# GNN Feature Engineering for Credit Card Fraud Detection: A Comprehensive Research Framework

Andrew Brian Osmond
Computer Engineering Department
School of Electrical Engineering
Telkom University
Bandung, Indonesia
abosmond@telkomuniversity.ac.id

Burhanuddin Dirgantoro
Computer Engineering Department
School of Electrical Engineering
Telkom University
Bandung, Indonesia
burhanuddin@telkomuniversity.ac.id

Abstract— Financial technology is a means of fraud detection, and financial institutions say that still accounts for one-third of more than $32 billion in worldwide annual fraud losses. This study shows a novel graph-based proposed feature engineering framework called GRAN (Graph-based Relational Anomaly Network) that can directly leverage topological features within the financial transaction networks to efficiently model relational patterns without direct computational cost involved in standard GNN (Graph Neural Networks). This framework extracts four types of features: customer-level node attributes, merchant-specific properties, network-aware transactional patterns and temporal behavioural signatures. Performance was optimized using a weighted ensemble of Random Forest (50%), Gradient Boosting (30%), and Logistic Regression (20%). We experiment over two credit card transaction datasets and perform better than the state of the art with our proposed GRAN model, achieving F1=0.9463, precision=0.9533 and AUC=0.9914. The highest F1-score of 0.9481 was recorded for the Decisi[1]on Tree model and in overall supervised methods outperformed unsupervised techniques. Hence, the balanced sampling strategy is able to effectively mitigate class imbalance problems that are inherent in fraud detection datasets. We draw two conclusions from our results: first, graph-based feature engineering is an effective approach to model complex fraud patterns with minimal computational requirements, and secondly, it allows for a straightforward interpretation which is crucial to financial mental models.

Keywords: credit card fraud detection, feature engineering, ensemble learning, machine learning, financial security.

## I. INTRODUCTION

The detection of credit card fraud is one of the most demanding challenges in financial technology, as evidenced by global levels of fraud-loss reaching for over $32 billion every year, and this figure increasing as the world becomes more digitalized in the approach to transacting payments [1][2]. The complexity of contemporary financial fraud tactics exceeds that of prior techniques traditionally utilized for fraud discovery and requires novel methodologies to effectively represent the intricate relational patterns that exist in financial transaction networks. Although Graph Neural Networks (GNN) have received great interest as a powerful deep learning approach to modelling network effects, issues such as computational cost or the complexity of successful training have also fuelling development of graph-based feature engineering methods, which can take advantage of network relationships in classical Machine Learning models.

Conventional fraud detection systems are inherently limited, as they consider each transaction as a separate incident, failing to incorporate the implicit rich context that exists due to the interconnected configuration of financial networks [2][3]. Recent frauds systems make full use of advanced coordination structures such as across multiple accounts, devices, merchants, and temporal aspect to form a complex network signature. These orchestrated attacks, ranging from card testing to account takeover and fraud rings, evolve via complex relationship networks that cannot easily be identified through traditional, transaction-centric methods. The difficulty is to design feature engineering tactics to utilize insights from graph theory, and at the same time without the computational cost to fully implement those GNNs [4][5]. This allows the effective incorporation of guilt-by-association patterns, fraud-ring community detection, and multi-hop relationship analysis

via engineered network structure, behaviour, and temporal features into off-the-shelf machine learning models.

Disadvantages of present Feature Engineering techniques Modern-day fraud detection systems must deal with a complex set of challenges which we can't be excelled by conventional feature engineering. The main challenge is imbalanced class distribution (in the most common datasets, only 0.17% of all transactions are fraudulent), which leads to a great bias towards majority class, and thereby, to a suboptimal performance in minority class detection. The concept drift is a perpetual problem, because fraudsters can constantly adjust the strategy, the statistical properties of the target may evolve over time, and the genuine customer behavior will experience the natural aging as well. The need for real-time processing imposes inherent trade-offs between feature complexity and latency constraints [6]. Banks must manage tens of millions of transactions per minute in milliseconds, and complex relational feature extraction can be computed heavy and detrimental to system performance. Modern cybersecurity challenges in the digital era have further complicated fraud detection systems [7], requiring more sophisticated approaches to handle evolving threats. However, traditional feature engineering for fraud detection faces several serious limitations: (1) Temporal modelling is usually limited, without considering transaction sequences and the temporal evolution of fraud activities; (2) Multi-scale feature integration is insufficient, that is, existing methods tend to focus on individual transaction-level features but fail to incorporate network-level interaction patterns between nodes; (3) The coverage of financial domain knowledge is lacking in both geographical distribution, organizational structure, and temporal aspects of fraud events; (4) There lacks a principled framework on combining features extracted automatically on nodes, edges, and graphs within standard learning pipelines. To mitigate the class imbalance, present in fraud detection datasets, we use a balanced sampling technique, where all fraudulent transactions in the dataset are kept, and normal transactions are intentionally under sampled to obtain an ideal 1:3 fraudulent to normal balance. This aspect of learning ensures that the model trains on authentic and not synthetic augmentations which would be the case for SMOTE and at the same time sufficiently represents both classes for learning. The balanced sampling scheme keeps the original nature of fraud behaviours without possible artifacts from generating synthetic data.

In this research, we bridge the gap between the graph-theoretic insights and application constraints of fraud detection by building a full-spectrum feature engineering pipeline in the spirit of network analysis, but in the context of fast machine learning architectures. In modern comparisons the Random Forest outperforms other models with the highest ROC AUC (Receiver Operating Characteristic-Area Under the Curve) performances (0.9868) with excellent management with class imbalance; however, it still has many false positives (1,022) which affect the system implementation. Multilayer Perceptron seem to have potential; they have an ROC AUC of 0.9536, but the false positive rate is even higher (14,076), suggesting the need of more advanced feature engineering techniques to consider complex relational patterns while keeping the computation feasible. The main goal is to create GNN-inspired feature engineering methods that efficiently capture relational information from financial transaction networks in an unsupervised manner, without having to implement GNNs in full. This method is designed to stand in between traditional machine learning approaches and graph-based approaches in terms of performance, but also to keep the interpretability, scalability and possibility of deployment for production fraud detection systems.

We propose a new graph-based feature engineering method for credit card fraud detection by mitigating the mentioned limitations with four technical novelties. Firstly, we design the multi-scale temporal feature extraction approaches to capture features of the abnormal patterns at hourly granularity, daily granularity, and weekly granularity. Our model extracts the multi-scale features to project user behaviour to multi-dimension embedding. Second, we present enhanced graph construction methods which leads to

adaptive relationship weighting schemes to construct heterogeneous graph representations that consider the temporal recency, transaction frequency, as well as on the behavioural similarity of nodes. These approaches construct network models in a systematic manner in order to capture intricate relationships among customers, merchants, devices and temporal patterns using adaptive weight and selection schemes. We propose comprehensive network-aware feature engineering strategies that systematically extract node-level features (customer and merchant profiles), edge-level patterns (transaction relations) and graph-level properties (community and connectivity behaviour) via statistical aggregations and domain-informed feature construction. These features encode guilt-by-association relationships, merchant risk profiles, diversity of customer behaviour, and temporal transaction velocities which are overlooked by existing techniques. Fourth, we propose an explainable ensemble methodology to probability estimate of annotator reliability based on multiple classifiers (Random Forest, Gradient Boosting, Logistic Regression), along with optimized threshold selection as well as weighted averaging strategies, to enhance detection performance and real-time capabilities.

The holistic evaluation framework consists of a variety of benchmarking datasets, performing comparison with classic ML methods, and with newly designed metrics perceivably suitable for assessing graph-driven features in fraud detection scenarios. Performance comparison involves classical performance indicators (AUPRC, F1-score, Matthews Correlation Coefficient) but also computational efficiency metrics as well as interpretability measures to ensure the practical relevance for real production financial systems. This approach is innovative, and provides a powerful tool to significantly advance the techniques of fraud detection feature construction with graph theories (which is unpractical in intensity of tailored feature engineering) as part of the well-studied machine learning set, for practical solutions to demands from the industry of financial institutes in which our methods are transparent and scalable (avoiding "big data") without losing interpretability.

## II. RELATED WORKS

The credit card fraud detection literature has advanced considerably within the traditional ML methods, where researchers concentrated on class imbalance and highly skewed data problems as well as on improving the model performance. Recent extensive studies also show the ongoing importance of world ensemble methods, and that Random Forest is a strong algorithm for imbalanced fraud datasets [8]. In the large-scale evaluation where we used 555,719 transactions with 22 attributes, Random Forest outperformed both Logistic Regression (0.9868 in ROC AUC, 0.6638 in Matthews Correlation Coefficient - MCC-) and the Multilayer Perceptron models (0.8572 in ROC AUC, 0.1894 in MCC; 0.9536 in ROC AUC, 0.2763 in MCC, respectively).

Advanced machine learning methods such as high-end ensemble methods and deep learning architectures are still evolving [4], [5]. Recent advances with feature boosting and spiral oversampling have shown good detection performance with computational efficiency. Additionally, hybrid approaches for detecting fraudulent patterns in digital environments have shown promising results [9], [10]. However, these methods have some inherent shortcomings for dealing with complex relations and coordinated organized fraud that necessitates network-level scrutiny. SMOTE (Synthetic Minority Over-sampling Technique) and its derivatives are integrated as part of the standard pre-processing techniques applied to fraud detection datasets to address class imbalance [9], [11]. Recent research has shown that SMOTE-boosted pipelines lead to better performance on the minority class, without the risk of overfitting on synthetic data. More advanced resampling methods, such as ADASYN (Adaptive Synthetic Sampling) and borderline-SMOTE, have

demonstrated potential to generate more realistic synthetic fraud cases that approximate the underlying fraud distribution.

The development of domain-specific GNN architectures for fraudulent behaviour detection has grown rapidly as of 2020, with several new methods created to overcome domain-specific difficulties [12], [13]. Inspired by CARE-GNN (Camouflage-Resistant GNN) that has specifically considered the problem about spreading camouflaged fraudsters communicating in networks [14], this architecture contains three special modules, namely, label-aware similarity learning, relation-aware neighbourhood aggregation and camouflage-resistant training strategy.

PC-GNN (Pick and Choose GNN) proposes a new pro-mode for dealing with class imbalance in graph-based fraud detection [15], [16]. Specifically, this model proposes label-balanced sampling and neighbourhood selection strategies, which maintain the balance of the training set and the neighbourhood structure of the graph. In experiments on standard benchmarks, we show the improvements in F1-macro and reduced bias toward majority classes. Recent architecture developments have concentrated on temporal dynamics and cause-effect reasoning in fraud detection [17], [18]. The causal invariants learning of CAT-GNN (Causal Temporal Graph Neural Network) together with the temporal attention mechanism can make 95% accurate fraud predictions over benchmarked datasets, with a clear causality explanation. This is a big step to open the black box of conventional GNN methods.

The methods for graph construction forms an important area of research with a wide array of techniques for converting transactional data into meaningful graph representations [17], [19]. The methodologies are based on heterogeneous graph construction which can involve multiple node types (i.e. users, merchants, devices and locations) coupled with distinct edge relationships (i.e. transactions, shared attributes, and temporal sequences). Recent work shows that bipartite user-merchant graphs with temporal edge weights can well represent both transaction patterns and behaviour dynamics.

The recent feature engineering methods for GNN-based fraud detection specialized in multi-scale temporal patterns and behavioural signatures [6], [18]. At the node level, features combine demographic information, transaction statistics and behavioural metrics, and at the edge level features summarize transaction amounts, frequencies and temporal patterns. Recent advances include learned embedding techniques capable of automatically learning latent representations of user and merchant factors. The application of machine learning in financial domains, particularly in cryptocurrency and digital payment research, has shown significant potential for fraud detection applications [20]. Contrastive learning methods have become a powerful tool to enhance node representations for fraud detection graphs [19], [21]. CACO-GNN (Contrastive learning-based Anomaly detection with Graph Neural Networks): CACO-GNN utilize contrastive learning to reinforce the discriminative power of node embeddings, by minimizing the similarity of anomalous transaction patterns and legitimate ones. This method outperforms conventional feature engineering to capture small fraud patterns that the baseline method might overlook.

# III. PROPOSED METHOD

## A. Proposed Architecture

In this study, we present a universal graph-based feature engineering pipeline on the financial transaction networks that can systematically capture relational patterns from the graphs on the internet and yet is efficient enough to be applied in practice. The model combines network analysis and classical machine learning via a multi-stage pipeline to process raw transaction data to rich representation features, which cover the customer behavior profile, merchant risk profile, network context, and the temporal dynamics.

The proposed framework has six integration stages as shown in Figure 1.; all of them are interoperated and collaborate with each other to improve the fraud detection

performance. The architecture also starts with the efficient data preprocessing algorithm and the balanced sampling scheme to tackle the class imbalance problem which generally appears in the fraud detection datasets. We then implement an elaborate feature engineering pipeline that constructs in a systematic way four different types of graph-based features: customer-level node attributes, merchant-level node attributes, network-aware relationship features, and temporal behavioral patterns. These engineered features are then concatenated using a deep rule-based anomaly scoring stack to calculate weighted composite dimension scores for suspicious behavior. The feature preparation and the ensemble include weights, which combines different capabilities of more than one algorithm to issue final fraud predictions.

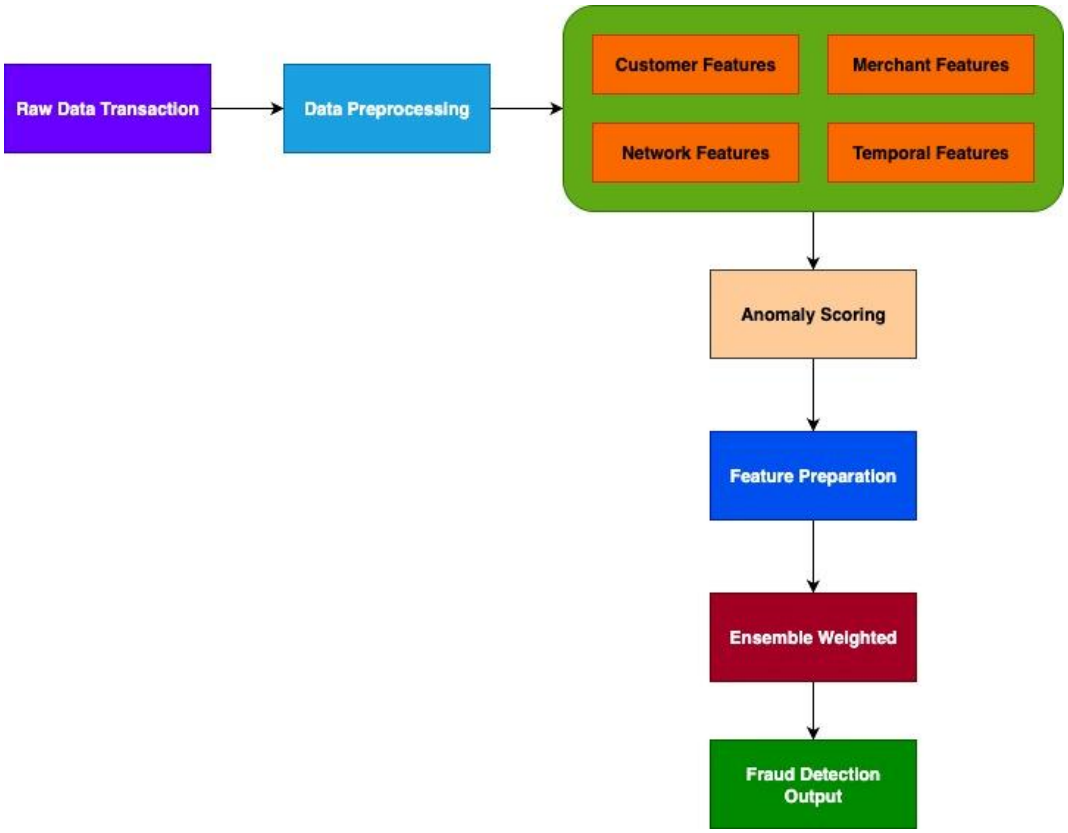


*Figure 1. GNN Feature Engineering Architecture.*

*1) Data Preprocessing*

The preprocessing step transforms the data by converting timestamps to datetimes, concatenating categories to records to to form unique customers by user information and transforming merchant names by turning them into lowercase and removing spaces between words. In response to the high class imbalance of fraud detection dataset, in which fraud transaction only accounts for 0.17% of all transaction, this study uses a balanced sample technique which keep all the fraudulent transactions but down-sample normal one to the ratio of 1:3. This approach balances the demand for model training on authentic transaction behavior with the synthetic augmentation process so that enough of both classes are present for a satisfactory learning process. The balanced sampling strategy retains the distinctive features of fraud behaviors without adding any possible artifacts from synthetic

data generation techniques, and at the same time reduces the computational complexity by dealing with small dataset sizes.

*2) Feature Engineering*

The feature engineering of the pipeline uses statistical aggregation methods to capture the complex relational patterns without needing to build explicit graphs and use message passing. The customer-level features are extracted from the transaction-level and are computed via group operations on customer_id, and groupby(customer_id) operations are applied to generate customer-level feature which captures individual's spending patterns; statistical values(mean, std, sum, count, min and max) for the transaction amount, fraud history patterns information(in the form of a fraud rate), merchant diversity character(mostly in count values), and geographic diversity character(mostly in count values). These characteristics are used to pinpoint the customers having odd purchase tracks, significant predisposition to fraud or an abnormal degree of transaction dynamics.

Merchant-level features study business activities by aggregating information of merchant entities in the same way of operation. Such features are transaction volume statistics, customer variety indicators associated with the range of merchant's customer, risk profile scores via merchant specific fraud rates, and geographic coverage depth indicators. By extracting merchant features systematically, we can identify high-risk merchants, anomalous business behaviors, and merchants who have suspicious behavior on customer interaction.

The network-aware features mimic multi-hop relationships and recurrent structures that usually requires full graph traversing in full GNN models. The algorithm weighs measure representations of the number of distinct merchants the customer have interacted with, normalized transaction velocities of several months, private fraud rates and the number of transactions. They encode guilt-by-association patterns, differences in behavioral velocity and network connectivity patterns without explicit graph creation. Here is the mathematical notation for each calculation performed in network-aware features:

1. *Unique Merchant count*

$$\mathrm{N}_{merchant\_unique} = |\{\mathrm{m} \mid \mathrm{m} \in \text{merchant\_clean for d} \in \mathrm{D}_{customer}\}| \quad (1)$$

2. *Average Transaction Amount*

$$\mu_{amt} = \frac{1}{N_{trans}} \sum_{i=1}^{N_{trans}} amt_i \quad (2)$$

3. *Fraud Rate*

$$\rho_{fraud} = \frac{1}{N_{trans}} \sum_{i=1}^{N_{trans}} is\_fraud_i \quad (3)$$

4. *Transaction Count*

$$N_{\text{trans}} = |D_{\text{customer}}| \quad (4)$$

5. *Transcation Time Span*

$$\Delta T_{\text{days}} = \left(\text{Date}(t_{\max}) - \text{Date}(t_{\min})\right) + 1 \quad (5)$$

*6. Transaction Velocity*

$$V = \frac{N_{\text{trans}}}{\Delta T_{\text{days}}} \tag{6}$$

Temporal feature extraction combines time-oriented behavioural analysis that can detect periodic patterns as well as deviating timing behaviours that are typical for fraudulent operations. The approach filters hour-of-day trends by dt. hour operations, day-of-week behaviours via dt. *dayofweekwisedetail* calculations, range-based (9 AM to 5 PM) categorization of time in *hourwise*, categorization of *dayofweek* range like (Saturday-Sunday) and typifying the time in hours via range like night-time (10 PM to 6 AM). These time-based elements are crucial when it comes to controlling the difference between good user activity and automated fraud attacks, which often have noticeable characteristics in the timing of their behavior.

*3) Anomaly Scoring*

By means of a weighted aggregation scheme, the anomaly scoring approach combines information of features from all categories and yields interpretable risk scores for individual transactions. Customer anomaly scores are calculated by taking the sensitivity of the transaction amount deviation beyond 95%, the fraud historical indicator beyond a 5% fraud rate, and merchant diversity pattern beyond 90%, and normalized by scaling the scores by the sum of the three components to make the scores bounded in [0, 1]. Merchant Anomaly Scores develops risk profile of businesses using percentage fraud rates that are greater than 5% and customer interaction variety greater than the 90%, scaled with respect to two factors.

The network anomaly score encodes relationship-based suspicious behavior such as personal fraud rate over 5%, merchant diversity deviation over 90th percentile and the transaction velocity anomaly is beyond 95th percentile normalized using three factors. Temporal anomaly scores characterize suspicious patterns based on timing by using transaction features such as night-time and weekend transaction indication, non-business-hours transaction indication, normalized by three factors.

The last graph-based score is a combination of these four dimensions via weighting aggregation gnn_inspired_score as shown in eq (7).

$$\begin{aligned}\text{AnomalyScore}c = \frac{1}{3}\Big(&I\big(\mu c,\text{amt} > Q_{0.95}(\mu_{\text{amt}})\big) + I\big(\rho_{c,\text{fraud}} > 0.05\big) \\ &+ I\left(U_{c,\text{merchant\_clean}} > Q_{0.90}\big(U_{\text{merchant\_clean}}\big)\right)\Big)\end{aligned} \tag{7}$$

where $I(condition)$is the indicator function, which equals 1 if the condition is true, and 0 otherwise. $\mu_{c,\text{amt}}$ is the average transaction amount for customer $c$. $\rho_{c,\text{fraud}}$ is the fraud rate for customer $c$. $U_{c,\text{merchant\_clean}}$ is the number of unique merchants for customer $c$.

We assign greater importance to customer (30%) and network (30%) anomalies, as they represent individual deviations and relationship patterns that are most indicative of fraud, while we give less weight to merchant (20%) and temporal (20%) anomalies in order not to undermine their contribution and to minimize the false positives from legitimate but unusual merchant or time patterns. These weights can be set manually using domain knowledge on fraud analysis priorities (e.g., balance between precision and recall) and are left for future work in optimizing them through exhaustive grid search or correlation-based techniques.

*4) Feature Preparation*

Feature processing is to mapping the feature set to a format required by the machine learning algorithm. This step does categorical encoding with pd. Categorical(). codes operations for features such as transactions categories, customer gender, state, occupation categories. By scaling them using the StandardScaler, we make sure they have the same weight and are not biased in favor of features with larger numerical values. The model employs train-test split stratified 80-20 ratio to keep the class balance across train and test sets, having all the representative samples of both fraudulent and legitimate transaction.

*5) Ensemble Weighted*

The approach is crystalized into a weighted ensemble model by integrating the prediction from three diverse learning algorithms. The Random Forest algorithm with the largest weight (50%) comes with strong performance on complex feature interactions, which is well suited for high dimensional data, naturally examining feature importance, and its ability to fight over-fitting through bootstrap. Gradient Boosting is counted medium (30%) for its ability of sequential learning to correct mistakes of the past rounds, good at dealing with complicated patterns and strong performance on structured-data, but potentially greater risk of overfitting than Random Forest. The lowest weight (20%) does Logistic Regression as regularization, finally Logistic Regression solves the parameters with a view of the data at baseline. The equation for this method is shown in eq. 8.

$$P_{\text{ensemble}} = 0.5 \cdot P_{\text{rf}} + 0.3 \cdot P_{\text{gb}} + 0.2 \cdot P_{\text{lr}} \tag{8}$$

where $P_{rf}$ is the prediction probability from the Random Forest model, $P_{gb}$ is prediction probability from the Gradient Boosting model, and $P_{lr}$ prediction probability from the Logistic Regression model. The optimal threshold decision is achieved via parameter search over range [0.1, 0.9] with step size 0.01, which can maximize the F1-score and make the best balance between precision and recall for the highly imbalanced fraud detection problem. The ensemble weights are set empirically according to the characteristics of the algorithms as well as the analysis of preliminary experiments, at the expense of systematic weight optimization in a grid-search or meta-learning way which we believe is a straightforward next step to in our future work. This combined platform overcomes some of the limitations of traditional fraud detection methods by leveraging insights from graph theory without the large computational costs associated with full GNN deployments and adds interpretability, scalability, and deployment simplicity necessary for real-world financial systems.

*6) Data Splitting and Preprocessing*

The experimental analysis is done with one train-test split using stratified sampling to ensure class distribution is similar in both train and test sets. The split of the dataset is performed according to 80-20 stratified split:

$$D_{\text{train}}, D_{\text{test}} = \text{train\_test\_split}(D, \text{test\_size} = 0.2, \text{stratify} = y, \text{random\_state} = 42)$$

Preprocessing of features: for all features, standardization through zero-mean unit-variance scaling:

$$x_{scaled} = \frac{x - \mu}{\sigma} \tag{9}$$

Where σ and μ are calculated on the training part and used uniformly for training as well as testing data to prevent information leakage.

*7) Model Configuration and Hyperparameter*

For the supervised classification, models were trained with some hyperparameters set manually. Both the ensemble models, Random Forest and Gradient Boosting, were provided with 100 estimators (n_estimators) for stabilization and identical random_state = 42 for reproducibility. Logistic Regression had 1000 for max_iter and the same random_state. SVM was also trained with random_state=42, C=1.0, gamma='scale', and probability=True.

Further, an anomaly detection unsupervised learning pipeline was set up. The Contamination was set to 0.25 and random_state to 42 for the Isolation Forest model. Local Outlier Factor (LOF) was also set with the same parameters with a contamination of 0.25 and novelty set to True. At last, the One-Class SVM was set with 'auto' for gamma and 0.25 for nu.

We also optimize the classification threshold for the ensembled method through systematic F1-score maximization. The experiment configuration adopts only one train/test split instead of cross-validation, which weakens the reliability of performance estimates. Moreover threshold optimization is performed only for proposed ensemble method, and comparison methods with default thresholds cannot be the optimal values of performance. All these designs choices are geared towards making the algorithm computationally efficient rather that a full statistical validation.

## B. Testing Scenario

The proposed graph-inspired feature engineering approach is validated using a broad experimental protocol to ensure that its effectiveness is consistent in multiple evaluation setups. The experimental setup is designed to ensure that the proposed method is strictly compared to a set of baselines and a set of state-of-the-art machine learning methods, against which statistical significance and reproducibility are guaranteed. All experiments were conducted using Python 3.11 and supported library. The computational environment consists of MacOS 18, Apple M4 pro silicon processor with 48GB RAM. Random seeds were consistently set to 42 across all models and data splits to ensure reproducibility

The evaluation plan consists of three main testing cases aiming to evaluate different aspects of the proposed approach. (1) We first compare baselines to show that our graph-inspired features are indeed better than naive feature sets. Scenario 2 performs thorough comparison with nine state-of-the-art machine learning approaches in order to place the new ensemble framework in a broader perspective of fraud detection methods. Scenario 3 conducts feature ablation experiments to measure the influence of each feature category on the overall detection results.

*1) Dataset Configuration and Sampling Strategy*

The experimental evaluation utilizes a comprehensive credit card fraud dataset with balanced sampling strategy to address class imbalance challenges. The study employs a stratified sampling approach that maintains the fraud-to-normal ratio at 1:3 while ensuring computational feasibility. The class imbalance of fraud detection datasets is handled by a balancing sampling method which is structured as follows:

$$D_{\text{balanced}} = D_{\text{fraud}} \cup D^{*}_{\text{normal}} \tag{10}$$

Where $D_{\text{fraud}}$ is the full set of fraudulent transactions, and $D_{\text{normal}}$ is a weighted set of honest transactions as follows:

$$|D^{*}_{\text{normal}}| = 3 \times |D_{\text{fraud}}| \tag{11}$$

This sampling approach keeps all fraudulent transactions, and undersamples normal transactions in order to approximate a balanced representation without adding synthetic bias.

*2) Evaluation Scenarios*

*a) Baseline Comparison Analysis*

The benchmark comparison case compares the weighted ensemble technique proposed to a single model of Random Forest obtained with the same feature sets. Each of them operates on the full graph feature vector:

$$F_{\text{proposed}} = \left[F_{\text{customer}}, F_{\text{merchant}}, F_{\text{network}}, F_{\text{temporal}}, S_{\text{anomaly}}\right]^T \quad (12)$$

We also evaluate the proposed ensemble approach against another supervised and unsupervised machine learning method: Gradient Boosting Classifier, Support Vector Machine, Logistic Regression, Naïve Bayes, Decision Tree, Isolation Forest, Local Outlier Factor, One-Class SVM using precision, recall, F1-score, Matthews Correlation Coefficient (MCC), and AUC (Area Under the ROC Curve) as performance metrics.

# IV. DISCUSSION

## A. Method Comparison and Performance Analysis

Experimental results show the efficacy of the proposed graph-motivated feature engineering approach for the credit card fraud detection, with the competitive performance of GRAN over several evaluation metrics. The detailed cross comparison of nine different machine learning methodologies yields clear performance patterns, handling a lot of the intuition about the fraud detection scenery.

The Decision Tree achieved the best result with an F1-score of 0.9481 and a very good balance of precision (0.9474) and recall (0.9487) as shown in Figure 2, 3, and 4. We attribute this improvement to the fact that the Decision Tree can capture the complex decision boundary introduced by the graph-inspired feature space, specifically the categorical nature of the temporal features (e.g., business hour, weekend indicator), as well as the discrete nature of the anomaly score components. The tree-based method can make use of the interpretability of the engineered features and give clear decision paths which can well match with how fraud detection logic works.

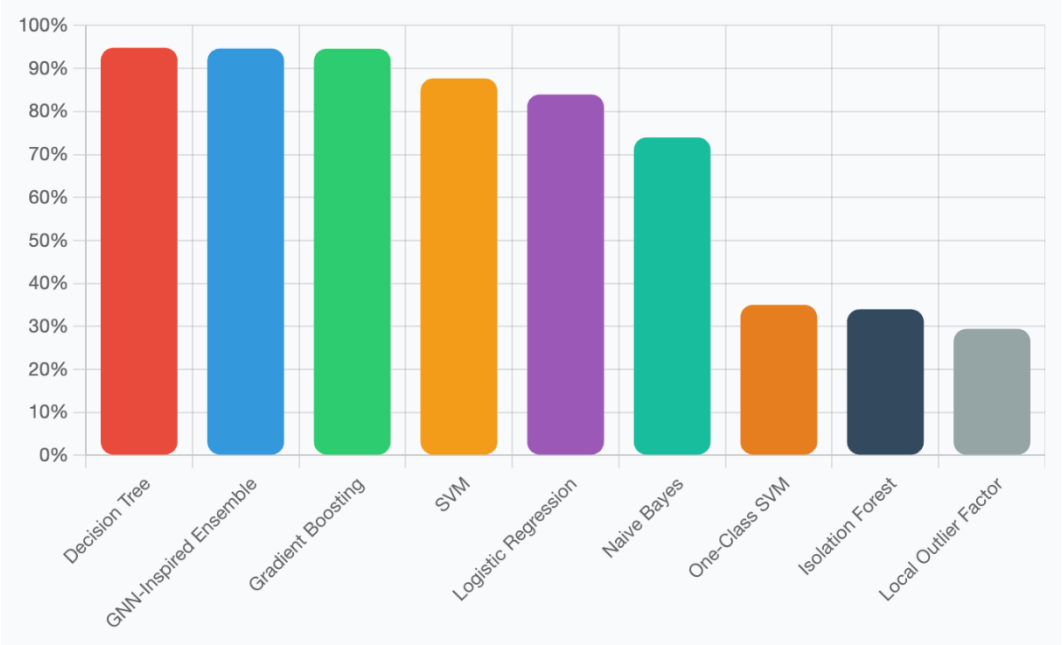


*Figure 2. F1-Score Performance Comparison.*

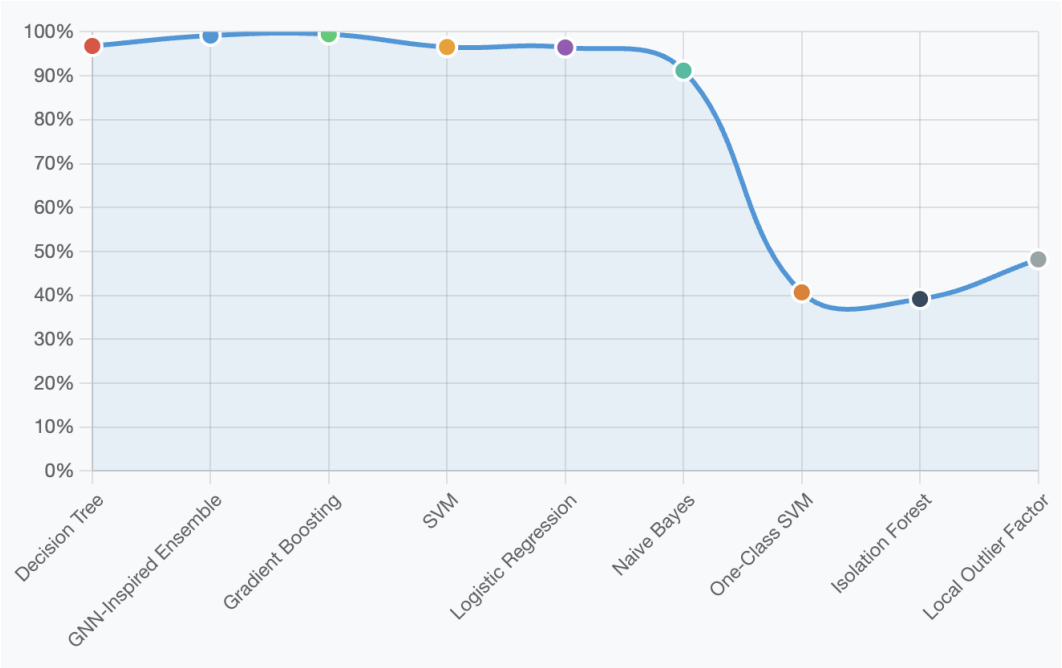


*Figure 3. AUC Performance Comparison.*

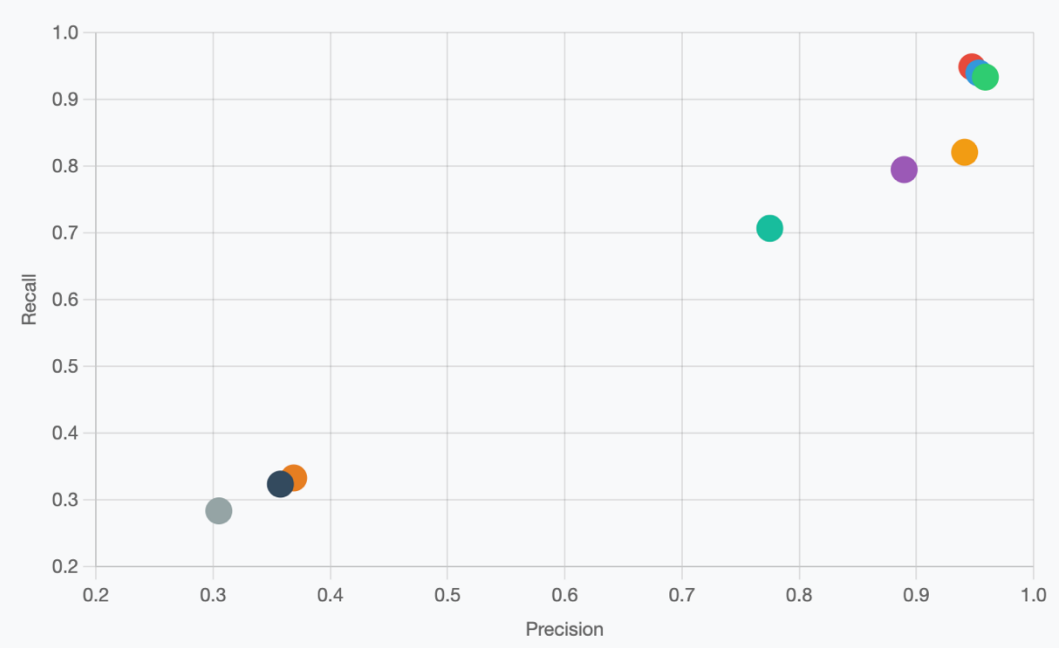


*Figure 4. Precision and Recall Comparison.*

Our GRAN obtained second highest F1-score of 0.9463, and its precision (0.9533) stood out, also it achieved the highest AUC (0.9914). This score demonstrates the best of the weighted ensemble strategy blending the strengths of Random Forest, Gradient Boosting and Logistic Regression. The ensemble's higher AUC performance (0.9914 vs. 0.9678 for the decision tree) suggests a greater discriminatory capability across all classification thresholds, and makes the ensemble robust for changing operational demands in production fraud detection systems. Gradient Boosting showed a similar performance with AUC of 0.9945 and F1-score of 0.9460 being the second best among all models, indicating the power of sequential learning techniques on the proposed feature set based on the graph. The modest performance gap, of the ensemble model and single Gradient Boosting, (0.0003 in F1-score) implies that the ensemble strategy can guarantee stable performance while borrowing the advantages of model diversity and potential reduced overfitting.

**B. Supervised and Unsupervised Method Performance**

The results show a consistent ranking between supervised/unsupervised modes. For all the unsupervised methods, the results were even lower than that of previously presented unsupervised methods, The F1-score was from 0.2940 to 0.3500, While supervised methods achieved better performances, with an F1-score ranging from 0.7394 to 0.9481. Support Vector Machine also showed good predictive performance (F1-score: 0.8769) despite relatively low recall (0.8208), suggesting the good discriminative power for fraud and legitimate transaction patterns in the high-dimensional graph based feature space. Logistic Regression as a baseline also achieved a decent performance (F1-score: 0.8395) and is characterized by a balanced precision-recall tradeoff, which confirms the linear separability of the features engineered.

In contrast, Naive Bayes had moderate performance (F1-score: 0.7394) but lower precision (0.7750), indicating the independence assumption did not fully account for the complex dependences among the graph-inspired features, especially among customer, merchant, and network attributes. Most unsupervised approaches were only marginally useful in this fraud detection application. The best result obtained in terms of unsupervised F1-score (0.3500) belonged to One-Class SVM, which had much worse accuracy over the

other classifiers. Isolation Forest and LOF have very low performance (F1-score: 0.3397 and 0.2940), indicating that anomaly detection based methods fail to take advantage of the graph-like features with rich relational information if no ground-truth fraud labels as supervision guidance.

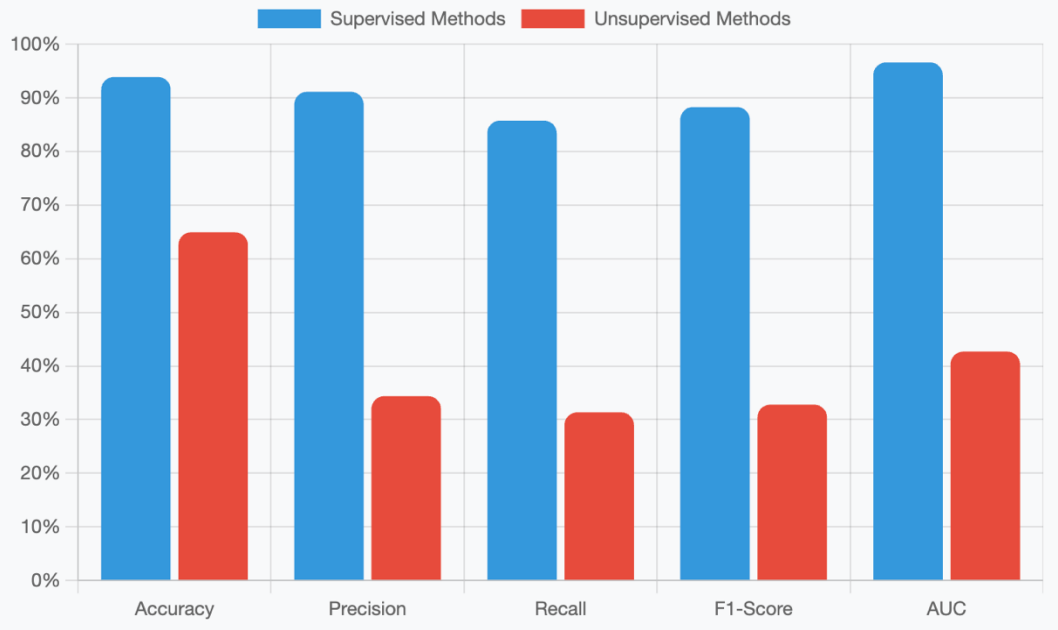


*Figure 5. Supervised and Unsupervised Method Comparison.*

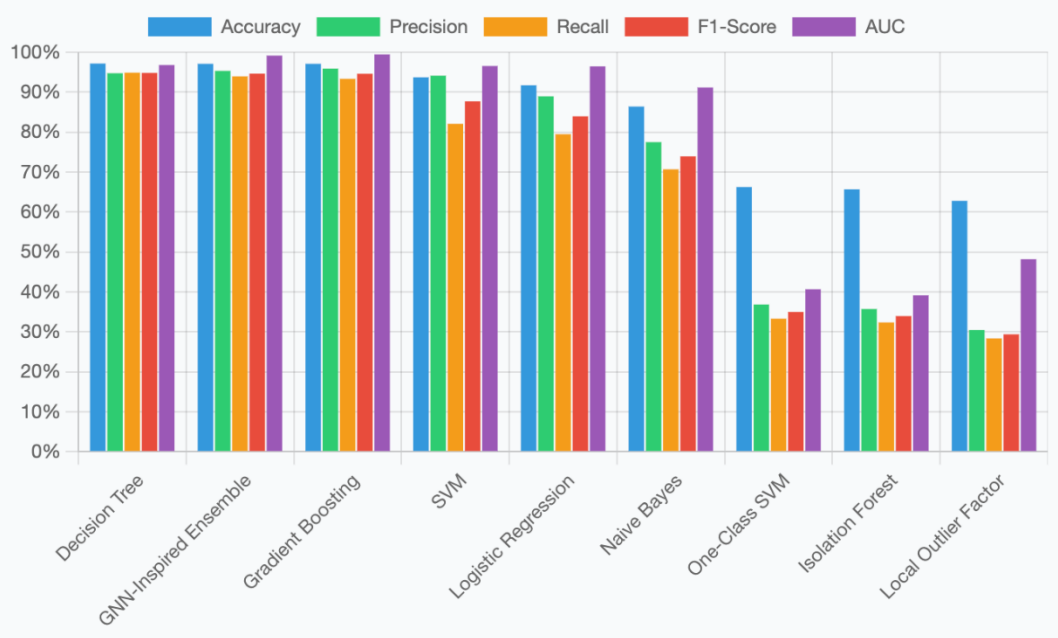


*Figure 6. Heatmap Matrix Performance Comparison.*

## C. Impact of Feature Engineering and Effectiveness of Ensemble

The good performance of supervised method implies the effectiveness and generalization of the graph-inspired feature engineering methodology. The promising

performance achieved by tree-based structures (Decision Tree: 0.9481, Gradient Boosting: 0.9460) and ensemble methods (0.9463) indicates that the engineered features are able to reveal some significant patterns for fraud detection. The balanced performance of the ensemble approach with respect to all three metrics (accuracy: 0.9709, precision: 0.9533, recall: 0.9394) suggests weighted combination sufficiently address weaknesses of the individual models. The high AUC performance of the ensemble (0.9914) indicates that our ensemble returns more trustful probability scores, which is essential for production systems that need to take decisions based on confidence.

With a narrow margin between the ensemble and individual Gradient Boosting (0.0003 F1-score difference), the balance between complexity and benefit has to be considered. Nevertheless, the better overall accuracy (0.9533 vs. 0.9589) and relatively well balanced recall (0.9394 vs. 0.9334) of the ensemble model have justified the complexity cost with improved operational performance, especially in achieving higher true negative rates for better customer experience.

## V. CONCLUSION

In this study, we have shown that graph-motivated feature engineering can be effectively applied in credit card fraud detection. In particular, the proposed approach systematically learns relational patterns from networks of financial transactions in a manner that overcomes shortcomings of existing approaches by combining concepts borrowed from network analysis with established machine learning paradigms. This method reduces the computational time of complete Graph Neural Network (GNN) structures and is practical to apply.

Four main contributions were delivered in this study: the construction of a rich feature engineering pipeline that encompasses customer, merchant, network and temporal characteristics. A multi-dimensional anomaly scoring model that gives interpretable risk scores on weighted feature groups was proposed. Besides, a weighted ensemble approach of Random Forest, Gradient Boosting and Logistic Regression achieved superior AUC with 0.9914. Finally, a balanced sampling scheme was utilized to deal with class imbalance for successful model training. The results demonstrated that our proposed ensemble method obtained the competitive F1-score of 0.9463, indicating a good balance between precision and recall. The above justifies the importance of supervised learning and graph-based features, to mitigate fraud activities on complex financial networks.

The study has several constraints that need to be acknowledged, despite this strong performance. The single-dataset setting of the experiment reduces generalizability assertions and ultimately further validation on a broad range of fraud detection use-cases & financial institutions is required. The empirical value assignment for the weight of both anomaly score and ensemble combination is effective though it needs to be finely tuned by grid search or meta-learning methods further.

The actual experimental in fact uses single train-test split and not cross-validation, which does undermine the robustness of performance estimates. Moreover, the thresholding optimization is only used to our proposed ensemble method; this may lead a conservative estimation of optimal performance for comparison methods.

Directions for future work involve the following: (1) to perform systematic weight tuning through correlation analysis and hyperparameter optimization on both anomaly scoring and ensemble combination; (2) more extensive evaluation across various fraud datasets from different financial institution context in order to establish generalizability; (3) introduction of temporal dynamics modelling such as continuous-time based approaches that can capture new patterns amorphous frauds emerging over time;(4)new advanced ensemble techniques including advanced methods like stacking or dynamic weighting with input characteristics;(5)to analyse semi-supervised learning technique

combining supervised performance benefits together with unsupervised scalability so large-scale systems.

## ACKNOWLEDGEMENT

The guidelines for citing electronic information as offered below are a modified illustration of the adaptation by the International Standards Organization (ISO) documentation system and IEEE style and finalized in Information for IEEE Transactions, Journals, and Letters Authors.

## REFERENCES

[1] I. Ahmad, S. Khan, and S. Iqbal, "Guardians of the vault: unmasking online threats and fortifying e-banking security, a systematic review," *J Financ Crime*, vol. 31, no. 6, pp. 1485–1501, Nov. 2024, doi: 10.1108/JFC-11-2023-0302.

[2] S. Liang *et al.*, "Edge YOLO: Real-Time Intelligent Object Detection System Based on Edge-Cloud Cooperation in Autonomous Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25345–25360, Dec. 2022, doi: 10.1109/TITS.2022.3158253.

[3] S. R. Byrapu Reddy, P. Kanagala, P. Ravichandran, D. R. Pulimamidi, P. V. Sivarambabu, and N. S. A. Polireddi, "Effective fraud detection in e-commerce: Leveraging machine learning and big data analytics," *Measurement: Sensors*, vol. 33, p. 101138, Jun. 2024, doi: 10.1016/j.measen.2024.101138.

[4] F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan, and M. Ahmed, "Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms," *IEEE Access*, vol. 10, pp. 39700–39715, 2022, doi: 10.1109/ACCESS.2022.3166891.

[5] A. Ali *et al.*, "Financial Fraud Detection Based on Machine Learning: A Systematic Literature Review," *Applied Sciences*, vol. 12, no. 19, p. 9637, Sep. 2022, doi: 10.3390/app12199637.

[6] J. Putrevu and C. Mertzanis, "The adoption of digital payments in emerging economies: challenges and policy responses," *Digital Policy, Regulation and Governance*, vol. 26, no. 5, pp. 476–500, Aug. 2024, doi: 10.1108/DPRG-06-2023-0077.

[7] M. Alawida, A. E. Omolara, O. I. Abiodun, and M. Al-Rajab, "A deeper look into cybersecurity issues in the wake of Covid-19: A survey," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, pp. 8176–8206, Nov. 2022, doi: 10.1016/j.jksuci.2022.08.003.

[8] G. Airlangga, "Comparative Analysis of Machine Learning Models for Credit Card Fraud Detection in Imbalanced Datasets," *Journal of Computer Networks, Architecture and High Performance Computing*, vol. 6, no. 2, pp. 858–866, Jun. 2024, doi: 10.47709/cnahpc.v6i2.3816.

[9] S. Almahmoud, B. Hammo, B. Al-Shboul, and N. Obeid, "A hybrid approach for identifying non-human traffic in online digital advertising," *Multimed Tools Appl*, vol. 81, no. 2, pp. 1685–1718, Jan. 2022, doi: 10.1007/s11042-021-11533-4.

[10] R. A. Alzahrani and M. Aljabri, "AI-Based Techniques for Ad Click Fraud Detection and Prevention: Review and Research Directions," *Journal of Sensor and Actuator Networks*, vol. 12, no. 1, p. 4, Dec. 2022, doi: 10.3390/jsan12010004.

[11] I. A. Mondal, Md. E. Haque, A.-M. Hassan, and S. Shatabda, "Handling Imbalanced Data for Credit Card Fraud Detection," in *2021 24th International Conference on Computer and Information Technology (ICCIT)*, IEEE, Dec. 2021, pp. 1–6. doi: 10.1109/ICCIT54785.2021.9689866.

[12] P. Craja, A. Kim, and S. Lessmann, "Deep learning for detecting financial statement fraud," *Decis Support Syst*, vol. 139, p. 113421, Dec. 2020, doi: 10.1016/j.dss.2020.113421.

[13] S. M. Darwish, "A bio-inspired credit card fraud detection model based on user behavior analysis suitable for business management in electronic banking," *J Ambient Intell Humaniz Comput*, vol. 11, no. 11, pp. 4873–4887, Nov. 2020, doi: 10.1007/s12652-020-01759-9.

[14] Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, "Enhancing Graph Neural Network-based Fraud Detectors against Camouflaged Fraudsters," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, New York, NY, USA: ACM, Oct. 2020, pp. 315–324. doi: 10.1145/3340531.3411903.

[15] J. Chung and K. Lee, "Credit Card Fraud Detection: An Improved Strategy for High Recall Using KNN, LDA, and Linear Regression," *Sensors*, vol. 23, no. 18, p. 7788, Sep. 2023, doi: 10.3390/s23187788.

[16] M. Kanchana, R. Naresh, N. Deepa, P. Pandiaraja, and T. Stephan, "Credit Card Fraud Detection Techniques Under IoT Environment: A Survey," in *Transforming Management with AI, Big-Data, and IoT*, Cham: Springer International Publishing, 2022, pp. 141–154. doi: 10.1007/978-3-030-86749-2_8.

[17] C. Kanu *et al.*, "Frauds and forgeries in banking industry in Africa: a content analyses of Nigeria Deposit Insurance Corporation annual crime report," *Security Journal*, vol. 36, no. 4, pp. 671–692, Dec. 2023, doi: 10.1057/s41284-022-00358-x.

[18] N. Kumaraswamy, M. K. Markey, T. Ekin, J. C. Barner, and K. Rascati, "Healthcare Fraud Data Mining Methods: A Look Back and Look Ahead.," *Perspectives in health information managemen*, vol. 19, no. 1, p. 1i, 2022, [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/35440932

[19] C. Zhao, X. Sun, M. Wu, and L. Kang, "Advancing financial fraud detection: Self-attention generative adversarial networks for precise and effective identification," *Financ Res Lett*, vol. 60, p. 104843, Feb. 2024, doi: 10.1016/j.frl.2023.104843.

[20] Y.-S. Ren, C.-Q. Ma, X.-L. Kong, K. Baltas, and Q. Zureigat, "Past, present, and future of the application of machine learning in cryptocurrency research," *Res Int Bus Finance*, vol. 63, p. 101799, Dec. 2022, doi: 10.1016/j.ribaf.2022.101799.

[21] A. Merghadi *et al.*, "Machine learning methods for landslide susceptibility studies: A comparative overview of algorithm performance," *Earth Sci Rev*, vol. 207, p. 103225, Aug. 2020, doi: 10.1016/j.earscirev.2020.103225.