Advance Sustainable Science, Engineering and Technology (ASSET)
Vol. 8, No.3, July 2026, pp. 02603032-01 ~ 02603032-013
ISSN: 2715-4211 DOI: https://doi.org/10.26877/asset.v8i3.3393

ECCFD-GNN: A Novel Risk-Sensitive Graph Neural Network Model for Fraudulent Transaction Detection

Shilpa Srivastava[1], Varuna Gupta[2], Alok Singh Chauhan[3*], Sakshi Kumar[4], Sonia Rani[3]

[1]Centre for AI, Christ University, Bengaluru, Karnataka, India

[2]Christ University, Bengaluru, Karnataka, India

[3]School of Computing Science and Engineering, Galgotias University, Greater Noida 203201, India

[4]School of Computer Applications, Noida Institute of Engineering and Technology, Greater Noida, India

*alok.chauhan@galgotiasuniversity.edu.in

Abstract. The study presents the integration of machine learning techniques for detecting the credit card fraud. Its integration maintains a behavioral profile of cardholders and other parameters like location, frequency, amount etc. resulting in the timely detection of any anomaly from the normal behavior. A novel approach ECCFD-GNN (Enhanced Credit Card Fraud Detection based on Graph Neural networks) is proposed enhancing the performance of fraud detection. The various behavioral indicators taken into consideration are number of months with late payments, the frequency of low payments and the length of the account, which are further combined to a newly introduced feature "risk score". The purpose of risk score is to increase the model's sensitivity to the transactions having complex fraud risks. The approach uses three Graph Neural Network architectures namely KNN graph GNN, Radius Graph GNN and Feature Correlation GNN. The experiment is performed with both the optimizers Adam and RAdam. With Adam optimizers the results show that KNN graph GNN provides better performance when compared on the basis of different evaluation parameters with accuracy 85%, precision 76% recall 70% and F1-score as 73%. The results are improved when tested with RAdam optimizers leading to increased accuracy, precision, recall and F1 score.

Keywords: Graph neural networks, K-Nearest Neighbors, GNNRadius, GNNFDCorrelation

(Received 2026-01-24, Revised 2026-02-27, Accepted 2026-03-04, Available Online by 2026-06-17)

1. Introduction

The increasing growth of digital financial services have increased the financial frauds specially involving the transactions made by credit card. In the recent years there has been a revolution in the financial landscape and although this shift has focused more on better speed and convenience but this has also opened new opportunities to the fraudsters. According to the Nilson report it has been estimated that losses from the credit card fraud will surpass $43billion by 2026. The integrity of the digital financial ecosystem is compromised because of these losses which further adds burden to the consumers as well as financial institutions. Reports by Forbes has also presented a very prominent case in United States(2023) that defrauded banks of more than $20 million by applying for credit cards using deep fake identities all generated by artificial intelligence. As per the European central bank(2023) similar complications were noted in the European Union, where cross-border fraud increased by 12% in 2022, highlighting the problem's global scope. Financial fraud is also becoming more prevalent in India. In 2023 alone, the Reserve Bank of India (RBI) reported over ₹300 crore in credit card and digital banking frauds. An important incident in Delhi in 2022 revealed a network of scammers who applied for credit cards using fake documents, evading traditional fraud detection systems and resulting in losses of over ₹10 crore[1] .

Robust fraud detection mechanisms are more important than ever as millions of new, digitally illiterate users enter the financial system as a result of the Indian government's push for financial inclusion through programs like Digital India and Jan Dhan Yojana. Fraud detection has made extensive use of traditional machine learning (ML) models, such as logistic regression, decision trees, and SVMs. These models, however, frequently fall short of capturing the intricate relationships and interdependencies that define fraud because they are based on static, tabular data. Furthermore, fraud trends are ever-changing, which eventually makes static feature-based systems less effective. Learning over graph-structured data is made possible by Graph Neural Networks (GNNs), which present an appealing solution. GNNs use nodes to represent entities like users, merchants, devices, and transactions, and edges to represent the relationships between these nodes. Higher-order dependencies and subtle behavioral patterns can be captured by GNNs thanks to this structure. In the context of a network of known fraudulent activities, for example, a transaction may seem legitimate in isolation but suspicious. Because GNNs learn from both node attributes and their graph-based context, they are highly effective at detecting such anomalies[2,3].

The literature review conducted by [4] addressed the various challenges being faced while applying the techniques such as large-scale machine learning, big data analytics and cloud computing while detecting fraud in credit card transactions. The authors of [5] have provided a summary of the latest state-of-the-art technologies for use in Graph Neural Networks (GNNs) for financial fraud detection. A graph construction algorithm was proposed on the basis of weighted feature similarity to map the local datasets of financial institutions into a transaction graph representation for model training [6]. The authors of [7] have developed a fresh encoder- decoder based GNNs for capturing the associations and dependencies involved in the credit card transactions. In a study [8] it was analyzed that methods based on Graph Neural Networks is the most effective approach for credit card fraud detection. The application of various cutting-edge preprocessing approaches like Smote-ENN for imbalance class, auto encoder for dimensionality reduction and TOPSIS for ideal feature selection has been suggested in [9].

Further the authors of [10] have proposed stacking ensemble model utilizing SVM(Support Vector machine), KNN (K-nearest neighbors and extreme learning machine(ELM) refining the precision level. For optimizing the ELM parameters the particle swarm optimization has been utilized. In another study [11] the challenge of fraud detection and class imbalance was addressed by proposing a framework following a hybrid model HNN-CUHIT integrating neural network with clustering based under sampling technique on identity and transaction features. The authors of [12] have presented TigerGraph for anomaly detection and Louvain algorithm for finding merchant communities used by the fraudsters.

The application of hierarchical graph attention network (HGAT) in detecting the financial fraud detection has been a part of discussion in [13]. The purpose of [14] is to propose two hybrid approaches first one is integrating Synthetic Minority Oversampling Technique (SMOTE) with one class support vector machine (OCSVM) and another one focussing random undersampling. Further, the two models Light Gradient-Boosting Machine (LightGBM) and Long Short term Memory (LSTM) were used for the analysis of the outputs of the hybrid approach. The authors of [15] have presented a hybrid algorithmic optimization-based

deep learning technique Jellyfish Namib Beetle Optimization Algorithm-SpinalNet (JNBO-SpinalNet) for detecting the fraudulent credit card transaction. In another study [16] the authors have emphasized on the usage of Brown-Bear Optimization (BBO) to improvise the accuracy while identifying the credit card financial fraud. A distributed neural network model (DDNN) is developed in [17] for identifying the frauds in credit card transactions by leveraging credit card transaction that take place between various financial institutions. In a study [18] the authors have channelled the supremacy of Deep Convolutional Neural Networks (DeepConvNet) in collaboration with different optimization techniques like Stochastic Gradient Descent (Sgd), Adaptive Gradient (Adagrad), Adaptive Moment Estimation (Adam) and Root mean squared propagation (Rmsprop). The effectiveness of ensemble model can be witnessed in the study conducted for detecting credit card fraud in imbalanced datasets [19]. The authors [20] propose a novel CCFD (Credit card fraud detection) model incorporating Multi-feature Fusion and Generative Adversarial Networks (MFGAN) leading to integration of static and dynamic behavior data of cardholders to a unified high-dimensional feature space.

To improve the predictions and mitigate unwanted biases in identifying credit card fraud the authors of [21] have proposed three uncertainty quantification (UQ) methods named Monte Carlo dropout, ensemble and ensemble Monte Carlo dropout to detect the card fraud. The experimental results have shown that ensemble is a better approach for capturing uncertainty. The purpose of the study conducted by [22] is to propose an enhanced XGBoost algorithm that tunes the hyperparameters of the algorithm with the help of Bayesian optimization. A new digital twin approach was highlighted in the study [23] to improve the fraud in credit card transactions. The authors [24] highlights the importance of a real time fraud detection framework to look into the issues related to non-stationary changes in transaction and class imbalance.

Using data mining techniques, the authors of [25] proposed a dynamic credit risk assessment model and empirically analyzed data from Iranian banks to show its efficacy. In another recent study the authors compared machine learning and deep learning models for predicting credit risk for credit card users, emphasizing the superiority of deep learning techniques [26].The study [27] has focused on the usage of AI with Natural Language Processing (NLP) in improving the performance of digital product sales services.

The literature review shows how credit card fraud detection techniques have changed over time, moving from conventional machine learning models to more sophisticated techniques like ensemble models, GNNs, and hybrid deep learning approaches. In this paper a risk-aware feature—a calculated risk score based on behavioral metrics like late payments and low payment frequency—is incorporated into the novel framework proposed, ECCFD-GNN (Enhanced Credit Card Fraud Detection using Graph Neural Networks). Three GNN variants are examined in the study to determine how well they capture fraud behavior: KNN Graph GNN, Radius Graph GNN, and Feature-Correlation GNN. This study offers a scalable and explicable solution to a significant worldwide problem by combining risk-driven features, graph-based learning, and multi-perspective evaluation.

2. Methods

In the study, deep learning techniques are applied using the graph neural networks. The purpose of using GNN for detecting fraud is evident from the fact that the financial transaction should not be visualized as an isolated transaction. The different entities are users, merchants, devices, location and time. These entities can be modeled as nodes and edges in a graph depicting a relationship for detecting fraud patterns. The detailed methodology is presented through Figure 1.

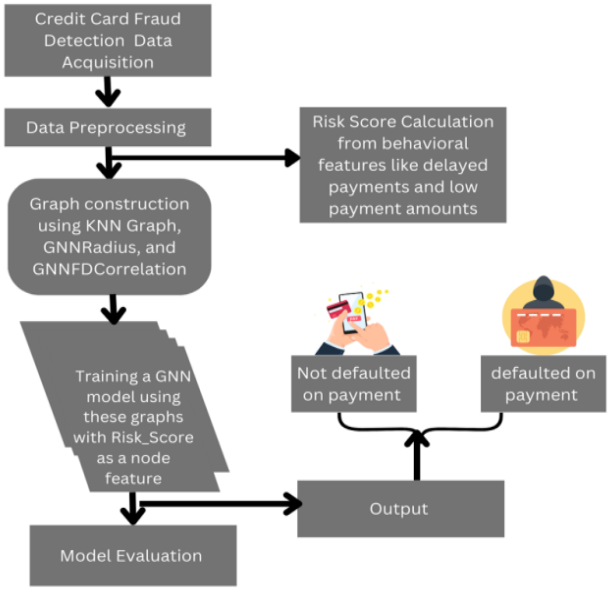

Figure 1. Methodology of the proposed model

The GNNs can aggregate information contained within the local neighborhood of a transaction enabling to identify the larger patterns that may be missed by just looking at a single transaction. GNNs excel at capturing relational, structural, and contextual patterns in non-Euclidean data [24]. Three distinct graph representations—a KNN network, a Radius-based graph, and a Feature-Correlation graph—are created from the preprocessed raw credit card transaction data. A Graph Neural Network (GCN-based) model runs on each graph. GNNs are trained to identify nodes(transactions) as either authentic or fraudulent. Each node refers to a transaction and each transaction contains features about the customer, including their credit behavior and demographics. Thus, the model indirectly learns about customer risk through transaction-level representation. At last, performance of the models is assessed and contrasted with accuracy and other measures.

2.1 Dataset preparation

This step involves the acquisition of the dataset (https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud?resource=download). This dataset has total of 30,001 records with 25 columns. Few columns are comprised of personal information like gender, education status, marital status and age of the customer. One column exhibits the credit limit in numeric form. Rest of the columns provide details of payment history, bill statements and total amount paid in the past six months. The response variable is the feature 'Class', where a legitimate transaction is denoted by a value of 0 and a fraudulent one by a value of 1. The 'Time' feature shows how many seconds passed between a particular transaction and the dataset's first transaction. Cost-sensitive learning scenarios that rely on example-specific costs can make use of the 'Amount' feature, which indicates the transaction's monetary value. The study also includes risk score which is a custom engineered feature that has been designed to quantify the customer's credit risk based on two behavioral features- delayed months of payment and low payment. The purpose of this customized feature is to make the model more efficient and robust. The dataset is divided into 80% training data and 20% testing data.

2.2 Graph Construction

The preprocessed raw data credit card transaction data is used for the construction of the graph. The three distinct representations- a KNN network, a radius based graph and a feature correlation graph GNNFD are

created for further analysis. A GCN (Graph Convolutional Network) based GNN (Graph Neural Network) is applied on each of the three graphs. The GCN layer aggregates information from neighbouring nodes. GNNs are trained to identify nodes and edges. Further, the transactions will be predicted as either authentic or fraudulent. Edge creation method will be different in all the algorithms. An edge is created between nodes i and j only if the distance between their feature vectors is less or greater than a specified threshold value. Once the graph G = (V, E) is constructed using the various methods, GNN will be used for message passing that will be same in all the algorithms (equation 1)

$$h_i^{(l+1)} = \sigma\left(\sum_{\mathrm{J}\in\mathrm{N(i)}} Wij.\mathcal{W}^l h_J^{(l)}\right) \quad (1)$$

Where:

$h_i^{(l+1)}$: embedding of node i at layer l

$\mathcal{W}^l$: learnable weight matrix for layer l

σ: activation function (e.g., ReLU)

$N(i)$: neighbors of node i (within radius)

$h_j^{(l)}$=xi: initial input features

2.2.1 KNN Graph

The graph is constructed by considering the nodes as transactions represented by a feature vector. Each transaction connects to its k most similar neighbors based on feature similarity. Here, euclidean distance is calculated for getting k nearest neighbors. Here, the legitimate transactions will cluster with other legitimate ones, while fraudulent transactions may form their own clusters or link to certain anomalous patterns. For each i, connect to j such that $|x_i - x_j|^2$ is among the k smallest. This ensures that every node has exactly k edges making the graph dense and uniformly connected (equation 2).

$$if, \mathcal{W}_{ij} = \exp\left(-\frac{\|x_i - x_J\|^2}{\sigma^2}\right) < threshold \Rightarrow h_i^{(l+1)} = \sigma\left(\sum_{\mathrm{J}\in\mathrm{N(i)}} Wij.\mathcal{W}^l h_J^{(l)}\right) \quad (2)$$

2.2.2 GNNRadius Graph

In this kind of graph edges are formed on the basis of radius(distance) in a feature space. If the distance between the nodes less than a given radius threshold (say ε) then an edge will be created. In case of credit card transactions nodes can be transactions, users, merchants or devices and the edges formed can take relationships like:- same card used for multiple transactions, same merchant used by different cards, similar time/location/amount or nearby IP addresses or geolocations. In this study the authors have presented transactions as nodes. The benefit of using radius graph is that it is adaptive in nature, providing denser graph in similar behavior clusters and is contextual in nature there by representing nearby transactions showing the fraud patterns (equation 3).

$$if,\ \mathcal{W}_{ij} = \|x_i - x_J\|_2 < r\ \Rightarrow h_i^{(l+1)} = \sigma\left(\sum_{J\in N(i)} Wij.\ \mathcal{W}^l h_J^{(l)}\right) \quad (3)$$

2.2.3 Feature-Correlation Graph

This type of graph is constructed where the nodes are representing features and edges encode the high correlations between the features. This will enable to model interdependencies among features before or during GNN learning. Highly correlated transaction attributes (for example, various bill amounts in consecutive months for a customer) can define a graph structure, transactions then become indirectly related through these feature correlations. The motivation for using this type of graph is the fact that each feature alone may not strongly indicate fraud but if features combined collectively may project potential fraud. For instance, high amount, unusual time and new merchant can be a potential fraud (equation 4).

$$\mathcal{W}_{ij} = corr(x_i - x_J) > P \Rightarrow h_i^{(l+1)} = \sigma\left(\sum_{J\in N(i)} Wij.\ \mathcal{W}^l h_J^{(l)}\right) \quad (4)$$

2.3 Model Training

Every GNN is trained with supervised node categorization. The cross-entropy (negative log-likelihood) between the projected class probabilities and the actual labels for the training nodes defines the training loss. Considering the class imbalance, we use a class weight in the loss function to penalize misclassifying frauds. Each class's weight is determined inversely depending on its training frequency. This enables the model to be attentive to the minority class (fraud) during training. Stochastic gradient descent is run using the Adam optimizer with an initial learning rate of 0.01. We track performance on a held-out validation set to do early halting if required; the models are trained for a maximum of 100 epochs. At every epoch throughout training, we calculate the forecasts for all nodes and assess interim accuracy, precision, recall, and F1-score on the training, validation, and test. Table 1 represents the details of the hyperparameters of the experiment.

Table 1. Hyperparameters

Component	Hyperparameter	Value	Description
Graph Construction	K (KNN)		Number of nearest neighbors
	Radius (GNNRadius)		Correlation threshold
	Threshold (GNNFDCorrelation)		Feature correlation threshold
Model Architecture	Hidden layer 1 Size	32	Neurons in 1st GCN layer
	Hidden layer 2 Size	16	Neurons in 2nd GCN layer
Training	Learning rate	0.01	For Adam optimizer
	Weight decay	5e-4	L2 regularization
	Dropout rate	0.4	Dropout after GCN layers
	Epochs	100	Total training epochs
	Loss function	softmax	Weighted loss

To ensure fairness, we train all three GNN models under identical settings (same initialization scheme, epochs, optimizer schedule, etc.), differing only in the input graph structure. By the end of training, each model yields a set of predicted probabilities for each node in its graph, which we threshold (at 0.5) to assign class labels for evaluation metrics. The model applies cross validation which repeatedly splits data into different training and validation sets. The study uses cross validation to analyze how the GNN based model works well on unseen data. Instead of splitting dataset only once, the graph aware cross validation has been done in this experiment to ensure that there is no leakage of information through shared nodes. The nodes are split into folds to maintain the graph connectivity fully. The model has been trained on the subset of edges and validating on the remaining edges. To evaluate the individual contribution of each architectural and feature component within ECCFD-GNN, a systematic ablation study was conducted. The experiments were designed by removing one component at a time while keeping all other hyperparameters constant. First, the engineered risk score feature was removed to assess its impact. The absence of this feature resulted in a noticeable reduction in recall and F1-score, indicating its effectiveness in enhancing sensitivity toward complex fraudulent transactions. This validates the significance of incorporating behavioral risk indicators into graph-based fraud modeling.

2.4 Evaluation Metrics

The results are evaluated with the metrics accuracy, precision, recall, F1 Score and AUC-ROC curve. The accuracy is the proportion of total correct predictions (both fraud and non-fraud) out of all predictions.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TP is true positives (fraud predicted as fraud), TN is true negatives (non-fraud predicted as non-fraud), FP is false positives (non-fraud predicted as fraud) and FN as false negatives (fraud predicted as non-fraud).

Since fraud datasets are highly imbalanced so relying solely on accuracy can be misleading. It is crucial to access the performance of the model on other metric precision, recall and F1 score as well.

Precision is defined as the fraction of detected frauds that are truly fraudulent (positive predicted value). Higher precision means fewer fraud alarms.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall is the fraction of actual frauds that the model successfully detected (true positive rate). High recall means fewer missed fraud cases. It is Important when missing a fraud case is more dangerous than a false alarm.

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1-score is the harmonic mean of precision and recall, providing a single measure of detection quality that balances false positives and false negatives.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1 score balances precision and recall. It is mostly useful when we need a balance between catching frauds and minimizing false alarms.

3. Results and Discussion

This section provides the results and its analysis. The figure 2 presents the correlation matrix among various parameters. Dark red indicates a strong positive correlation, while dark blue indicates a strong negative correlation. Lighter colors (near white) represent weak or no correlation. According to the correction graph this research has two basic observations like high positive correlation and high negative correlation.

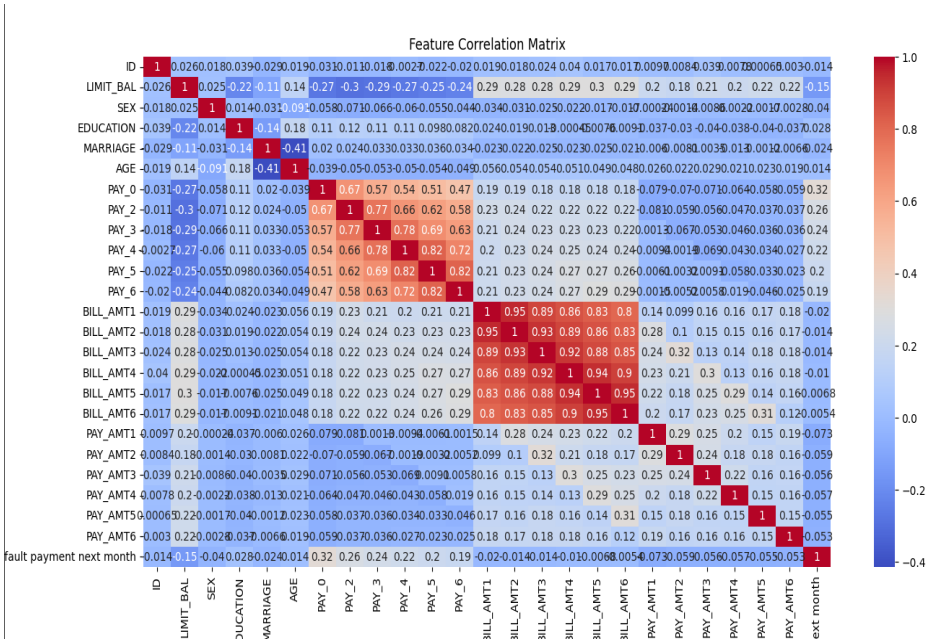

Figure 2. Feature Correlation matrix

3.1 Positive Correlations

BILL_AMT series (e.g., BILL_AMT1, BILL_AMT2, BILL_AMT3, etc.) and PAY_AMT series (e.g., PAY_AMT1, PAY_AMT2, PAY_AMT3, etc.) both series shows very high correlation to each other. It shows that bill amounts and pay amounts across different months are highly correlated, which is expected as bill amounts and pay amount usually follow similar patterns over time. Apart from that PAY_0 and PAY_2show a very high correlation (0.77). This suggests that if a person fails to pay in the current month (PAY_0), there's a strong likelihood that the same person might default again in subsequent months (PAY_2).

3.2 Negative Correlations

PAY_0 and **PAY_AMT6** shows slight negative correlations, suggesting that individuals who have missed payments are less likely to make larger payments (PAY_AMT) in the future.

$$Risk\ Score = \alpha.\left(\frac{No.\ of\ Delayed\ Months}{Total\ Months}\right) + \beta.\left(\frac{No.\ of\ Low\ Payment}{Total\ Months}\right)$$

Where,
Delayed Month = Any Pay_x>0
Low Payment = Any Pay_Amt_x< Median (Pay_Amt_x)
Total Month= Total month of Pay_x or Pay_Amt_x fields
$\alpha\ and\ \beta$ are weighted factors (e.g. $\alpha = 0.6\ and\ \beta = 0.4$)

Using this negative correlation, Risk factor has been calculated to improve the transaction quality and to enhance model sensitivity to transactions with subtle risk indicators. In this dataset risk score is a custom-engineered feature that has been designed to quantify a customer's credit risk based to two already given feature that is delayed months of payment and low payment. Table 2 presents the interpretation of the risk factor range.

Table 2. Interpretation of the risk factor range

Range	Explanation	Risk Severity
0.0 to 1.0	No delayed months & no low payments	Very low risk
1.0 to 3.0	Minor irregularities	Low to moderate risk
3.0 to 5.0	Several delays or consistent low payments	High Risk
5.0 to 6.0	Mostdelayed months & high low payments	Very High Risk

Figure 3 presents the boxplot presenting the distribution of Risk_Score by default status (Y). Defaulters (Y = 1) tend to have higher risk scores, with many values concentrated between 1.5 and 4.0, indicating frequent delayed or low payments. Non-defaulters (Y = 0) mostly have a risk score of 0, meaning they maintained timely and sufficient payments.50% of users have a score of 0, meaning they made timely and adequate payments in the last 6 months. The upper quartile score is 0.66, meaning most scores is skewed toward the low-risk side. On the basis of risk score calculated the performance of the model is evaluated on the different evaluation metrics (accuracy, precision, recall and F1 score).

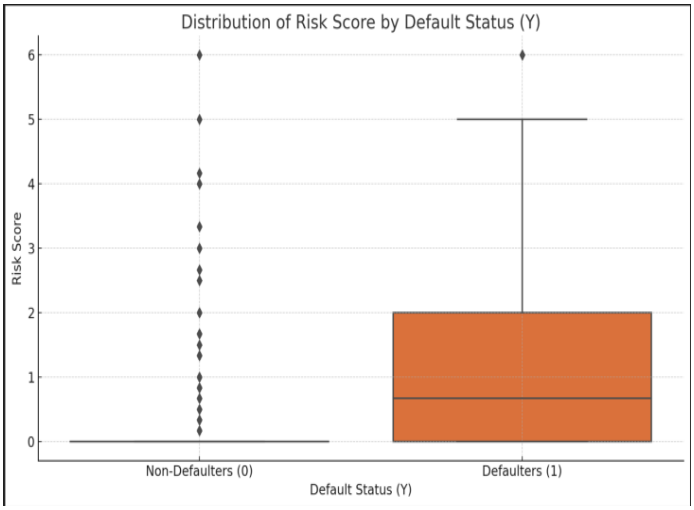

Figure 3. Distribution of risk score

3.3 With different optimizers (Adam and RAdam)

The study uses two optimizers Adam and RAdam for improvising the results. Fraud detection datasets are typically highly imbalanced and noisy. RAdam handles this better by rectifying the variance in the early stages of training, resulting in improved recall and F1-scores. Table 3 presents the comparative analysis of GNN methods with Adam optimizer. Figure 4 depicts the results graphically.

3.3.1 Using Adam Optimizer

Table 3. Comparative analysis of GNN methods with Adam Optimizer

Method	Train Accuracy %	Test Accuracy %	Train Precision %	Test Precision %	Train Recall %	Test Recall %	Train F1%	Test F1%
KNN Graph GNN	93.0	85.0	87.0	76.0	84.0	70.0	85.0	73.0
Radius Graph GNN	91.0	83.0	82.0	74.0	78.0	68.0	80.0	71.0
Feature-Corr GNN	89.0	81.0	79.0	71.0	75.0	65.0	77.0	68.0

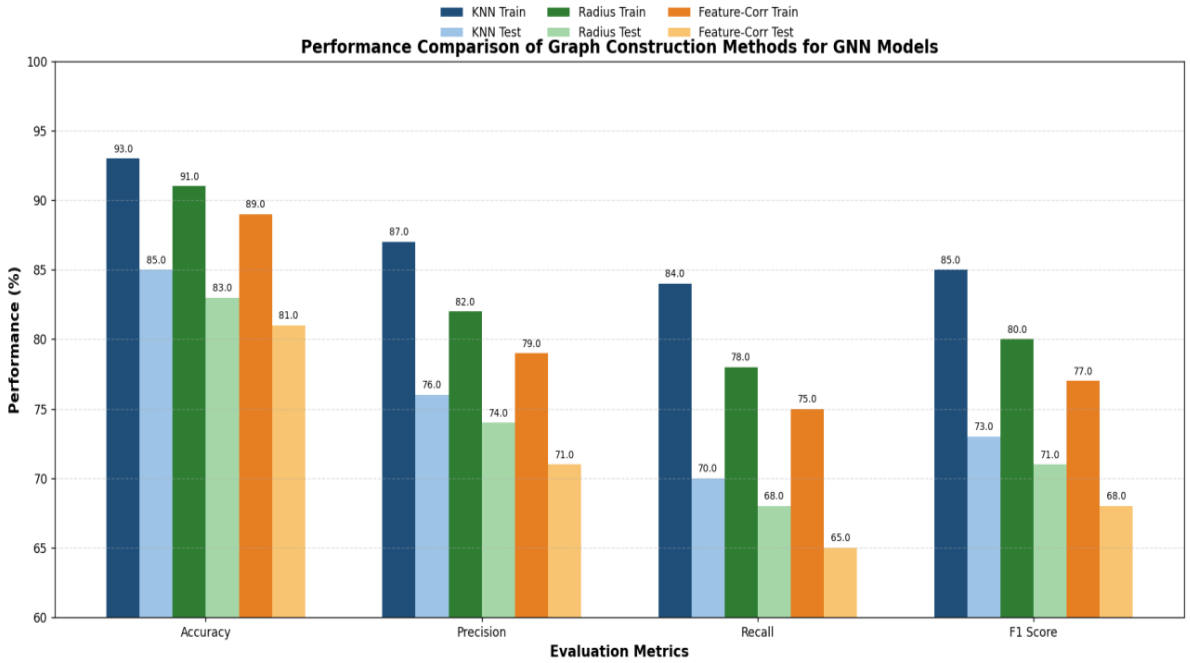

Figure 4. Accuracy, Precision, Recall and F1 comparison of three GNN Methods

GNN models with Adam optimizer show good training accuracy, but the Feature-Corr GNN outperforms the KNN Graph GNN in terms of test accuracy. Figure 4 illustrates that the KNN Graph GNN attains the highest test accuracy of 85%, while the Radius Graph GNN and Feature-Corr GNN achieve 83% and 81%, respectively. The KNN Graph GNN also records the highest test precision (76%) and training precision (87%). Similar patterns can be seen in the Radius Graph GNN and Feature-Corr GNN. The percentage of true positive predictions among all of the model's positive predictions is known as precision. The models do well on training data, but they are less successful on unseen data, as evidenced by a noticeable drop from training to testing. In conclusion we can say that out of all the variants KNN graph GNN outperforms in this experiment. Since the local similarity relationships between the transactions is captured effectively through the KNN Graph GNN which further enables the model to explore the subtle fraud patterns among closely related nodes. It connects the transactions which are having same behavioral features. The structure of KNN improves the detection of anomalous or coordinated fraudulent activities. In both training and testing, the KNN Graph GNN performs the best overall, indicating that it generalizes more effectively than the other models. The KNN Graph GNN is the most successful method according to the evaluation metrics, which include test accuracy, precision, recall, and F1-score. On the other hand, the Feature-Corr GNN performs the worst on tests, especially when it comes to recall and F1-score, which suggests that it is not as good at capturing important features as the other approaches.

3.3.2 Using RAdam Optimizer

Table 4 presents the comparative analysis of GNN methods with RAdam optimizer. The graphical results have been depicted by Figure 5.

Table 4. Comparative analysis of GNN methods with RAdam Optimizer

Method	Train Accuracy %	Test Accuracy %	Train Precision %	Test Precision %	Train Recall %	Test Recall %	Train F1 %	Test F1 %
KNN Graph GNN	95.0	93.5	89.0	86.0	89.0	80.0	88.0	83.0
Radius Graph GNN	93.0	92.0	87.0	84.0	87.0	86.0	86.0	81.0
Feature-Corr GNN	92.0	91.0	89.0	84.0	84.0	85.0	84.0	86.0

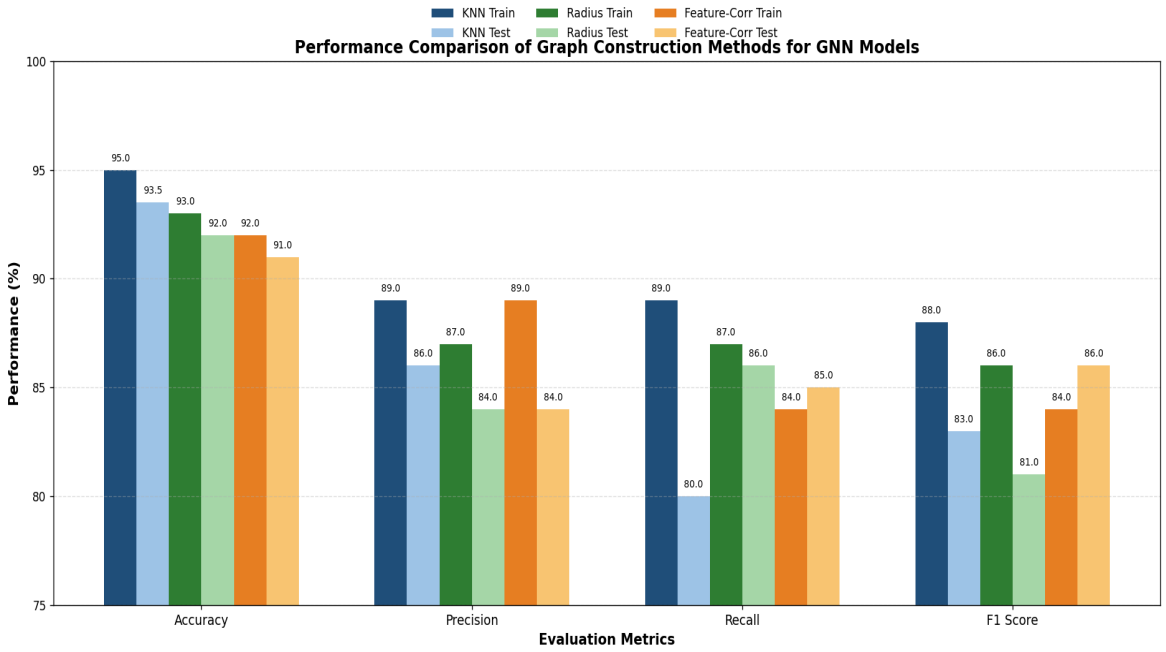

Figure 5. Accuracy, Precision, Recall and F1 of three GNN methods with Radam

It can be analyzed that RAdam performs well overall, there is a consistent drop from KNN → Radius → Feature-Corr, suggesting KNN Graph GNN benefits most from RAdam. RAdam clearly outperforms Adam across all GNN methods, indicating it handles convergence and adaptive learning rates better.

4. Conclusion

This study presents ECCFD-GNN, a graph neural network–based framework developed to enhance credit card fraud detection. To improve the model's capability of identifying subtle fraud patterns, a custom risk-driven feature called the risk score is introduced, which incorporates factors such as transaction duration, low payment counts, and delayed payments. The research further evaluates three different GNN approaches—KNN-GNN, Radius Graph GNN, and Feature-Corr GNN—within the same experimental environment to ensure a consistent comparison. Model performance is assessed using several evaluation metrics, including Accuracy, Precision, Recall, and F1-score, where the nearest neighbor–based approach demonstrates consistently stronger results than the other techniques. In addition, the study examines the impact of two optimizers, Adam and RAdam, and finds that the model performs better when trained with the RAdam optimizer. The relative performance among the three graph-based neural network techniques KNN-CNN,

Radius Graph GNN and Feature-Corr GNN shows a noticeable difference in their efficacy. Maintaining a good balance between accuracy, precision, recall, and F1-score, it implies that it not only learns the patterns efficiently during training but also generalizes well to unseen data. The radius-based graph model shows consistent and dependable outcomes. Although it might not lead in every measure, its general performance stays strong; therefore, it is a reliable option in some important cases. Conversely, especially in its capacity to remember positive occurrences and keep a balanced F1-score, the feature-correlation-based model shows fairly poorer performance. This suggests that in the feature space it could be challenging to catch deeper or more significant associations, which would reduce its efficacy in situations where finding all true cases is essential. Especially in recall-related measures, there is a clear drop from training to testing performance across all models. This emphasizes possible overfitting and points to the requirement for greater fine-tuning using methods. From a societal perspective, these models—especially the nearest neighbor approach—have great promise in helping organizations to more effectively identify fraudulent actions, financial irregularities, or dangerous behavior patterns.

Declaration of AI and AI assisted technologies in the writing process

During the preparation of this work the author(s) used Grammarly, Quillbot and ChatGPT to check grammar, spelling and summarize referenced papers. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors acknowledge the support of Christ University, Bengaluru for funding this research through the Seed Money Project for the session 2024-2025(CU-ORS-SM-24/69.) Their contribution has been instrumental in advancing our work.

References

[1] Bureau TH. Delhi Police arrest 10 persons for inter-State digital fraud. The Hindu 2025.

[2] Akoglu L, Tong H, Koutra D. Graph based anomaly detection and description: a survey. Data Min Knowl Disc 2015;29:626–88. https://doi.org/10.1007/s10618-014-0365-y.

[3] Kipf TN, Welling M. Semi-Supervised Classification with Graph Convolutional Networks 2017. https://doi.org/10.48550/arXiv.1609.02907.

[4] Cherif A, Badhib A, Ammar H, et al. Credit card fraud detection in the era of disruptive technologies: A systematic review. Journal of King Saud University - Computer and Information Sciences 2023;35:145–74. https://doi.org/10.1016/j.jksuci.2022.11.008.

[5] Motie S, Raahemi B. Financial fraud detection using graph neural networks: A systematic review. Expert Systems with Applications 2024;240:122156. https://doi.org/10.1016/j.eswa.2023.122156.

[6] Tang Y, Liang Y. Credit card fraud detection based on federated graph learning. Expert Systems with Applications 2024;256:124979. https://doi.org/10.1016/j.eswa.2024.124979.

[7] Cherif A, Ammar H, Kalkatawi M, et al. Encoder–decoder graph neural network for credit card fraud detection. Journal of King Saud University - Computer and Information Sciences 2024;36:102003. https://doi.org/10.1016/j.jksuci.2024.102003.

[8] Lou C, Wang Y, Li J, et al. Graph neural network for fraud detection via context encoding and adaptive aggregation. Expert Systems with Applications 2025;261:125473. https://doi.org/10.1016/j.eswa.2024.125473.
[9] Gupta RK, Hassan A, Majhi SK, et al. Enhanced framework for credit card fraud detection using robust feature selection and a stacking ensemble model approach. Results in Engineering 2025;26:105084. https://doi.org/10.1016/j.rineng.2025.105084.
[10] Huang H, Liu B, Xue X, et al. Imbalanced credit card fraud detection data: A solution based on hybrid neural network and clustering-based undersampling technique. Applied Soft Computing 2024;154:111368. https://doi.org/10.1016/j.asoc.2024.111368.
[11] Mauliddiah N, Suharjito. Implementation Graph Database Framework for Credit Card Fraud Detection. Procedia Computer Science 2023;227:326–35. https://doi.org/10.1016/j.procs.2023.10.531.
[12] Shi F, Zhao C. Enhancing financial fraud detection with hierarchical graph attention networks: A study on integrating local and extensive structural information. Finance Research Letters 2023;58:104458. https://doi.org/10.1016/j.frl.2023.104458.
[13] Yousefimehr B, Ghatee M. A distribution-preserving method for resampling combined with LightGBM-LSTM for sequence-wise fraud detection in credit card transactions. Expert Systems with Applications 2025;262:125661. https://doi.org/10.1016/j.eswa.2024.125661.
[14] Venkata Krishna Reddy V, Vijaya Kumar Reddy R, Siva Krishna Munaga M, et al. Deep learning-based credit card fraud detection in federated learning. Expert Systems with Applications 2024;255:124493. https://doi.org/10.1016/j.eswa.2024.124493.
[15] Sorour SE, AlBarrak KM, Abohany AA, et al. Credit card fraud detection using the brown bear optimization algorithm. Alexandria Engineering Journal 2024;104:171–92. https://doi.org/10.1016/j.aej.2024.06.040.
[16] Lei Y-T, Ma C-Q, Ren Y-S, et al. A distributed deep neural network model for credit card fraud detection. Finance Research Letters 2023;58:104547. https://doi.org/10.1016/j.frl.2023.104547.
[17] Tekkali CG, Natarajan K. Assessing CNN's Performance with Multiple Optimization Functions for Credit Card Fraud Detection. Procedia Computer Science 2024;235:2035–42. https://doi.org/10.1016/j.procs.2024.04.193.
[18] Wijaya MG, Pinaringgi MF, Zakiyyah AY, et al. Comparative Analysis of Machine Learning Algorithms and Data Balancing Techniques for Credit Card Fraud Detection. Procedia Computer Science 2024;245:677–88. https://doi.org/10.1016/j.procs.2024.10.294.
[19] Xie Y, Li A, Hu B, et al. A Credit Card Fraud Detection Model Based on Multi-Feature Fusion and Generative Adversarial Network. Computers, Materials & Continua 2023;76:2707–26. https://doi.org/10.32604/cmc.2023.037039.
[20] Habibpour M, Gharoun H, Mehdipour M, et al. Uncertainty-aware credit card fraud detection using deep learning. Engineering Applications of Artificial Intelligence 2023;123:106248. https://doi.org/10.1016/j.engappai.2023.106248.
[21] Tayebi M, El Kafhali S. A novel approach based on XGBoost classifier and Bayesian optimization for credit card fraud detection. Cyber Security and Applications 2025;3:100093. https://doi.org/10.1016/j.csa.2025.100093.
[22] Chatterjee P, Das D, Rawat DB. Digital twin for credit card fraud detection: opportunities, challenges, and fraud detection advancements. Future Generation Computer Systems 2024;158:410–26. https://doi.org/10.1016/j.future.2024.04.057.
[23] Charizanos G, Demirhan H, İçen D. An online fuzzy fraud detection framework for credit card transactions. Expert Systems with Applications 2024;252:124127. https://doi.org/10.1016/j.eswa.2024.124127.

[24] Zhou J, Cui G, Hu S, et al. Graph neural networks: A review of methods and applications. AI Open 2020;1:57–81. https://doi.org/10.1016/j.aiopen.2021.01.001.
[25] Moradi S, Mokhatab Rafiei F. A dynamic credit risk assessment model with data mining techniques: evidence from Iranian banks. Financ Innov 2019;5:15. https://doi.org/10.1186/s40854-019-0121-9.
[26] Chang V, Sivakulasingam S, Wang H, et al. Credit Risk Prediction Using Machine Learning and Deep Learning: A Study on Credit Card Customers. Risks 2024;12:174. https://doi.org/10.3390/risks12110174.
[27] Alia PA, Kartika Sari D, Azis N, et al. Implementation Artificial Intelligence with Natural Language Processing Method to Improve Performance of Digital Product Sales Service. ASSET 2024;6:0240301. https://doi.org/10.26877/asset.v6i3.521.