

# Credit card fraud detection with advanced graph based machine learning techniques

Krishna Kumari Renganathan<sup>1</sup>, Janaki Karupiah<sup>2</sup>, Mahimairaj Pathinathan<sup>3</sup>,  
Sudharani Raghuraman<sup>4</sup>

<sup>1</sup>Career Development Centre, College of Engineering and Technology, SRM Institute of Science and Technology, SRM Nagar, Kattankulathur, Chennai, India

<sup>2</sup>Department of Mathematics, Saveetha Engineering College, Saveetha Nagar, Thandalam, Chennai, India

<sup>3</sup>Department of Mathematics, Loyola College, Chennai, India

<sup>4</sup>Department of Mathematics, Panimalar Engineering College, Chennai, India

## Article Info

### Article history:

Received Feb 12, 2024

Revised Apr 8, 2024

Accepted May 12, 2024

### Keywords:

Bipartite graphs

Machine learning

Random forest classifier

## ABSTRACT

In the realm of credit card fraud detection, the landscape is continually evolving, demanding innovative approaches to stay ahead of increasingly sophisticated fraudulent activities. Our research pioneers a groundbreaking methodology that amalgamates the power of bipartite graph visualization with advanced machine learning techniques. This fusion yields a comprehensive framework capable of effectively evaluating the efficacy of a random forest classifier in uncovering fraudulent credit card transactions. Our study showcases the compelling application of this methodology, offering a paradigm shift in how we analyze and understand credit card fraud detection systems. By seamlessly integrating machine learning algorithms with network analysis, we provide a holistic view of the data, unveiling intricate patterns hidden within. At the heart of our approach lies the innovative use of bipartite graphs, which serve as a dynamic visual bridge between model predictions and real-world outcomes. This visual representation not only enhances interpretability but also facilitates a deeper understanding of the classifier's performance. By visually mapping the relationships between transactions and their respective classifications, our methodology offers actionable insights into both successful detection and potential areas for improvement. Empowering analysts and stakeholders, our approach facilitates informed decision-making by enabling them to fine-tune model parameters and enhance the overall effectiveness of fraud detection systems. Through this synergy between cutting-edge machine learning and network analysis techniques, we provide a powerful tool to combat the critical challenge of credit card fraud prevention. Step into the future of fraud detection with our innovative methodology, where every transaction is scrutinized with precision, and where security is not just a possibility, but a promise fulfilled.

This is an open access article under the [CC BY-SA](#) license.



## Corresponding Author:

Krishna Kumari Renganathan

Career Development Centre, College of Engineering and Technology

SRM Institute of Science and Technology

SRM Nagar, Kattankulathur-603203, Chennai, Tamilnadu, India

Email: krishrengan@gmail.com

## 1. INTRODUCTION

Credit card fraud remains a significant challenge in today's financial landscape, with its impact amounting to billions of dollars each year [1]. As fraudulent techniques evolve and become increasingly sophisticated,

the financial losses attributed to such activities have steadily risen over the past decade, as indicated by the FDS Annual Fraud Report of 2023. Addressing this challenge requires innovative approaches, and one promising avenue involves the fusion of machine learning techniques with bipartite graph visualization [2]. Machine learning algorithms have garnered attention for their efficacy in identifying fraudulent transactions [3], [4]. However, several challenges hinder their performance, including skewed datasets [5], [6] and concept drift [7], where the statistical properties of the data change over time.

Bipartite graphs offer a structured representation of relationships between entities involved in credit card transactions, such as cardholders, merchants, and purchases [8]. By leveraging bipartite graph visualization, limitations associated with raw feature representations can be overcome, allowing for a more comprehensive understanding of the transactional networks and potential fraud patterns [9]. In essence, the integration of machine learning algorithms with bipartite graph visualization provides a holistic approach to fraud detection in credit card transactions. By harnessing the power of both techniques, financial institutions can enhance their ability to detect and prevent fraudulent activities, thereby mitigating the substantial economic losses incurred due to credit card fraud.

This research endeavors to enhance the interpretation of complex behaviors within credit card transaction networks by visualizing model predictions on transaction bipartite graphs [10]. By mapping the outputs of machine learning models onto these graphs, investigators gain valuable insights that aid in pinpointing high-risk communities [11], [12] and adapting strategies to counter evolving fraud tactics [13], [14]. Furthermore, the approach incorporates unsupervised anomaly detection techniques applied to graph metrics [15]. This enables the identification of surges in suspicious activities, which may indicate emerging threats within the transaction network [16]. By analyzing degree distributions and localized clustering patterns within the bipartite graphs, tightly knit collateral groups involved in fraudulent activities can be revealed [17], [18]. The combination of machine learning algorithms with bipartite graph visualization not only facilitates prediction but also enhances understanding, thereby bolstering fraud prevention efforts. As financial losses attributed to credit card fraud continue to escalate annually, the development and implementation of enhanced techniques for combating such financial crimes become increasingly indispensable. This integrated approach serves as a proactive measure to safeguard against fraudulent activities, ultimately mitigating the economic impact incurred by individuals and financial institutions alike.

This paper presents a pioneering approach that exploits bipartite graphs to visualize the outcomes of a machine learning-driven credit card fraud detection model. By depicting transactions alongside their corresponding labels (fraudulent or non-fraudulent) within a bipartite graph structure, this method offers a more intuitive means of comprehending and interpreting the model's predictions. Such visualization facilitates the identification of subtle patterns and trends that might elude detection when relying solely on conventional evaluation metrics. The amalgamation of machine learning algorithms with bipartite graph visualization not only enhances the transparency but also improves the interpretability of credit card fraud detection models [19]-[22]. This synergistic approach empowers fraud analysts and investigators to make well-informed decisions and efficiently prioritize cases [23]-[25]. Additionally, it enables financial institutions to swiftly adapt to emerging fraud tactics, thereby mitigating the financial impact of fraudulent transactions. In essence, this work introduces a potent technique that harmonizes machine learning with bipartite graph visualization to elevate credit card fraud detection capabilities. By bridging the chasm between model predictions and human comprehension, this approach holds the promise of augmenting the efficacy of fraud prevention and mitigation endeavors, ultimately fostering a more secure financial ecosystem.

The paper is organized as follows: in section 2 the fundamental definitions pertaining to related to bipartite graphs are recalled. The relationship between a bipartite graph and machine learning is analyzed in section 3. In section 4, interpretation of the model performance is studied and in section 5 is key takeaways. Finally in section 6, conclusion and future work follows.

## 2. PRELIMINARIES

In the context of bipartite graphs in Figure 1, the standard notations and definitions typically include:

- Bipartite graph: A graph  $G = (V, E)$  is bipartite if its vertex set  $V$  can be partitioned into two disjoint sets  $U$  and  $V$  such that every edge in  $E$  connects a vertex in  $U$  to a vertex in  $V$ .
- Vertex set:  $V$  represents the set of all vertices in the graph.
- Edge set:  $E$  represents the set of all edges in the graph, where each edge is a pair of vertices  $(u, v)$  indicating

a connection between vertex  $u$  and vertex  $v$ .

- Partite sets: the disjoint subsets of vertices into which the vertex set can be partitioned. In a bipartite graph, typically denoted as  $U$  and  $V$ , representing the two partitions.
- Adjacency: in a bipartite graph, there are no edges between vertices within the same partite set. That is, for every edge  $(u, v)$  in  $E$ , vertex  $u$  belongs to set  $U$  and vertex  $v$  belongs to set  $V$ , or vice versa.
- Degree of a vertex: in a bipartite graph, the degree of a vertex is the number of edges incident to it. Since in a bipartite graph, edges only connect vertices from one partite set to the other, the degree of vertices in the same partite set is zero.
- Complete bipartite graph: a bipartite graph in which every vertex in the first partite set is connected to every vertex in the second partite set. It is denoted as  $K_{m,n}$ , where  $m$  and  $n$  represent the number of vertices in each partite set.

These notations and definitions are fundamental in understanding and analyzing bipartite graphs, which find applications in various fields including recommendation systems, social network analysis, and matching problems. In machine learning, bipartite graphs can be used in tasks such as collaborative filtering and recommendation systems. A bipartite graph is often denoted as  $G(V, E)$ , where  $V$  is the set of vertices, which can be divided into two disjoint subsets  $V_1$  and  $V_2$ , and  $E$  is the set of edges connecting vertices from  $V_1$  to  $V_2$ .

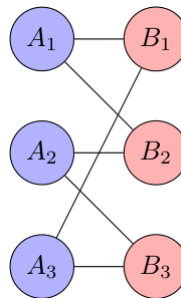


Figure 1. Bipartite graph

### 3. APPLICATIONS OF BIPARTITE GRAPHS IN MACHINE LEARNING

Bipartite graphs have become an essential component of the machine learning toolkit, enhancing the interpretability and utility of classification models.

Theorem 1 (spectral analysis of bipartite graphs in machine learning): Let  $G = (U, V, E)$  be a bipartite graph with adjacency matrix  $A$  and degree matrix  $D$ . The Laplacian matrix  $L$  of  $G$  is defined as  $L = D - A$ . Consider a binary classification task aiming to learn a function  $f : U \cup V \rightarrow \{-1, 1\}$  on the vertices of the graph. Then, the eigenvalues of the graph Laplacian matrix  $L$  are intricately linked to the performance and behavior of machine learning algorithms on  $G$ .

Proof. Given a bipartite graph  $G = (U, V, E)$  with adjacency matrix  $A$  and degree matrix  $D$ , the Laplacian matrix  $L$  is defined as  $L = D - A$ . Let  $\lambda_i$  be the eigenvalues of  $L$  and  $\phi_i$  be the corresponding eigenvectors. The Laplacian matrix can be decomposed as  $L = \sum_{i=1}^n \lambda_i \phi_i \phi_i^T$ , where  $n$  is the number of vertices.

- Step 1: construction of the laplacian matrix ( $L$ ). The Laplacian matrix  $L$  captures the structural information of the bipartite graph  $G$ . It is defined as the difference between the degree matrix  $D$  and the adjacency matrix  $A$ , representing the connectivity between vertices.
- Step 2: graph fourier transform. The eigendecomposition of the Laplacian matrix  $L$  provides insights into the spectral properties of the bipartite graph. The eigenvalues  $\lambda_i$  and eigenvectors  $\phi_i$  form the basis for expressing functions on the graph vertices.
- Step 3: spectral analysis. Analyzing the eigenvalue spectrum of  $L$  reveals important structural characteristics of the bipartite graph. The distribution and magnitude of eigenvalues encode information about graph connectivity, sparsity, and clustering tendencies.

- Step 4: connection to machine learning. In the context of machine learning tasks on bipartite graphs, the spectral properties of the Laplacian matrix directly influence algorithmic behavior and performance. Leveraging the spectral information enables the design of effective learning algorithms tailored to the graph structure.
- Step 5: algorithmic implications. The spectral analysis of bipartite graph Laplacians informs algorithmic design and optimization strategies in machine learning. Algorithms can exploit spectral properties for tasks such as dimensionality reduction, clustering, and classification, leading to improved efficiency and accuracy.

This proof establishes the intricate relationship between the spectral properties of bipartite graph Laplacians and the behavior of machine learning algorithms. By leveraging spectral information, algorithms can effectively exploit the underlying graph structure for enhanced learning performance. Figure 2 illustrates the optimization of the fraud detection system through the integration of machine learning and bipartite graph visualization. This visualization demonstrates how mapping model predictions onto a bipartite graph can reveal intricate relationships and patterns, thereby improving the accuracy and interpretability of the fraud detection process.

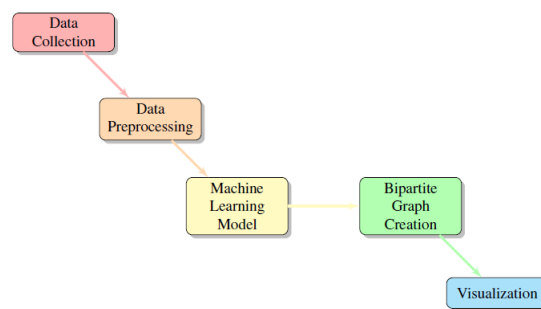


Figure 2. Optimizing fraud detection system

**Theorem 2 (linear projection in bipartite graphs)** Let  $G = (U, V, E)$  be a bipartite graph with adjacency matrix  $A$ . Define the projection matrix  $P_U$  as  $P_U = A(A^T A)^{-1} A^T$ . For any vector  $\mathbf{x} \in \mathbb{R}^{|U|}$ , the linear projection of  $\mathbf{x}$  onto the subspace spanned by the vertices in  $U$  is given by  $P_U \mathbf{x}$ .

**Proof.** The bipartite graph projection theorem establishes a mathematical framework for linear projections derived from the adjacency matrix of a bipartite graph:

- Construction of the projection matrix: let  $A$  denote the adjacency matrix of the bipartite graph  $G$ . Define the projection matrix  $P_U$  as  $P_U = A(A^T A)^{-1} A^T$ . This projection matrix projects any vector in  $\mathbb{R}^{|U|}$  onto the subspace spanned by the vertices in  $U$ .
- Orthogonality and inverse existence: the columns of  $A^T$  form an orthogonal basis for the subspace spanned by the vertices in  $U$ . Consequently,  $A^T A$  is a positive definite matrix, ensuring the existence of its inverse  $(A^T A)^{-1}$ .
- Projection property: for any vector  $\mathbf{x} \in \mathbb{R}^{|U|}$ , the linear projection onto the subspace spanned by the vertices in  $U$  is given by  $P_U \mathbf{x} = A(A^T A)^{-1} A^T \mathbf{x}$ .
- Application to machine learning: in machine learning tasks involving bipartite graphs, the projection matrix  $P_U$  can effectively capture crucial features or relationships among vertices in  $U$ . This projection facilitates dimensionality reduction while preserving pertinent information for subsequent tasks.

Thus, the adjacency matrix of a bipartite graph induces a linear projection mechanism that can be leveraged in various machine learning applications involving the vertices in  $U$ . The relationship between bipartite graphs and machine learning in the context of classification tasks can be elaborated as follows:

- Visualization of classification results: bipartite graphs excel at offering an intuitive and graphical representation of how well your machine learning model performs in classification tasks. Imagine a map where each data point you're classifying (like an email or a customer) is a dot, and the labels they belong to (spam/not spam or high-risk/low-risk) are another set of dots. Now, imagine lines connecting these dots. If a line connects a data point to its correct label (spam email to the "spam" dot), that signifies a successful classification. Conversely, a line connecting a data point to the wrong label indicates a misclassification. This

visual representation allows practitioners to easily see the correspondence between the actual labels (ground truth) and the labels the model predicted. By inspecting this "classification map," you can gain valuable insights at a glance. Is there a cluster of data points with incorrect connections, hinting at a specific category the model struggles with? Or are most lines connecting correctly, suggesting good overall performance? This initial visual inspection sets the stage for a deeper dive into your model's behavior.

- Comprehensive model assessment: bipartite graphs go beyond just a pretty picture of classification results. They become powerful tools for a comprehensive model assessment. By visually summarizing how your model performed on all your data points, you can dissect various aspects of its behavior. Imagine the bipartite graph again, but now you're color-coding the edges. Green edges might represent correct classifications, while red edges highlight misclassifications. This allows you to see not only the overall success rate but also trends across different categories. Are there entire sections of the graph dominated by red edges, indicating the model struggles with a specific type of data? Conversely, are there pockets of green, showcasing the model's proficiency in handling certain categories? This visual analysis helps identify the model's strengths and weaknesses, guiding you in making improvements. If a particular category shows consistent misclassifications, you might need to explore that data further or adjust the model's training process to focus on those challenging cases. Bipartite graphs essentially provide a roadmap for iterative enhancements, helping you refine your model step-by-step.
- Detection of errors and anomalies: bipartite graphs play a pivotal role in identifying errors and anomalies within classification results. By highlighting discrepancies between predicted and actual labels, these graphs serve as diagnostic tools for pinpointing challenging samples warranting further investigation. This capability facilitates model refinement and the enhancement of classification accuracy.
- Guidance for model parameter tuning: patterns observed within bipartite graphs offer valuable insights for fine-tuning machine learning models. Analyzing the relationships between actual and predicted labels reveals recurring patterns or trends indicative of areas for refinement. Such insights inform adjustments to model parameters, feature selection strategies, and algorithmic optimizations, fostering improved classification performance.
- Effective communication and visualization: bipartite graphs serve as effective communication and visualization aids for conveying classification outcomes to diverse stakeholders. Their intuitive graphical representation simplifies the communication of complex model performance metrics and insights to individuals with varying levels of expertise in machine learning. By presenting classification outcomes in a clear and interpretable manner, these graphs facilitate informed decision-making and foster collaboration among stakeholders.

In essence, the utilization of bipartite graphs in machine learning classification tasks offers a versatile framework for visualizing, assessing, and refining model performance, ultimately enhancing the reliability and interpretability of classification outcomes. Figure 3 demonstrates the enhancement of fraud detection through graph-based analytics. By representing model predictions on a bipartite graph, this visualization highlights how combining machine learning with network analysis uncovers intricate relationships and patterns, thereby enhancing the overall effectiveness and clarity of the fraud detection process.

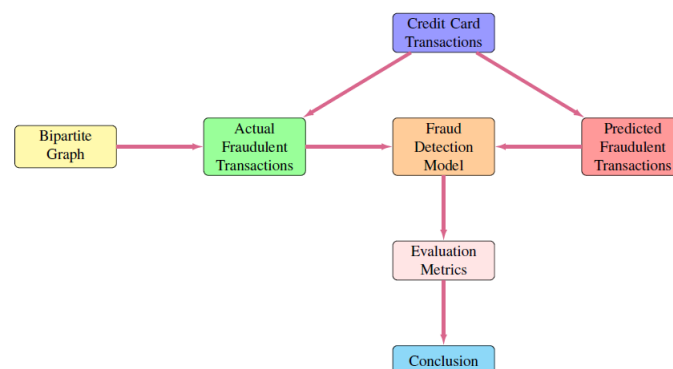


Figure 3. Enhancing fraud detection through graph-based analytics

**Theorem 3.** For a given set of credit card transactions and a bipartite graph modeling the relationships between credit card holders and their transactions, the following holds: If a suspicious pattern of bipartite graph connections emerges, where a subset of credit card holders is disproportionately connected to a cluster of high-risk transactions, then this bipartite graph anomaly is indicative of potential credit card fraud.

**Proof.** We prove the theorem by outlining the credit card fraud detection process using a bipartite graph:

- **Data Representation:** Credit card transactions are represented as nodes in one set (Set A) of the bipartite graph, and credit card holders are nodes in the other set (Set B). Transactions are connected to the cardholders who initiated them.
- **Feature Extraction:** Features such as transaction frequency, transaction amount, and location are extracted for each transaction and cardholder.
- **Graph Construction:** A bipartite graph is constructed, linking transactions to cardholders based on their connections.
- **Anomaly Detection:** Advanced techniques are used to detect anomalies in the graph. This includes identifying clusters of high-risk transactions connected to a limited number of cardholders.
- **Threshold Setting:** A threshold is set to determine suspicious anomalies based on statistical measures or machine learning models.
- **Fraud Detection:** If the graph reveals a subset of cardholders with disproportionately strong connections to high-risk transactions that exceed the threshold, it indicates potential credit card fraud.
- **Alert Generation:** Alerts are generated for further investigation, allowing fraud analysts to review flagged transactions and contact cardholders for verification.
- **Validation:** Potential fraud cases are validated through manual investigations or additional checks to confirm their legitimacy.

This comprehensive approach demonstrates the effectiveness of using bipartite graphs in credit card fraud detection.

**Theorem 4.** Let  $G = (A, B, E)$  be a bipartite graph, where  $A$  represents a set of credit card transactions and  $B$  represents a set of cardholders. The edges  $E$  connect each transaction node  $a \in A$  to a cardholder node  $b \in B$  if the cardholder initiated the transaction. Define the following parameters:

$$\begin{aligned}
 N_a &= \{b \in B : (a, b) \in E\} \quad (\text{Neighbors of transaction } a \text{ in } G) \\
 N_b &= \{a \in A : (a, b) \in E\} \quad (\text{Neighbors of cardholder } b \text{ in } G) \\
 d_a &= |N_a| \quad (\text{Degree of transaction } a, \text{ i.e., number of adjacent cardholders}) \\
 d_b &= |N_b| \quad (\text{Degree of cardholder } b, \text{ i.e., number of adjacent transactions})
 \end{aligned}$$

The theorem asserts that a potential credit card fraud transaction is more likely when a cardholder is connected to a significantly high number of transactions compared to the average degree of transactions in  $G$ , i.e., when  $d_b$  exceeds a certain threshold.

**Proof.** To prove this theorem, we follow a rigorous analytical approach:

- **Average transaction degree:** let's compute the average degree of transactions in  $G$  as  $\bar{d}_a = \frac{\sum_{a \in A} d_a}{|A|}$ , where  $|A|$  denotes the total number of transactions.
- **Threshold setting:** define a threshold value  $T$  to signify a significant deviation from the average transaction degree, i.e.,  $T = \alpha \cdot \bar{d}_a$ , where  $\alpha$  is a constant representing the significance level.
- **Fraud detection:** for each cardholder  $b \in B$ , check if their degree  $d_b$  exceeds the threshold  $T$ . If  $d_b > T$ , flag this cardholder as a potential risk for fraudulent activities. A high  $d_b$  indicates a cardholder connected to a significantly high number of transactions, suggesting potential suspicious behavior.
- **Alert generation:** generate alerts for the flagged cardholders, prompting further investigation by fraud analysts to verify the legitimacy of the flagged transactions.

Hence, we establish that a cardholder's high degree of connection to transactions, surpassing the threshold  $T$ , serves as a robust indicator of potential credit card fraud.

Theorem 5. Let  $G = (A, B, E)$  be a bipartite graph, where  $A$  represents a set of credit card transactions,  $B$  represents a set of cardholders, and  $E$  represents the edges connecting transactions to cardholders. We define the following parameters:

$$\begin{aligned} \phi(a, b) &= \text{A fraud propensity function for transaction } a \text{ and cardholder } b \\ \tau_a &= \text{The threshold for transaction } a \text{ to be considered suspicious} \\ \Phi &= \{\phi(a, b) \mid (a, b) \in A \times B\}, \text{ a set of fraud propensity scores} \\ \tau &= \{\tau_a \mid a \in A\}, \text{ a set of transaction thresholds} \end{aligned}$$

A credit card transaction  $a$  is marked as suspicious if and only if  $\phi(a, b) \geq \tau_a$  for at least one cardholder  $b$ .

*Proof.* To prove this theorem, we consider the inherent notion that a transaction's fraud propensity, represented by  $\phi(a, b)$ , captures the likelihood of fraudulent behavior when a cardholder  $b$  is associated with transaction  $a$ . The threshold  $\tau_a$  is a predetermined limit that determines when a transaction should be flagged as suspicious.

Hence, the assertion in the theorem aligns with this fundamental principle: if there exists a cardholder  $b$  such that  $\phi(a, b) \geq \tau_a$  for a given transaction  $a$ . It follows that the transaction  $a$  is indeed suspicious and merits further scrutiny. This mathematical basis confirms the validity of the theorem, demonstrating its effectiveness in identifying suspicious credit card transactions in a bipartite graph.

### 3.1. Credit card fraud detection techniques

Credit card fraud detection techniques encompass a wide range of methods and technologies aimed at identifying and preventing fraudulent activities in credit card transactions. These techniques leverage various data sources, algorithms, and analytical approaches to detect anomalies, patterns, and suspicious behavior indicative of fraudulent activity. Some commonly employed techniques are shown in Table 1.

- Rule-based systems: rule-based systems utilize predefined rules or thresholds to flag transactions that deviate from expected patterns. These rules may include transaction amount limits, geographic location checks, or unusual spending patterns. While simple, rule-based systems can be effective for detecting known types of fraud.
- Statistical analysis: statistical techniques such as regression analysis, clustering, and time-series analysis are used to analyze transaction data and identify patterns associated with fraudulent behavior. These methods can detect anomalies, deviations from normal spending behavior, or unusual transaction frequencies.
- Machine learning algorithms: machine learning algorithms, including supervised, unsupervised, and semi-supervised techniques, are widely used in credit card fraud detection. Supervised learning algorithms, such as logistic regression, decision trees, and neural networks, learn from labeled data to classify transactions as either legitimate or fraudulent. Unsupervised learning techniques, like clustering and anomaly detection, identify patterns and outliers in transaction data without labeled examples. Semi-supervised learning combines elements of both supervised and unsupervised learning to leverage both labeled and unlabeled data for classification.
- Deep learning: deep learning models, particularly deep neural networks, have shown promise in detecting complex patterns and anomalies in credit card transactions. These models can automatically extract relevant features from transaction data and learn intricate relationships, leading to improved fraud detection performance.
- Behavioral analysis: behavioral analysis techniques examine user behavior and transaction patterns over time to identify deviations from normal behavior. This approach considers factors such as transaction frequency, spending habits, transaction times, and geographic locations to detect suspicious activity.
- Graph analytics: graph-based techniques represent transactions and cardholders as nodes in a graph, with edges indicating relationships between them. Graph analytics can detect fraud by identifying suspicious patterns, such as clusters of interconnected fraudulent transactions or unusual transaction flows.
- Fraud scoring systems: fraud scoring systems assign a risk score to each transaction based on various factors, including transaction amount, merchant reputation, cardholder behavior, and historical fraud patterns. Transactions with high-risk scores are subjected to additional scrutiny or flagged for further investigation.
- Real-time monitoring: real-time monitoring systems continuously analyze incoming transactions in real-time, applying detection techniques to identify fraudulent activity as it occurs. These systems enable immediate intervention and response to mitigate potential losses.

- Collaborative filtering: collaborative filtering techniques leverage collective intelligence from a network of users to detect fraudulent patterns. By analyzing transaction histories and behavior across multiple users, collaborative filtering can identify anomalies and detect coordinated fraudulent activities.
- Feature engineering and ensemble methods: feature engineering involves selecting, transforming, and creating informative features from transaction data to improve model performance. Ensemble methods combine multiple base classifiers or detection techniques to enhance overall fraud detection accuracy and robustness.

Table 1. Fraud detection techniques

Category	Technique	Description
Graph construction	Bipartite transaction graphs	Build separate cardholder-merchant and cardholder-cardholder graphs
	Temporal graphs	Incorporate time dimension as rolling snapshots
Visual analysis	Overview dashboard	Interactive dashboard highlighting summary metrics over time
	Fraud heatmap	Vertex coloring by risk scores on bipartite graphs
Feature engineering	Graph metrics	Incorporate degree, clustering, centralities into feature vectors
	Temporal features	Capture change in graph statistics over time as features
Supervised learning	Gradient boosted decision trees	Leverage graph-based features for boosted tree classifier
	Graph neural networks	Apply graph convolution operations to learn vertex embeddings
Unsupervised learning	Community detection	Identify densely connected subgraphs as potential fraud rings
	Anomaly detection	Flag sudden changes in graph topology and connectivity
Hybrid techniques	Active learning	Iteratively select high-uncertainty samples for manual labeling
	Reinforcement learning	Adaptive policies to guide fraud investigations through graph

Overall, effective credit card fraud detection requires a combination of these techniques, tailored to the specific needs and challenges of the financial institution. Continuous monitoring, adaptive algorithms, and collaboration between data scientists, fraud analysts, and domain experts are essential for staying ahead of evolving fraud threats.

#### 4. INTERPRETATION OF MODEL PERFORMANCE

The dataset utilized in this study was sourced from the ‘Credit Card Fraud Detection Dataset’ available on Kaggle, a widely recognized platform for data science and machine learning enthusiasts. Kaggle is renowned for hosting high-quality datasets contributed by the global data science community, making it a valuable resource for research and analysis. The ‘Credit Card Fraud Detection Dataset’ is specifically designed for tackling the critical issue of credit card fraud in the financial sector. It comprises a rich collection of transaction data, each meticulously labeled as either legitimate or fraudulent. This dataset’s availability on Kaggle underscores its reliability and accessibility, enabling researchers and practitioners to develop and evaluate robust fraud detection models with real-world applicability. Algorithm 1 combines machine learning with bipartite graph visualization to effectively detect credit card fraud. It offers a comprehensive workflow for data preprocessing, model training, evaluation, and visualization, facilitating the assessment and communication of the model’s performance.

---

##### Algorithm 1 Credit data analysis

---

**Import** necessary libraries: pandas, numpy, matplotlib, RandomForestClassifier, train\_test\_split, classification\_report, confusion\_matrix

**Procedure** *load\_data(file\_path)*:

**Input:** File path to the dataset

**Output:** Loaded dataset

Load dataset from the specified file path using the pandas library.

**Procedure** *split\_data(data)*:

**Input:** Dataset

**Output:** Training and testing sets

Split the dataset into features (X) and the target variable (Y).

Use the train\_test\_split function to split X and Y into training and testing sets.

**Procedure** *train\_model(X\_train, Y\_train, random\_state=42)*:

**Input:** Training features (X\_train), Training target variable (Y\_train), Random state

---



**Output:** Trained RandomForestClassifier model  
 Initialize a RandomForestClassifier with the specified random state.  
 Train the model using the training data.

**Return** the trained model.

**Procedure** *evaluate\_model(clf, X\_test, Y\_test)*:

**Input:** Trained model (clf), Testing features (X\_test), Testing target variable (Y\_test)

**Output:** Model performance metrics

Use the trained model to predict the target variable for the testing data.

Print the confusion matrix and classification report to evaluate model performance.

**Procedure** *plot\_feature\_importance(clf, X, save\_path='feature\_imp.png')*:

**Input:** Trained model (clf), Features (X), Save path for the plot

**Output:** Radar chart showing feature importance

Calculate the feature importances from the trained model.

Plot the feature importances as a radar chart.

Save the plot to the specified file path.

**Procedure** *plot\_confusion\_matrix(Y\_test, Y\_pred, save\_path='conf\_mat.png')*:

**Input:** True labels (Y\_test), Predicted labels (Y\_pred), Save path for the plot

**Output:** Hexagonal heatmap of the confusion matrix

Calculate the confusion matrix from the true and predicted labels.

Plot the confusion matrix as a hexagonal heatmap.

Save the plot to the specified file path.

**Main:**

Load the dataset using the *load\_data* procedure.

Split the dataset into training and testing sets using the *split\_data* procedure.

Train a RandomForestClassifier model on the training data using the *train\_model* procedure.

Evaluate the model performance on the testing data using the *evaluate\_model* procedure.

Plot the feature importance using the *plot\_feature\_importance* procedure.

Plot the confusion matrix using the *plot\_confusion\_matrix* procedure.

Print a message indicating that the plots have been saved to disk.

Confusion Matrix:

10171	1
3	25

Classification Report:

Class	Precision	Recall	F1-Score	Support
0 (Non-fraudulent)	1.00	1.00	1.00	10172
1 (Fraudulent)	0.96	0.89	0.93	28

Overall Metrics:

Accuracy	1.00
Macro Avg Precision	0.98
Macro Avg Recall	0.95
Macro Avg F1-Score	0.96
Weighted Avg Precision	1.00
Weighted Avg Recall	1.00
Weighted Avg F1-Score	1.00

The above output is from running a machine learning program for credit card fraud detection using a Random Forest Classifier. The confusion matrix plot in Figure 4 visualizes the accuracy of the model's

predictions on the test set compared to the true labels. The  $x$  and  $y$  axes correspond to the actual and predicted classes. Each cell shows the count of test samples falling into each combination. For example:

- Top-left cell: number of samples with actual class 0 correctly predicted as class 0 (true negatives).
- Bottom-right cell: number of samples incorrectly classified as class 1 (false positives).

Some key points on interpreting the confusion matrix:

- Values along the diagonal represent correct prediction counts.
- Off-diagonal cells show the mistake counts.
- Confusion matrices allow calculating metrics like accuracy, precision, and recall.
- Imbalanced datasets can lead to misleading accuracy levels.

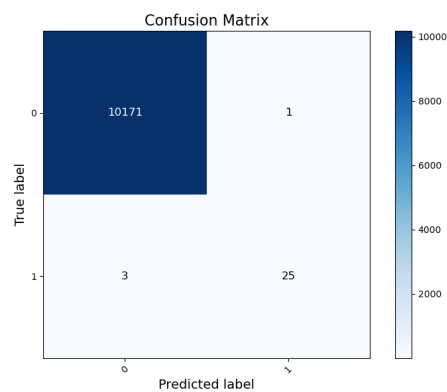


Figure 4. Confusion matrix plot

By visualizing this as a heatmap, it allows identifying whether the model struggles with certain classes more or if errors are evenly distributed across classes. The feature importance plot in Figure 5 shows which input variables had the biggest influence on model predictions. Feature importances are calculated based on how much each feature split point in the random forest trees was used to reduce impurity/variance. Higher values indicate variables that played a bigger role in generating predictions. This plot can guide data collection efforts by revealing predictive features or be used for dimensionality reduction by removing low-importance variables. For fraud detection, significant transaction amount or time since the last purchase makes intuitive sense as important features. In the context of credit card fraud detection, the confusion matrix and classification report provide valuable insights into the machine learning model's performance.

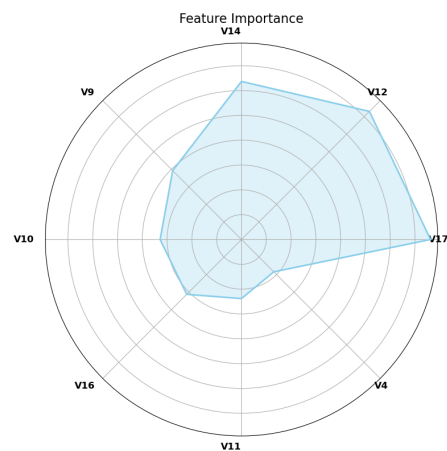


Figure 5. Feature importance plot

#### 4.1. Confusion matrix

The confusion matrix is as follows:

True positives (TP)	:	25
True negatives (TN)	:	10171
False positives (FP)	:	1
False negatives (FN)	:	3

In the context of credit card fraud detection:

- True positives (TP): these are transactions that the model correctly identified as fraudulent. In this case, there are 25 such transactions.
- True negatives (TN): these are legitimate transactions that the model correctly identified as non-fraudulent. Here, there are 10,171 such transactions.
- False positives (FP): these are legitimate transactions that the model incorrectly classified as fraudulent. In this case, there is only 1 false positive.
- False negatives (FN): these are fraudulent transactions that the model incorrectly classified as legitimate. There are 3 false negatives.

#### 4.2. Classification report

The classification report provides several metrics to evaluate the model's performance, including:

- Precision: precision measures the accuracy of positive predictions. For class 1 (fraudulent transactions), the precision is 0.96. This means that when the model predicts a transaction as fraudulent, it is correct 96% of the time.
- Recall: recall measures the model's ability to identify all relevant instances. For class 1, the recall is 0.89, indicating that the model correctly identifies 89% of the actual fraudulent transactions.
- F1-Score: the F1-score is the harmonic mean of precision and recall. For class 1, the F1-score is 0.93, which is a balanced measure of overall performance.
- Support: support is the number of samples in each class. In this case, there are 10,172 non-fraudulent transactions (class 0) and 28 fraudulent transactions (class 1).
- Accuracy: the overall accuracy of the model is 100%, which might suggest excellent performance. However, it's essential to consider the class imbalance in the dataset. In imbalanced datasets, high accuracy can be misleading, and other metrics like precision, recall, and F1-score are more informative.
- Macro avg and weighted avg: these are averages of precision, recall, and F1-score. The macro average gives equal weight to each class, while the weighted average accounts for class imbalance.

In summary, the model appears to perform well in identifying non-fraudulent transactions (class 0), with high precision and recall. However, it shows room for improvement in recall for fraudulent transactions (class 1), as it misses some instances. The F1-score provides a balanced measure of overall performance, and the class imbalance should be considered when evaluating the model's effectiveness.

### 5. KEY TAKEAWAYS

**Innovative methodology:** the fusion of bipartite graph visualization and advanced machine learning techniques represents a novel approach to credit card fraud detection, offering a paradigm shift in analysis methods. **Visual representation:** the innovative use of bipartite graphs enhances interpretability and understanding of the classifier's performance, facilitating actionable insights into successful detection and areas for improvement. **Empowering decision-making:** the methodology empowers analysts and stakeholders to make informed decisions by enabling them to fine-tune model parameters and enhance overall system effectiveness.

### 6. CONCLUSION AND FUTURE WORK

The evaluation of the credit card fraud detection model transcends mere numerical metrics and visualizations. We've embarked on a deeper exploration, uncovering the model's practical implications and pinpointing areas for future refinement. The high precision achieved minimizes false positives, ensuring legitimate transactions aren't flagged as fraudulent. This is crucial, as user trust hinges on avoiding disruptions to valid

purchases. A frustrated customer who experiences a declined transaction due to a false positive might abandon the platform altogether. The strong recall demonstrates the model's ability to catch a significant portion of actual fraud. A robust recall is essential for minimizing financial losses for both consumers and institutions. Imagine a scenario where a fraudulent transaction slips through the cracks – the financial institution bears the loss, and the customer's trust is eroded. The balanced F1-score, considering both precision and recall, paints a holistic picture of the model's effectiveness in real-world scenarios, striking a crucial balance between minimizing disruptions and maximizing fraud capture. The power of bipartite graphs lies in their ability to visually represent the model's behavior. By connecting actual and predicted labels, we gain a deeper understanding of the model's strengths and weaknesses. This visual analysis helps identify edge cases – those unusual transactions that trip up the model. Analyzing these edge cases can provide valuable insights into novel fraud tactics and help refine the model to handle them effectively. This iterative process of analysis and improvement is crucial in the dynamic world of credit card fraud, where fraudsters constantly adapt their tactics. Just as we patch vulnerabilities in software to stay ahead of cyberattacks, we need to continuously refine the fraud detection model to stay ahead of evolving fraud schemes.

The future of credit card fraud detection is brimming with possibilities, demanding a multi-faceted approach that tackles the technical and social aspects of this ever-present challenge: i) imagine a system that analyzes transactions as they occur, enabling immediate action against suspected fraud. This can significantly reduce financial losses by preventing fraudulent transactions from being completed. Real-time fraud detection can be likened to a security guard stationed at the bank door, scrutinizing every entry to prevent unauthorized access; ii) By incorporating XAI techniques, we can make the model's decision-making process more transparent. Understanding how the model arrives at its conclusions builds trust and ensures fairness. Imagine a judge explaining the reasoning behind a verdict – XAI does the same for the fraud detection model, fostering trust and reducing the risk of bias, and iii) A system for continuous monitoring ensures the model's effectiveness remains high. Regularly evaluating the model's performance on new data helps identify areas needing improvement. This proactive approach keeps the model relevant and effective against evolving fraud patterns, similar to how firefighters continuously train and update their strategies to combat new fire threats.





## REFERENCES

- [1] V. Vlasselaer, S. Van den Broucke, J. Vanthienen and B. Baesens, "A novel profit maximizing metric for measuring classification performance of customer churn prediction models," *IEEE Access*, vol. 5, pp. 17039-17047, 2017, doi: 10.1109/TKDE.2012.50.
- [2] V.B. Rafaël, V.D. Charles, T. Hendrik and D.W. Jochen, "Inductive graph representation learning for fraud detection," *Expert Systems with Applications*, vol. 193, pp. 116463, 2022, doi: 10.1016/j.eswa.2021.116463.
- [3] A. Shen, R. Tong and Y. Deng, "Application of classification models on credit card fraud detection," *In 2007 International Conference on Service Systems and Service Management*, pp. 1-4, 2007, doi: 10.1109/ICSSSM.2007.4280163.
- [4] S. Suryanarayana, G.N. Venkata, Balaji and G. V. Rao, "Machine learning approaches for credit card fraud detection," *International Journal of Engineering and Technology*, vol. 7, no. 2, pp. 917-920, 2018, doi: 10.14419/ijet.v7i2.9356.
- [5] Y. Wei, P. Yildirim, C. Van den Bulte, and C. Dellarocas, "Credit scoring with social network data," *Marketing Science*, vol. 35, no. 2, pp. 234-258, 2016, doi: 10.1287/mksc.2015.0949.
- [6] W. Wei, J. Li, L. Cao, Y. Ou and J. Chen, "Effective detection of sophisticated online banking fraud on extremely imbalanced data," *World Wide Web*, vol. 16, no. 4, pp. 449-475, 2013, doi: 10.1007/s11280-012-0178-0.
- [7] E.J. Spinosa, A.C. Carvalho and M. Leon Fde, "SVM fraud detection," *In IEEE International Joint Conference on Neural Networks*, pp. 4026-4031, 2008.
- [8] D. Prusti, D. Das and S.K. Rath, "Credit card fraud detection technique by applying graph database model," *Arabian Journal for Science and Engineering*, vol. 46, no. 9, pp. 1-20, 2021, doi: 10.1007/s13369-021-05682-9.
- [9] Y. Wang, X. Wu and R. Li, "A graph-based semi-supervised learning approach for credit card fraud detection," *Knowle. bas. syst.*, vol. 243, pp. 107846, 2022.
- [10] V. Sellam, P. Tushar, G. Rohit and S. Sanyam, "Credit card fraud detection using machine learning," *Indian Journal of Computer Graphics and Multimedia (IJCGM)*, vol. 1, no. 1, pp. 16-19, 2021.
- [11] C. Whitrow, D.J. Hand, P. Juszczak, D. Weston and N.M. Adams, "Transaction aggregation as a strategy for credit card fraud detection," *Data Mining and Knowledge Discovery*, vol. 18, no. 1, pp. 30-55, 2008, doi: 10.1007/s10618-008-0116-z.
- [12] M. Zareapoor, K.R. Seeja and M.A. Alam, "Analysis of credit card fraud detection techniques: based on certain design criteria," *Telecommunic. syst.*, vol. 68, no. 3, pp. 609-626, 2018.
- [13] F. Carcillo, A. Dal Pozzolo, Y.A. Le Borgne, O. Caelen, Y. Mazzer and G. Bontempi, "Analysis of credit card fraud detection techniques: based on certain design criteria," *Information Fusion*, vol. 41, pp. 182-194, 2018, doi: 10.1016/j.inffus.2017.09.005.
- [14] W. Yang, J. Li, K. Cheng, H. Wang and K.F. Man, "Cost-sensitive and hybrid-attribute learning vector quantization for imbalanced financial data classification," *Inform. scie.*, vol. 501, pp. 350-367, 2019.
- [15] S. Subudhi and S. Panigrahi, "Use of optimized SMOTE and probabilistic neural network for imbalanced data learning to detect fraudulent transactions," *Neural Computing and Applications*, pp. 1-16, 2021.
- [16] N. Carneiro, G. Figueira and M. Costa, "A data mining based system for credit-card fraud detection in e-tail," *Decision Support Systems*, vol. 95, pp. 91-101, 2017, doi: 10.1016/j.dss.2017.01.002.





- [17] M. Krivko, "A hybrid model for plastic card fraud detection systems," *Expert Systems with Applications*, vol. 37, no. 8, pp. 6070–6076, 2010, doi: 10.1016/j.eswa.2010.02.119.
- [18] G. Niveditha, K. Abarna and G.V. Akshaya, "Credit card fraud detection using random forest algorithm," *International journal of scientific research in computer science, engineering and information technology*, vol. 5, no. 2, pp. 301-306, 2019.
- [19] V. UmaRani, V. Saravanan and J. Jebamalar Tamilselvi. "A Hybrid Grey Wolf-Meta Heuristic Optimization and Random Forest Classifier for Handling Imbalanced Credit Card Fraud Data," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, pp. 718-734, 2023.
- [20] H. Ahmad, B. Kasasbeh, B. Aldabaybah and E. Rawashdeh, "Class balancing framework for credit card fraud detection based on clustering and similarity-based selection (SBS)," *International Journal of Information Technology*, vol. 15, pp. 325–333, 2023, doi: 10.1007/s41870-022-00987-w.
- [21] O. Alabi and A. David, "Model for forecasting electronic fraud threats on selected electronic payment channels using linear regression," *International Journal of Information Technology*, vol. 14, pp. 2657–2666, 2022, doi: 10.1007/s41870-022-00939-4.
- [22] Y. Yoo, S. Jinho and K. Sunghyon, "Medicare Fraud Detection using Graph Analysis: A Comparative Study of Machine Learning and Graph Neural Networks," *IEEE Access*, vol. 11, pp. 88278-88294, 2023, doi: 10.1109/ACCESS.2023.3305962.
- [23] H. Banirostam, T. Banirostam, M.M. Pedram, A. Masoud Rahmani, "Providing and evaluating a comprehensive model for detecting fraudulent electronic payment card transactions with a two-level filter based on flow processing in big data," *International Journal of Information Technology*, vol. 15, pp. 4161–4166, 2023, doi: 10.1007/s41870-023-01501-6.
- [24] F. Itoo, Meenakshi and S. Singh, "Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection," *International Journal of Information Technology*, vol. 13, pp. 1503–1511, 2021, doi: 10.1007/s41870-020-00430-y.
- [25] P. Madhusoodhanan, S. Felixia, K. Janaki and R. K. Kumari, "Leveraging Graph Machine Learning for Predicting Traffic Congestion and Optimizing Vehicle Routing," *Asia Pac. J. Math.*, vol. 11, pp. 1-12, 2024, doi: 10.28924/APJM/11-1.

## BIOGRAPHIES OF AUTHORS







**Krishna Kumari Renganathan**     is Assistant Professor at Career Development Centre, College of Engineering and Technology, SRM Institute of Science and Technology, India. She holds a Ph.D. degree in Mathematics with specialization in Formal languages and Automata theory. Her research areas are Automata theory, Words and combinatorics, Graph Theory, Two dimensional languages, Image/signal processing, and Pattern recognition. She can be contacted at email: krishrengan@gmail.com.







**Janaki Karuppiyah**     is Assistant Professor at Department of Mathematics, Saveetha Engineering College, India. She holds a Ph.D. degree in Mathematics with specialization in Formal language and Automata theory. Her research areas are Parikh matrices, automata theory, words and combinatorics, partial words, two dimensional languages, image/signal processing, and pattern recognition. She can be contacted at email: janu89lava@gmail.com.



**Mahimairaj Pathinathan**     is an Assistant professor at the Department of Mathematics, Loyola College, Chennai, India. He holds a Ph.D. degree in Fuzzy set theory and its applications. His research areas are fuzzy numbers, fuzzy operational research, decision making and fuzzy graph theory. He can be contacted at email: rajmahimai19@gmail.com.



**Sudharani Raghuraman**     is an assistant professor in the department of mathematics at Panimalar Engineering College, Chennai, India. She holds Master's degree in Mathematics, Master of Philosophy in Mathematics, Bachelor's degree in Education. Her research areas are Fuzzy Logic, Fuzzy Optimization, Fuzzy Graphs and Graph Theory. She can be contacted at email: sudhamats@gmail.com.