In contrast to algorithms that utilize a training set, k-Neighbors employs a training dataset without learning from it; instead, it memorises the data and identifies the nearest neighbors to generate predictions. Initially, a k value is established. The k value is analysed concerning numerous elements equivalent to its magnitude. The Euclidean function computes the distance between the incoming value and the nearest neighbor element. Manhattan, Minkowski, and Hamming metrics can be alternatives to the Euclidean metric. After the distance computation, the ranking is executed, and the incoming value is allocated to the corresponding class [32].
- The PangeRank algorithm: is the most popular graph algorithm that measures the transition effect or connectivity of nodes and was named after Larry Page [33]. This algorithm was used in Neo4j in calculations that depend on the priority of nodes. The Degree algorithm measures the number of relationships connected to a node [34]. In the study, this algorithm is used to determine which node has the most sub nodes. The Community algorithm detects communities in networks based on maximizing the modularity score [35]. In the study, this algorithm is used to find subcommunities in the graph.

4. Results and Discussion

Before processing the graph dataset created in Neo4j, the performance comparison of classical machine learning methods and SVM, XGB, k-NN and RF algorithms was made according to k-fold cross validation. A confusion matrix was used for performance evaluations.

4.1. Confusion Matrix: The confusion matrix is a tool used to evaluate machine learning models addressing classification tasks. It is a square matrix whose dimensions are given by the number of classes or categories. In the case of the binary classification problem of fraud detection in online banking transfers, the confusion matrix has a size of 2 × 2. The value of the confusion matrix is found in the simultaneous recording of the information generated by the predictors or machine learning models and the actual information, remembering that the classification task is primarily addressed through supervised machine learning models. The structure of the confusion matrix is illustrated in Figure 4.

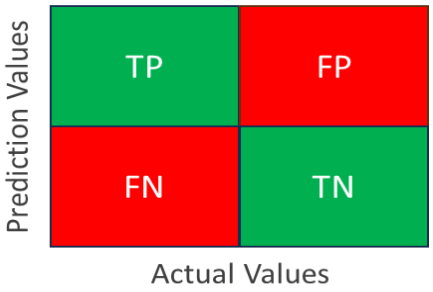


Figure 4. Confusion Matrix.

Where:

- TP (True Positive): Positive instances in reality that were inferred as positive by the machine learning model.
- FN (False Negative): The machine learning model inferred Positive instances in reality as negative.
- FP (False Positive): The machine learning model inferred Negative instances in reality as positive.
- TN (True Negative): Negative instances in reality that were inferred as negative by the machine learning model.
  The metrics selected explicitly for the binary classification problem of banking fraud detection are specified below.

- Accuracy: This metric measures the model's overall performance. Its formula is:

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

- Precision: This metric, also known as Positive Predictive Value (PPV), aims to determine the rate or percentage of instances inferred as positive that are actually positive [36].

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

- Sensitivity (Recall): This metric, referred to as Sensitivity, Hit Rate, or True Positive Rate (TPR), seeks to ascertain the proportion of accurately identified positive instances.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

- F1: This metric, also known as the Harmonic Average of Precision and Sensitivity, aims to combine both metrics that operate on positive instances to create a “harmonic” average from them

$$F1\ score = 2*\frac{Precision*Recall}{Precision + Recall} \quad (4)$$

From a business perspective, Accuracy measures the ratio of detected fraudulent transfers to legitimate transfers incorrectly inferred as fraudulent transfers. The higher the accuracy value, the lower the degree of error. This error has a non-monetary impact, as it means customers are upset about receiving blocked legitimate transfers. Sensitivity measures the ratio of detected fraudulent transfers to fraudulent transfers incorrectly inferred as legitimate. The higher the sensitivity value, the lower the degree of error. This error has an economic impact, as it means undetected fraudulent transfers, the amount of which will eventually have to be absorbed. Therefore, F1 becomes relevant in the detection of fraudulent online banking transfers, since by representing an average of the precision and sensitivity, it allows maintaining a balance and searching for the model that minimizes both simultaneously.

In this study, the values obtained from the confusion matrix of the SVM, XGB, RF, k-NN classifier models are as seen in Figure 6. According to this Figure 6, the precision and F1 score

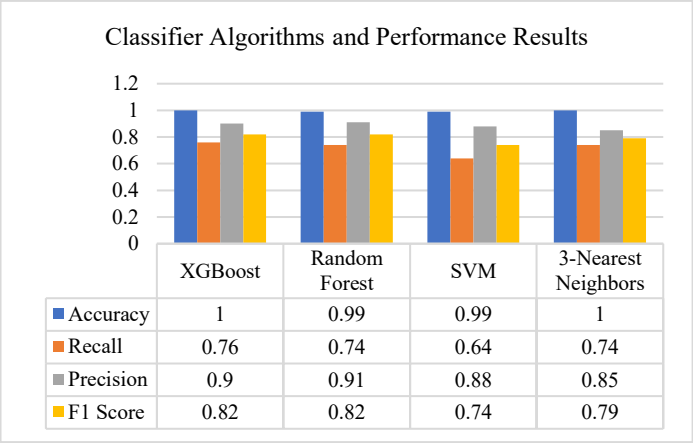
values of the XGB, RF, SVM and k_NN algorithms are quite similar. These results are the results obtained on the unbalanced data set, i.e., before the under-sampling method was applied. The results obtained for the RF, XGB, SVM and k-NN algorithm classifiers as a result of the application of the under-sampling method were compared with K-fold cross-validation. The obtained results are as seen in Figure 7.

4.2. The K-fold cross-validation: is a robust technique for evaluating machine learning model performance. It works by dividing the dataset into k equal subsets (folds). During k iterations, each fold serves as the test set once, while the remaining $k-1$ folds form the training set. The model's performance metrics (e.g., error rate) are averaged across all iterations, yielding a reliable estimate of generalization error. In Figure 7, as a result of the under-sampling application of the XGB, RF, SVM and k-NN algorithms on the data set, the data set was balanced and more consistent values were obtained for the precision and F1 score metrics. After connecting to the Neo4j database via Python, the data nodes in the database were pulled with CypherQL. The pre-processing steps shown in the previous sections were performed on this data. After pre-processing, graph algorithms were applied to determine the main canter and degree heights of nodes such as PageRank, Community, Degree on the data set. Finally, the performances of the applied XGB, RF, SVM and k-NN nearest neighbour algorithms and the extended graph dataset were compared using 5-fold cross-validation. The results are as seen in Figure 8. The F1 score and precision values of the RF and k-NN algorithms were optimized, albeit low. A small performance improvement was observed in the sensitivity (recall) value for the XGB algorithm. In addition, the training and prediction times of the method applied to the graph dataset with the classical machine learning method and graph algorithms for RF were compared. The result is as seen in Figure 5.

```
---------time evaluation----------
training_time_std: 50.490270488262176
training_time_enh 49.02800601005558:
pred_time_std 1.3038832473754878:
pred_time_enh 1.2147937798500064:
```

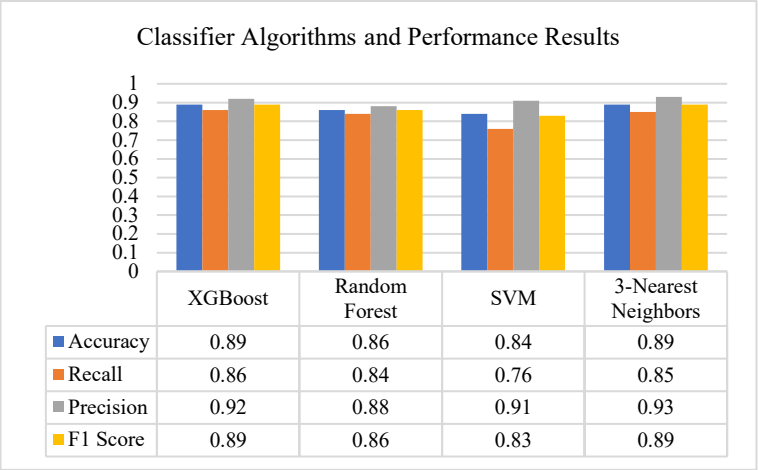
Figure 5. Prediction and training time.

Training_time_std and pred_time_std represent the training and prediction times with standard machine learning algorithms. Training_time_enh and pred_time_enh represent the training and prediction times when extended graph analysis algorithms and standard machine learning algorithms are used together. Although small, the combined use of extended graph analysis and classical machine learning methods shortened the training and prediction times.



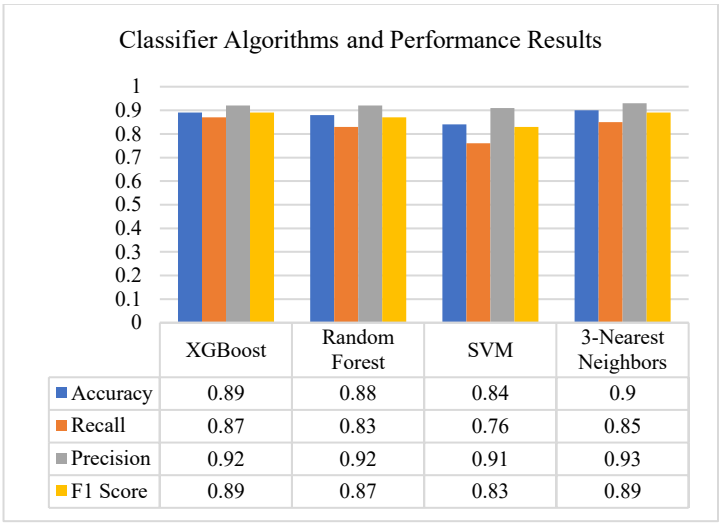
	XGBoost	Random Forest	SVM	3-Nearest Neighbors
Accuracy	1	0.99	0.99	1
Recall	0.76	0.74	0.64	0.74
Precision	0.9	0.91	0.88	0.85
F1 Score	0.82	0.82	0.74	0.79

Figure 6. Performance Results and Classifier Algorithms.



	XGBoost	Random Forest	SVM	3-Nearest Neighbors
Accuracy	0.89	0.86	0.84	0.89
Recall	0.86	0.84	0.76	0.85
Precision	0.92	0.88	0.91	0.93
F1 Score	0.89	0.86	0.83	0.89

Figure 7. Performance results and classifier algorithms (under sampling method).



	XGBoost	Random Forest	SVM	3-Nearest Neighbors
Accuracy	0.89	0.88	0.84	0.9
Recall	0.87	0.83	0.76	0.85
Precision	0.92	0.92	0.91	0.93
F1 Score	0.89	0.87	0.83	0.89

Figure 8. Graph Mining and ML Performance Results, And Classifier Algorithms.

Figure 6 presents the outcomes from the unbalanced dataset utilising solely machine learning techniques. In contrast, the dataset employs only machine learning methods after its balancing through the under-sampling approach Figure 7. The results Figure 8 of the machine learning methods applied after determining the node degrees by applying graph algorithms such as Community, PageRank, and Degree on the nodes of the dataset, visualised with the Neo4j tool. When Figures 7 and 8 are compared, a small improvement is observed in the RF and k-NN precision values. In addition, the XGB and RF sensitivity and F1 score values also showed small improvements. It is known that even very small improvements are important for corporations in terms of providing profit to the company [20].

Therefore, even minor improvements are important for corporations. In order to observe the improvements made more clearly, the results obtained with the results of the studies in [18] and [19] examined in the literature review section are compared in Table 2. Precision, sensitivity, and F1 score values are given for each algorithm.

Table 2. Comparison of the results obtained with other studies.

Reference	k-NN	RF	XGB	SVM	Metrics [Precision (p)/Recall (R)/Accuracy (A) /F1)
[18]	✓	✓	✓	✗	k-NN: P:0.83, R:0.61, F1:0.70 XGB: P:0.89, R:0.76, F1:0.82 RF: P:0.24, R:0.98, F1:0.82
[19]	✓	✓	✗	✓	k-NN: P:0.80, R:0.80, F1:0.79, A:0.80 RF: P:0.93, R:0.93, F1:0.93, A:0.93 SVM: P:0.76, R:0.77, F1:0.76, A:0.77
Combined ML & Graph Algorithm Results (Current work)	✓	✓	✓	✓	k-NN: P:0.93, R:0.85, F1:0.89, A:0.90 RF: P:0.92, R:0.83, F1:0.87, A:0.88 SVM: P:0.91, R:0.76, F1:0.83, A:0.84 XGB: P:0.92, R:0.87, F1:0.89, A:0.89

Table 2 Lists the results obtained in this study with the results of studies in the literature that use the same data set, the same pre-processing method and only machine learning algorithms. The evaluation metrics for each algorithm and their values are given. It has been observed that the machine learning algorithm method used with graph algorithms for K-NN, XGB and SVM achieves better results.

5. Conclusions

In this study, firstly, the data preprocessing stage was performed on the BankSim data set and it was made ready for machine learning models. Then, the classification of the data set was performed with XGB, RF, SVM, k-NN algorithms. Due to the inconsistent precision and F1 score values caused by the imbalance of the data set, more consistent precision and F1 score values were obtained by applying under sampling on the data set. After creating the data set graph in the Neo4j database, standard machine learning algorithms were applied together with Community, PangeRank and Degree algorithms. It was observed that the results of XGB, RF, k-NN algorithms were optimized with graph algorithms. It is possible to obtain better results and perform easier operations by using machine learning algorithms and graph algorithms together. The Neo4j tool is quite useful in performing these operations and using graph algorithms. It allows the use of data collected from multiple sources, and it also contains various data sets for training purposes and the number of sources is quite high. As a result of the continuous extraction and optimization of new algorithms, the study can be carried out in the future by combining different machine learning algorithms and graph algorithms. In the study, it is possible to make fraud risk estimates using scripts on Neo4j without machine learning algorithms by using only CypherQL. The scripts used here can be specified as a feature. This study demonstrates that integrating graph algorithms with machine learning techniques significantly enhances fraud detection systems, yielding measurable improvements in both computational efficiency and detection performance.

Funding: This research received no external funding.

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Can B, Yavuz AG, Karsligil EM, Guvensan MA. A closer look into the characteristics of fraudulent card transactions. *IEEE Access* **8**, 166095–166109 (2020). https://doi.org/10.1109/ACCESS.2020.3022315

2. Somasundaram A, Reddy US. Risk-based bagged ensemble (RBE) for credit card fraud detection. In: *Proc. Int. Conf. on Innovations in Communication, Information and Computing (ICICI 2017)*, 670–674 (2017). https://doi.org/10.1109/ICICI.2017.8365220

3. Parthasarathy S, Tatikonda S, Ucar D. A survey of graph mining techniques for biological datasets. In: Aggarwal C, Wang H (eds) *Managing and Mining Graph Data*. Adv Database Syst, vol 40. Springer, Boston (2010). https://doi.org/10.1007/978-1-4419-6045-0_18

4. Aggarwal C, Wang H. Graph data management and mining: a survey of algorithms and applications. In: Aggarwal C, Wang H (eds) *Managing and Mining Graph Data*. Adv Database Syst, vol 40. Springer, Boston (2010). https://doi.org/10.1007/978-1-4419-6045-0_2

5. Jazayeri A, Yang CC. Frequent subgraph mining algorithms in static and temporal graph-transaction settings: a survey. *IEEE Trans Big Data* **8**(6), 1443–1462 (2022). https://doi.org/10.1109/TBDATA.2021.3072001

6. Najafabadi MM, Villanustre F, Khoshgoftaar TM, et al. Deep learning applications and challenges in big data analytics. *J Big Data* **2**, 1 (2015). https://doi.org/10.1186/s40537-014-0007-7

7. Sarker IH. Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. *SN Comput Sci* **2**, 420 (2021). https://doi.org/10.1007/s42979-021-00815-1

8. Doğan B. The importance of graph databases in detection of organized financial crimes. In: Bozkuş Kahyaoğlu S (ed) *The Impact of Artificial Intelligence on Governance, Economics and Finance*, vol 2. Springer, Singapore (2022). https://doi.org/10.1007/978-981-16-8997-0_8

9. Lira H, Martí L, Sanchez-Pi N. A graph neural network with spatio-temporal attention for multi-sources time series data: an application to frost forecast. *Sensors* **22**(4), 1486 (2022). https://doi.org/10.3390/s22041486

10. Ye Z, Zhao H, Zhang K, Zhu Y, Wang Z. An optimized network representation learning algorithm using multi-relational data. *Mathematics* **7**(5), 460 (2019). https://doi.org/10.3390/math7050460

11. Ye Z, Zhao H, Zhang K, Zhu Y. Multi-view network representation learning algorithm research. *Algorithms* **12**(3), 62 (2019). https://doi.org/10.3390/a12030062

12. Malik EF, Khaw KW, Belaton B, Wong WP, Chew X. Credit card fraud detection using a new hybrid machine learning architecture. *Mathematics* **10**(9), 1480 (2022). https://doi.org/10.3390/math10091480

13. Bin Sulaiman R, Schetinin V, Sant P. Review of machine learning approach on credit card fraud detection. *Hum-Cent Intell Syst* **2**, 55–68 (2022). https://doi.org/10.1007/s44230-022-00004-0

14. Benchaji I, Douzi S, El Ouahidi B, et al. Enhanced credit card fraud detection based on attention mechanism and LSTM deep model. *J Big Data* **8**, 151 (2021). https://doi.org/10.1186/s40537-021-00541-8

15. Zou H. Analysis of best sampling strategy in credit card fraud detection using machine learning. In: *Proc. 6th Int. Conf. on Intelligent Information Technology (ICIIT 2021)*, 40–44 (2021). https://doi.org/10.1145/3460179.3460186

16. Wang C, Zhu H. Representing fine-grained co-occurrences for behavior-based fraud detection in online payment services. *IEEE Trans Dependable Secure Comput* **19**(1), 301–315 (2022). https://doi.org/10.1109/TDSC.2020.2991872

17. Ogundokun RO, Misra S, Fatigun OJ, Adeniyi JK. Naïve Bayes based classifier for credit card fraud discovery. In: Themistocleous M, Papadaki M (eds) *Information Systems*. LNBIP, vol 437. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-95947-0_37

18. Lopez-Rojas EA, Axelsson S. BankSim: a bank payment simulation for fraud detection research. In: *Proc. 26th European Modeling and Simulation Symposium (EMSS 2014)* (2014).

19. Gorton D. IncidentResponseSim: an agent-based simulation tool for risk management of online fraud. In: Buchegger S, Dam M (eds) *Secure IT Systems*. LNCS, vol 9417. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26502-5_12

20. Kocay W, Kreher DL. *Graphs, Algorithms, and Optimization*. Chapman & Hall/CRC, Boca Raton (2005). https://doi.org/10.1201/9781315272689

21. Khairuddin J, Hiekata K, Maimun A, Siow C. Graph theory and deep learning applications in predicting passenger ship principal design parameters. *SSRN Electron J* (2022). https://doi.org/10.2139/ssrn.4063501

22. Guia J, Soares V, Bernardino J. Graph databases: Neo4j analysis. In: *Proc. Int. Conf. on Enterprise Information Systems*, 351–356 (2017). https://doi.org/10.5220/0006356003510356

23. Hilbert S, Coors S, Kraus E, et al. Machine learning for the educational sciences. *Rev Educ* **9**, e3310 (2021). https://doi.org/10.1002/rev3.3310

24. Lopez-Rojas EA, Axelsson S. A review of computer simulation for fraud detection research in financial datasets. In: *Proc. Int. Conf. on Future Technologies (FTC 2016)*, 932–935 (2016). https://doi.org/10.1109/FTC.2016.7821715

25. imón-Reina S, Rincón M, Martínez-Tomás R. An overview of graph databases and their applications in the biomedical domain. *Database (Oxford)* **2021**, baab026 (2021). https://doi.org/10.1093/database/baab026

26. Hajek P, Abedin MZ, Sivarajah U. Fraud detection in mobile payment systems using an XGBoost-based framework. *Inf Syst Front* **25**, 1985–2003 (2023). https://doi.org/10.1007/s10796-022-10346-6

27. Murty MN, Raghava R. *Support Vector Machines and Perceptrons: Learning, Optimization, Classification, and Application to Social Networks*. Springer, Cham (2016). https://link.springer.com/book/10.1007/978-3-319-41063-0

28. Kumar S, Gunjan VK, Ansari MD, Pathak R. Credit card fraud detection using support vector machine. In: Gunjan VK, Zurada JM (eds) *Proc. 2nd Int. Conf. on Recent Trends in Machine Learning, IoT, Smart Cities and Applications*. LNNS, vol 237. Springer, Singapore (2022). https://doi.org/10.1007/978-981-16-6407-6_3

29. Ileberi E, Sun Y, Wang Z. A machine learning-based credit card fraud detection using the GA algorithm for feature selection. *J Big Data* **9**, 24 (2022). https://doi.org/10.1186/s40537-022-00573-8

30. Alarfaj FK, Malik I, Khan HU, Almusallam N, Ramzan M, Ahmed M. Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms. *IEEE Access* **10**, 39700–39715 (2022). https://doi.org/10.1109/ACCESS.2022.3166891

31. Patange, G. S., Hamad, A. A., Gowda, V. D., Gobinath, S., & Ramya, K. (2026). Wireless Sensor Networks for Real-Time Infrastructure Monitoring and Management Using ML

Approach. In *Integrating Modern Mathematics and Sensor Technologies in Civil Engineering* (pp. 409-434). IGI Global Scientific Publishing.

32. Gajjar A, Kashyap P, Aysu A, Franzon P, Dey S, Cheng C. FAXID: FPGA-accelerated XGBoost inference for data centers using HLS. In: *Proc. IEEE Int. Symp. on Field-Programmable Custom Computing Machines (FCCM 2022)*, 1–9 (2022). https://doi.org/10.1109/FCCM53951.2022.9786085

33. Malini N, Pushpa M. Analysis on credit card fraud identification techniques based on KNN and outlier detection. In: *Proc. 3rd Int. Conf. on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB 2017)*, 255–258 (2017). https://doi.org/10.1109/AEEICB.2017.7972424

34. Sangers A, et al. Secure multiparty PageRank algorithm for collaborative fraud detection. In: Goldberg I, Moore T (eds) *Financial Cryptography and Data Security*. LNCS, vol 11598. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32101-7_35

35. Das, B., Gowda, V. D., Hamad, A. A., Kavitha, B. C., & Kottala, S. Y. (2026). Machine Learning Approaches for Predictive Maintenance in Infrastructure Systems. In *Integrating Modern Mathematics and Sensor Technologies in Civil Engineering* (pp. 321-350). IGI Global Scientific Publishing.

36. Pavel HR, Santra A, Chakravarthy S. Degree centrality algorithms for homogeneous multilayer networks. *arXiv preprint* arXiv:2207.11661 (2022). https://arxiv.org/abs/2207.11661

37. Drakopoulos G, Gourgaris P, Kanavos A. Graph communities in Neo4j: four algorithms at work. *Evolving Syst* **11**, 1–14 (2020). https://doi.org/10.1007/s12530-018-9244-x.