Telecommunications Fraud Detection via Dual Hypergraph Neural Network under Multi-Dimensional Sparsity

Jiyuan Li, Jianwu Dang*, Na Jiang, and Jingyu Yang

School of Electronic and Information Engineering, Lanzhou Jiaotong University No. 88, Anning West Road, Anning District, Lanzhou 730070, P.R. China

Keywords: Telecommunications fraud detection; Graph neural networks; Multi-dimensional sparsity; Dual hypergraph; Class imbalance; Graph augmentation.

Citation: Li J.Y., Dang J.W., Jiang N., et al.: Telecommunications Fraud Detection via Dual Hypergraph Neural Network under Multi-Dimensional Sparsity. Data Intelligence, 2026, Vol. 8, Art. No.: 20260104. DOI: https://doi.org/10.3724/2096-7004.di.2026.0104

ABSTRACT

With the increasing sophistication and concealment of telecommunications fraud, traditional detection technologies face significant challenges. Graph Neural Networks (GNNs) offer a new paradigm for addressing these challenges by concurrently modeling user nodes and their communication links to learn high-order representations of latent interaction patterns. However, their performance in practical applications is severely constrained by the "multi-dimensional sparsity" inherent in telecommunications data: structural sparsity (missing relations), sample sparsity (extreme class imbalance), and semantic sparsity (underutilized edge attributes). To address these issues, this paper proposes a telecommunications fraud detection model based on a Dual Hypergraph Neural Network. First, a relationship-aware graph structure enhancement mechanism is proposed. It reconstructs latent homophilic links via behavioral similarity to mitigate structural sparsity, and subsequently extracts high-quality discriminative subgraphs by evaluating relationship strength through communication intimacy and term frequency-inverse document frequency (TF-IDF) measures. Second, to tackle sample sparsity, an improved data augmentation algorithm incorporating a graph structure expansion mechanism is designed, utilizing feature space interpolation and decoder-based structure reconstruction to mitigate class imbalance. Finally, a Dual Hypergraph Transformation (DHT) mechanism is introduced to map communication edges to hypergraph nodes, enabling the explicit modeling of high-order communication semantics. Experimental results on the test set demonstrate that the proposed model successfully overcomes the zero recall dilemma that plagues traditional baseline models when labels are extremely scarce. By explicitly modeling higher-order interactions, the model achieves a macro F1 score of 46.90% and a fraud recall rate of 69.90%. Compared

*Corresponding author: Jianwu Dang (E-mail: 13909400100@139.com, dangjw@mail.lzjtu.cn).

© 2026 Chinese Academy of Sciences. Published under a Creative Commons Attribution 4.0 International (CC BY 4.0) license.

with other existing approaches, the proposed method shows strong capability in identifying fraudulent users under the challenging conditions of sparse and imbalanced data.

1. INTRODUCTION

WITH the widespread adoption of mobile communication networks and smart terminals, mobile communication has become deeply integrated into every facet of social life, significantly enhancing the efficiency of information dissemination and social interaction. However, the openness and complexity of the communication infrastructure have also provided fertile ground for telecommunications network fraud, causing it to gradually evolve into a significant threat to social security and public property. In recent years, the number of telecommunications fraud cases has continued to rise, with the methods employed demonstrating characteristics of intellectualization, concealment, and cross-domain operations, posing unprecedented challenges to traditional prevention and detection technologies [1-2].

Current methodologies for detecting telecommunications fraud fall into two primary classes: static, rule-based detection methods and behavioral modeling approaches leveraging machine learning. The first class depends on preset features or blacklisting rules (like keyword matching or flagging of abnormal number ranges) to achieve initial filtering [3-7]. Its main characteristics are ease of implementation at the cost of weak generalization. The second class, conversely, utilizes supervised learning algorithms, including support vector machines, decision trees, and random forests, to discern patterns from the behavioral characteristics of individual users [8-12].

While these approaches demonstrated a degree of effectiveness in their initial applications, their suitability has markedly declined as fraudulent activities have become more dynamic and sophisticated. For one, perpetrators circumvent detection by employing tactics like masquerading as legitimate users and rotating through virtual number pools, which quickly renders static rules ineffective [13]. Furthermore, learning models that rely solely on individual user features overlook the intricate network of relationships between users. Consequently, these models fail to capture underlying group behavioral patterns and inter-node correlations, thus constraining their overall detection performance [14].

To overcome these limitations, recent research has pivoted towards Graph Neural Networks (GNNs), which offer a completely new paradigm for addressing the challenges of telecommunications fraud detection [16-17]. This approach concurrently models user nodes and their communication links within a graph-based framework, demonstrating advantages in capturing users' local multi-hop network topology [15]. However, standard GNN models face two major limitations in practical applications. First, their core mechanism inherently relies on node-to-node message passing, strictly limiting them to modeling pairwise relations. In highly camouflaged and organized telecommunications fraud networks, fraudulent behaviors often transcend simple one-to-one interactions, manifesting as "high-order" interaction patterns, such as multi-party collusion and coordinated syndicate operations. Because traditional GNNs lack the capability to directly and jointly model multi-ary relations involving three or more entities, they

inevitably degrade and dilute these group synergistic effects during the aggregation process. Second, alongside the recognized weakness in capturing higher-order dependencies, the inherent multi-dimensional sparsity of telecommunications data severely constrains the performance of standard GNN models. This sparsity is primarily manifested in the following aspects:

- Structural Sparsity: Due to the limitations of data acquisition (e.g., a single carrier can only obtain the communication records of its intra-network users), the constructed communication graph is often incomplete, containing a large number of potential but unobserved edges. This leads to a highly fragmented graph structure where many fraudulent nodes become "islands" or are situated in extremely sparse neighborhoods, making it difficult for the GNN's message-passing mechanism to effectively propagate and aggregate information. [18-20]
- Sample Sparsity: In real-world communication networks, the proportion of explicitly labeled fraudulent users is extremely low, resulting in a severe class imbalance problem. This imbalance is reflected not only in the skewed label distribution but also in the graph's topological structure, manifesting as structural imbalance. Such characteristics make traditional imbalance learning methods difficult to apply directly, as they fail to establish reasonable topological connections for synthesized minority-class samples [21-22]. For GNNs, this dual imbalance in both samples and structure severely undermines the message-passing mechanism, leading the model to be dominated by the majority class and consequently underfitting critical fraud-related patterns [23]. Furthermore, fraudsters often operate in isolated micro-clusters to evade detection. Consequently, in localized topological neighborhoods, the number of ground-truth fraud labels exhibits "extreme label scarcity" (e.g., only a handful of known fraudsters among thousands of normal users). This extreme scenario poses a much more severe challenge than standard class imbalance.
- Semantic Sparsity: Traditional GNN models primarily focus on node attributes and the graph's topological structure, often treating edges as simple connections while neglecting the rich semantic information carried by the communication edges themselves [24]. This sparse representation at the semantic level makes it difficult for the model to fully capture deep features at the behavioral level [25].

Current studies confronting these challenges typically offer solutions that are confined to a single dimension of the problem. For example, researchers have employed oversampling to counteract class imbalance or utilized graph completion techniques to bolster structural connectivity [19, 21]. The fundamental limitation of such methods, however, is their failure to systematically model the underlying interdependencies between structural, sample, and semantic sparsity. As a result, they are ill-equipped to holistically address the multi-dimensional sparsity issue that plagues telecommunications fraud data. For instance, if feature-space interpolation (e.g., GraphSMOTE) is directly applied to an original network characterized by a lack of homophilic connections and a sparse topology, the synthesized virtual minority nodes will be embedded into heterophily-dominated local neighborhoods (i.e., primarily establishing edges with majority class nodes), thereby inevitably introducing heterophilic topological noise. Conversely, if global graph densification strategies (e.g., indiscriminately constructing edges based on feature similarity) are blindly applied without mitigating the extreme class imbalance, the newly added topology will be completely dominated by the overwhelming majority class, making it highly

prone to severe "neighbor explosion" and "over-smoothing" phenomena. Consequently, a naive pipeline integration (i.e., indiscriminately stacking data resampling and graph structure learning tools in sequence) not only fails to resolve the issue, but may instead lead to a significant degradation in the model's discriminative ability for anomalous nodes.

To address this research gap, this paper proposes a telecommunications fraud detection model based on a Dual Hypergraph Neural Network. The model is designed with a multi-level mechanism for structure completion and feature enhancement, aiming to alleviate the learning difficulties caused by sparsity from the three dimensions of graph structure, sample distribution, and semantic representation. The main contributions of this paper are as follows:

1) We propose a unified graph learning framework that seamlessly integrates relationship densification, rare-class expansion, and hypergraph representation learning. Unlike existing approaches that treat structural sparsity, sample sparsity, and semantic sparsity in isolation, our framework explicitly models their theoretical interdependencies and provides a collaborative optimization strategy: structural refinement guides reliable rare-class expansion, which in turn facilitates effective higher-order semantic extraction.
2) The framework adopts a densification-then-refinement graph optimization strategy. First, it enhances the structural integrity of the graph through relation completion. Then, it innovatively introduces a computation method based on communication intimacy and the TF-IDF principle to evaluate the strength of relationships. On this basis, high-quality subgraphs are extracted, enabling the model to focus on interaction relationships with discriminative value.
3) To overcome the fundamental limitations of traditional graph models in handling semantic sparsity and their inherent inability to capture higher-order relations beyond pairwise interactions, we introduce a Dual Hypergraph Neural Network. By elevating communication "edges" to primary learning nodes and constructing hyperedges to encompass multi-party interactions, this mechanism explicitly breaks the pairwise bottleneck, enabling the model to directly learn high-order interaction patterns present in coordinated fraud behaviors.

2. RELATED WORK

2.1 Structural Sparsity

In graph-structured data, the constructed graph often suffers from a significant number of latent but unobserved edges due to limited collection scope or missing observational information. This leads to a highly sparse and fragmented overall structure. Such structural sparsity causes some nodes to become isolated or situated in sparse neighborhoods, thereby restricting the effectiveness of message passing and feature aggregation in Graph Neural Networks (GNNs).

To address the challenges of graph sparsity, a primary research direction has been Graph Completion, which focuses on enhancing graph connectivity by predicting latent edges. The central idea is to infer these missing links from the existing graph structure and node features. Pioneering this area, the

Graph Autoencoder (GAE) and Variational Graph Autoencoder (VGAE) by Kipf and Welling learn low-dimensional node embeddings via a GNN encoder and then reconstruct the adjacency matrix with a simple decoder (e.g., inner product) to perform link prediction [19]. Subsequent works have built upon this framework. For instance, the Variational Graph Normalizing Auto-Encoder (VGNAE) by Salha et al. incorporates Normalizing Flows to model a more expressive posterior distribution over node representations, thus improving the accuracy of graph completion [26]. Other notable approaches include the Adversarially Regularized Graph Autoencoder (ARGA), which uses adversarial training to regularize the latent space and improve robustness [27]. Meanwhile, GraphRNN (often referred to as GraphRNA in early drafts) takes inspiration from NLP, modeling graph generation as a sequential process with an RNN to dynamically complete the graph [28]. These diverse approaches have significantly advanced the field of graph completion for sparse data.

2.2 Sample Sparsity

In real-world communication networks, the proportion of explicitly labeled fraudulent users is extremely low, often constituting less than 0.01% of the total samples. This creates a severe class imbalance problem. Such sample sparsity leads to the GNN training process being dominated by normal users, causing the model to develop a majority class bias and resulting in a low identification capability for fraudulent nodes.

To mitigate this problem, researchers have proposed various imbalanced learning strategies tailored for graph data. Zhao et al. proposed GraphSMOTE, which generates synthetic minority-class nodes and their corresponding edge structures in the graph embedding space to enhance the model's perception of the minority class [21]. ImGAGN (Imbalanced Graph Adversarial Generative Network) combines a generative adversarial mechanism with balanced sampling to produce more discriminative minority-class representations [22]. Furthermore, Collaboration-Aware Relational GNN (CARE-GNN), proposed by Dou et al., reduces interference from the majority class through relation selection and neighbor re-weighting mechanisms, significantly improving the minority class recall in fraud detection tasks [13]. Graph Consistency Learning (GCL), by Zheng et al., utilizes consistency constraints to maintain the semantic consistency of samples under label-scarce conditions, effectively improving the generalization performance of minority-class nodes. Recently, to bypass the limitations of traditional node-level resampling, advanced topological augmentation frameworks like BAT [42] have been proposed to mitigate class-imbalance bias by rectifying topological asymmetries directly without class rebalancing. However, in extreme label scarcity scenarios where minority nodes are severely isolated, purely topological approaches often struggle to establish reliable initial connectivity.

2.3 Semantic Sparsity

Traditional GNNs often treat graph edges as unweighted or equally weighted connections, neglecting the rich semantic features contained within communication behaviors. This semantic sparsity forces the model to rely solely on graph structure and node attributes, making it difficult to capture deep, behavior-level fraud patterns.

To address this problem, Gong et al. proposed the Edge-enhanced Graph Neural Network (EGNN), which incorporates edge feature embeddings into the message-passing process, enabling the model to dynamically adjust neighbor weights based on communication intensity or frequency [24]. The Relational Graph Attention Network (conceptually similar to Heterogeneous Graph Transformer by Hu et al.) introduces edge types and relational semantics into the attention mechanism to achieve discriminative modeling of different communication behaviors, such as outgoing calls, incoming calls, and text messages [29]. Furthermore, the Hypergraph Neural Network (HGNN) proposed by Feng et al. represents multiparty communication acts as hyperedges, which are capable of capturing high-order interactions among multiple nodes [30].

Furthermore, recent state-of-the-art anomaly detection models have adopted various approaches. For instance, CAREGNN [13] utilizes reinforcement learning to enhance camouflage resistance; BWGNN [43] captures high-frequency anomalous signals in the spectral domain; and CaT-GNN [44] enhances credit card fraud detection via causal temporal graph neural networks. Nevertheless, in highly fragmented telecommunications networks, the underlying assumptions of these models, such as sufficient initial homophilic connectivity or densely populated temporal interactions often fail to hold, exposing them to the vulnerabilities of multi-dimensional sparsity.

3. METHODOLOGY

3.1 Overview

As illustrated in Figure 1, the proposed framework constitutes a systematic pipeline for detecting telecommunications fraud using Call Detail Record (CDR) data. The process begins with Data Preprocessing and Feature Engineering, where raw metadata is processed into standardized behavioral vectors via statistical extraction and CDF normalization. To optimize network topology, the model employs a Graph Structure Enhancement strategy based on behavioral similarity to infer latent connections. Furthermore, Relationship Strength Modeling and Subgraph Extraction are introduced; this module integrates temporal dynamics and contextual importance to evaluate interaction quality, extracting communication subgraphs characterized by high intensity and reciprocity. To address class imbalance, a Graph Augmentation Strategy is applied, synthesizing minority nodes and reconstructing structural connections via a learnable decoder. Finally, the refined graph is processed by a Dual Hypergraph Neural Network, which utilizes an edge-centric message-passing mechanism to capture high-order interaction semantics for accurate node classification.

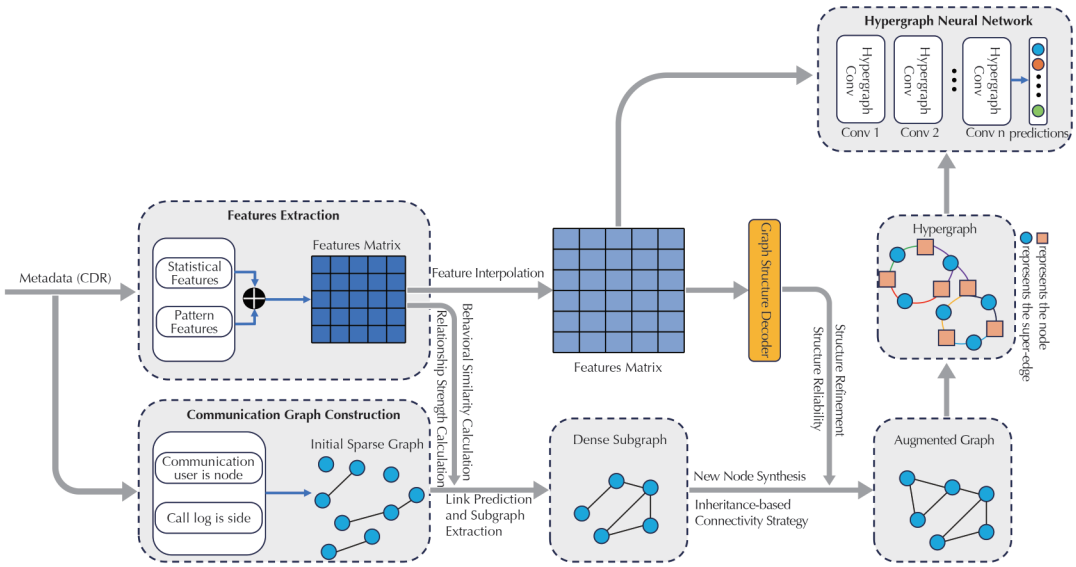


Figure 1. Overview of the proposed model.

At the same time, these three components are not simply stacked but organically integrated to work in concert. Relationship-aware structural densification serves as a prerequisite, establishing reliable homophilic bridges. This structural foundation effectively prevents the subsequent GraphSMOTE module from generating noisy, heterogeneous connections during the synthetic node interpolation process. In turn, the decoder-based Synthetic Minority Over-sampling Technique (SMOTE) balances class distributions and enriches the topological connectivity of minority classes. Finally, it is precisely this densified and category-balanced structural input that enables the Dual Hypergraph Neural Network to extract complex higher-order semantic features, no longer constrained by the original pairwise sparsity or suppressed by majority-class bias. Through this cyclic enhancement mechanism, the framework addresses sparsity issues at the structural, sample, and semantic levels in a holistic manner.

3.2 Data Preprocessing and Feature Engineering

The dataset utilized in this experiment is the Call Detail Record (CDR), a fundamental data source for detecting telecommunications network fraud. CDRs are a type of metadata generated within an operator's core network, documenting various behavioral characteristics of users during their communication activities. It does not contain the communication content itself but rather depicts users' communication behaviors by logging fields such as the called party, call duration, and call type. This characteristic of CDR data makes it highly valuable for user behavior modeling, particularly for identifying anomalous activities like telecommunications fraud. By statistically analyzing user communication behaviors within specific time windows, it is possible to effectively uncover their behavioral patterns, providing core data support

for building fraud detection models. However, raw CDR data is unsuitable for direct input into machine learning models due to issues such as high dimensionality, noise, and skewed distributions. Therefore, a series of data preprocessing steps must be undertaken before model training to ensure the data's quality and suitability.

First, feature engineering is performed on the raw metadata to extract statistical and pattern-based features that are conducive to modeling.

The statistical features primarily include metrics such as: mean, median, mode, standard deviation, variance, range, maximum, minimum, sum, frequency counts, and various quantiles. These features can reflect the data's central tendency, degree of dispersion, and distribution shape from multiple perspectives, which facilitates the subsequent identification of anomalous communication patterns or group characteristics. Let the raw dataset be denoted as:

$$D = \left\{ x_v^k \mid v \in V,\ k = 1,\ 2,\ \ldots,\ K \right\} \tag{1}$$

where V is the set of all user nodes, K is the number of metadata dimensions for each user, and x_v^k represents the set of observations for a user $v \in V$ along the k-th dimension. For each node $v \in V$ and each metadata dimension k, we generate a statistical description vector, denoted as $\mathbf{S}_v^{(k)}$. This vector is derived by applying a collection of statistical functions to the set of observed values $\mathbf{x}_v^{(k)}$. The process can be formalized as:

$$\mathbf{S}_v^{(k)} = [f_1(\mathbf{x}_v^{(k)}), f_2(\mathbf{x}_v^{(k)}), \ldots, f_m(\mathbf{x}_v^{(k)})] \tag{2}$$

where each f_i is a specific statistical function (e.g., f_1 = mean, f_2 = std) from the set of functions mentioned previously, and m is the total number of statistical features extracted.

Pattern features are a class of important features used to describe the behavioral regularities and habits of entities. They are typically closely related to the domain knowledge of a specific problem, endowing them with strong specificity and discriminative ability. In contrast to statistical features, which primarily focus on numerical distributions and fluctuations, pattern features are more oriented towards capturing the structural and periodic characteristics of behavior across dimensions such as time, category, and frequency.

In telecommunications network data, a user's behavioral patterns often reflect the latent features behind their communication activities, thus playing a critical role in the identification of anomalous behaviors (such as telecommunications fraud). Typical behavioral pattern features include, but are not limited to, the following categories: Active time periods describe the primary time ranges during which a user's communications are concentrated, capturing the temporal rhythm of daily interactions. Active days measure how many days a user remains active within a given time window, indicating the continuity and

stability of their communication habits. Call-type distribution characterizes the proportions of outgoing versus incoming calls and the distribution across different call categories to reveal structural patterns in communication behavior. Communication behavior periodicity examines whether activities like calls or text messages exhibit recurring temporal patterns.

To address the significant differences in numerical scales and statistical distributions among features, and to prevent certain features from unduly influencing model training and degrading performance, this study employs Cumulative Distribution Function (CDF) normalization after feature extraction. The core idea of this method is to map any feature value to its quantile in the empirical distribution, that is, the proportion of samples in the dataset that are less than or equal to that valuethereby unifying the value range of all features to the [0, 1] interval. For an arbitrary feature x, its normalized form is:

$$x' = F(x) = \frac{1}{n}\sum_{i=1}^{n} I(x_i \leq x) \tag{3}$$

where $I(\cdot)$ is the indicator function, and n is the total number of samples. The normalized result x' is in the range [0, 1], ensuring that the contribution of each feature is balanced during model training.

Compared to traditional methods like Z-score standardization, which rely on mean and standard deviation, CDF normalization does not depend on specific data distribution assumptions, possessing a distribution-free property. This characteristic allows it to exhibit superior robustness when handling data containing outliers or skewed distributions, enabling a more stable and equitable characterization of a sample's relative position within the population.

3.3 Graph Structure Enhancement

In practical communication scenarios, calls and SMS messages are no longer the sole means of interaction. As fraud techniques constantly evolve, malicious actors increasingly utilize cross-platform encrypted communications to bypass traditional surveillance. Furthermore, the Call Detail Records (CDR) utilized in this study are sourced from a single operator. Since users freely operate across multiple carriers in reality, the captured communication network inevitably suffers from observational missingness. Due to these objective limitations, the topological graph constructed directly from physical interactions exhibits high sparsity.

To mitigate this topological incompleteness, we propose a graph structure enhancement strategy based on behavioral similarity. The core idea is that while fraud conspirators may deliberately avoid direct communication to evade detection (thereby forming isolated topological islands), they often exhibit highly consistent behavioral patterns in the feature space. By inferring latent connections between nodes with similar behaviors, we construct a reliable foundation of homogeneous structure. This step serves as a critical prerequisite for subsequent graph augmentation modules, as it prevents the generation of significant heterogeneous noise during the process of synthesizing and interpolating nodes.

Regarding the similarity metric, this paper adopts the Euclidean distance (L2 distance). Compared to other common metrics such as L1 distance and cosine similarity, the L2 distance offers advantages including computational simplicity, an intuitive meaning, and stable performance across various similarity tasks [31-32]. Therefore, it was selected as the primary metric for this paper.

After obtaining the L2 distance between nodes, the similarity between node v_i and node v_j can be defined as follows, where σ is the bandwidth parameter:

$$S(v_i, v_j) = \exp\left(-\frac{\text{Distance}(v_i, v_j)^2}{2\sigma^2}\right) \tag{4}$$

Once the similarity calculation is complete, the dense communication graph is constructed based on a link prediction method. The specific procedure is as follows: a similarity threshold θ is predefined. When the similarity between two nodes is greater than this threshold, an edge is established between them; otherwise, no edge is created. This is formally described as:

$$L(v_i, v_j) = \begin{cases} 1, & \text{if } S(v_i, v_j) > \theta \\ 0, & \text{if } S(v_i, v_j) \le \theta \end{cases} \tag{5}$$

By introducing edges based on behavioral similarity, an incomplete topological fragment is ultimately reconstructed into a more structurally coherent whole. On this reconstructed graph, information can propagate effectively between nodes, enabling the detection model to learn high-quality node embeddings capable of distinguishing between different node groups based on rich neighborhood information.

3.4 Relationship Strength Modeling and Subgraph Extraction

Given that urban communication social networks contain a massive number of user nodes and communication edges, performing computations directly on the complete network would not only incur extremely high computational overhead but also introduce interference into the model training process due to the presence of a large number of irrelevant or weakly correlated nodes. This can reduce the efficiency and accuracy of anomalous user identification. To address this, this paper first introduces a subgraph extraction strategy to reduce the computational scale and enhance the effectiveness of the training data.

In the subgraph construction process, this paper adopts an extraction method based on relationship strength. Relationship strength was chosen as the core criterion for subgraph filtering for two primary reasons. On the one hand, fraudulent users typically struggle to form long-term, stable connections with legitimate users. Their interactions within the communication network are often characterized by short durations, low frequencies, and scattered distributions, leading to a significantly lower relationship strength with other users compared to that of normal users [33-34]. On the other hand, incorporating a relationship strength metric at the subgraph extraction stage not only accentuates the interactional

differences between users but also preserves those interactive relationships that hold discriminative value, all while reducing computational complexity [35].

When evaluating the relationship strength between users, although methods based on global network topology or random walks could be employed, they would introduce a substantial computational burden when processing massivescale communication data. This contradicts the real-time requirements of many application scenarios. Concurrently, we observe that interactions in communication networks exhibit distinct locality characteristics: users are more inclined to establish connections with individuals who are geographically close or share similar calling habits. This pattern suggests that measuring relationship strength based on users' local information and individual behavioral features is not only more computationally feasible but also more aligned with authentic user social patterns [36].

Therefore, to accurately and efficiently measure the relationship strength between users, this paper employs a novel model. This model not only evaluates the communication behaviors themselves but also innovatively incorporates two additional dimensions: temporal dynamic evolution and the importance of the social context. Its core lies in the mathematical formalization of the influence of a single communication event. For a call initiated by user u and received by user v, the relationship strength perceived by the receiver v, denoted as $H_{v\leftarrow u}$, is defined as:

$$H_{v\leftarrow u} = w \cdot \log_2(d+1) \cdot e^{-\beta \cdot \Delta t} \times (1 + \mathrm{TF} - \mathrm{IDF}_{v\leftarrow u}) \tag{6}$$

where w is the weight coefficient, d is the call duration in seconds, β is the time decay factor, and Δt represents the time interval.

The formula consists of two key parts:

- **Basic Interaction Strength:** The first part of the expression quantifies the interaction strength inherent in the communication event itself, evaluating the contribution of the call duration d (Its unit is seconds) through a weighted logarithmic model. The exponential decay term, $e^{-\beta \cdot \Delta t}$, then models the natural cooling of relationship intensity over the time interval (Δt), ensuring that recent interactions are given a higher weight.
- **Contextual Importance Weighting:** This part is used to evaluate the relative importance of a communication event within its specific social context, in order to adjust the basic interaction strength. The $\mathrm{TF}-\mathrm{IDF}_{v\leftarrow u}$ factor, inspired by the field of information retrieval [37], is used to measure the focus and scarcity of user u's calling behavior within user v's communication network. It is calculated as follows:

  TF: This measures the proportion of calls from user u among all calls received by user v:

$$\mathrm{TF}_{uv} = \frac{n_{uv}}{N_v}$$

  where n_{uv} is the number of calls initiated by user u to user v in a specific period, and N_v is the total number of calls received by user v in that period.

IDF: This measures the specificity of user u's communication behavior:

$$\mathrm{IDF}_u = \log \frac{|P|+1}{p_u}$$

where $|P|$ is the total number of users considered in the period, and p_u is the number of distinct users that user u has contacted. If user u's communication behavior is highly concentrated on a few users, the IDF value for interactions with these users will be higher.

To perform the subgraph filtering process, we aggregate the overall relationship strength of each edge based on the calculated relationship strengths. We then prune edges from the network that exhibit weak connections—specifically, those in the bottom 10% of the relationship strength distribution, which typically represent noise or random contacts. Upon completing this filtering process, we obtain an optimized and relatively compact relationship network. Finally, using this refined network as a basis and centering on the target user, we extract their k-hop neighbors to generate the final communication subgraph for downstream tasks.

3.5 Graph Augmentation Strategy for Imbalanced Node Classification

In real-world graph application scenarios such as telecommunications fraud detection, node class imbalance is a prevalent and highly challenging problem. Statistical analysis of the real-world telecommunications network dataset used in this study reveals that fraudulent user nodes account for only about 0.2%–1% of the total nodes, forming a typical, extremely imbalanced distribution. This imbalance causes traditional Graph Neural Networks (GNNs) to be dominated by the majority class (benign users) during the training process. As a result, the feature patterns of the minority class (fraudulent users) are difficult to be fully learned and accurately identified.

Unlike traditional tabular data, the imbalance in graph data manifests not only in the skewed distribution of node labels but also in the topological characteristics of the graph. Majority-class nodes typically form dense community structures within the network, whereas minority-class nodes often occupy sparse and marginalized positions. This structural imbalance further weakens the effectiveness of GNN message passing, causing the information received by minority-class nodes to be overwhelmed by majority-class features, thereby exacerbating the classification bias.

To address the aforementioned challenges, this paper proposes an improved end-to-end method based on graph structure augmentation and bilinear decoding joint modeling, building upon the GraphSMOTE framework [21, 38]. This method performs synergistic optimization from two aspects—graph data augmentation and structural reconstruction of the representation space—to achieve a complete modeling process from synthetic node generation to graph reconstruction.

Specifically, we optimized the graph data augmentation component. The original GraphSMOTE method performs random interpolation within the node embedding space learned by a GNN. However, we identified a flaw in this strategy, particularly in scenarios of extreme class imbalance: because the training process is dominated by the vast majority of normal samples, the embedding space learned by the GNN is itself severely skewed. When interpolation is performed in such a space, the features of the synthesized minority-class samples will inevitably drift towards the majority-class centroid. This not only severely undermines the intended effect of data augmentation but may even introduce spurious noise into the training process. To this end, this paper proposes an improved solution of performing random interpolation directly in the original feature space. The main process of the improved method is as follows:

First, the minority-class nodes are identified from the training set to form a minority node set, denoted as:

$$V_{\min} = \{v \mid v \in V_{\text{train}}, y_v \in C_{\min}\} \tag{7}$$

where $C_{\min}$ represents the set of minority-class labels.

For each minority-class node $v_i \in V_{\min}$, the Euclidean distance between it and all other minority-class nodes is calculated in the feature space:

$$d_{i,j} = \left\| \mathbf{X}_i - \mathbf{X}_j \right\|_2, \quad \forall v_j \in V_{\min} \tag{8}$$

Then, based on the calculated Euclidean distances, the nearest neighbor node $v_{\text{nn}(i)}$ in the feature space is selected for each node v_i:

$$v_{\text{nn}(i)} = \arg \min_{\substack{v_j \in V_{\min} \\ j \neq i}} d_{i,j} \tag{9}$$

A new node feature is generated via linear interpolation between the node and its nearest neighbor:

$$\mathbf{X}_{\text{new}} = \mathbf{X}_i + \alpha \cdot (\mathbf{X}_{\text{nn}(i)} - \mathbf{X}_i) \tag{10}$$

where $\alpha \sim U(0, 1)$ is a random interpolation coefficient.

The newly generated node features are added to the node set to form an expanded set of nodes, laying the foundation for subsequent graph structure expansion and decoding reconstruction.

After new minority-class nodes are generated, the core challenge in achieving graph structure-level data augmentation is how to construct topologically consistent connections for them. To this end, this study introduces a Graph Structure Expansion Mechanism, aimed at generating structurally plausible adjacency relationships for the synthetic nodes while preserving the topological properties of the original graph.

The schematic illustration of this strategy is presented in Figure 2, which visually outlines the progression from feature synthesis to structure inheritance and final refinement.

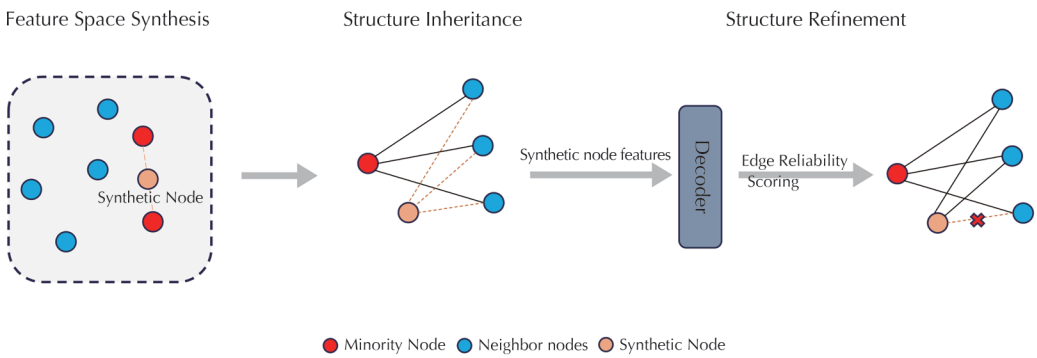


Figure 2. Illustration of the Graph Structure Augmentation Strategy. The process is composed of three phases: (a) Feature Space Synthesis: A synthetic minority node (beige) is generated via linear interpolation between a seed minority node (red) and its nearest minority neighbor within the feature space. (b) Structure Inheritance: The synthetic node tentatively inherits the topological connections (dashed orange lines) from its seed node to the local neighborhood (blue nodes). (c) Structure Refinement: A learnable structure decoder evaluates the reliability of the inherited connections. Edges with high reliability scores are retained as solid lines, while implausible connections are filtered out.

First, we formally model the expanded graph. Let the adjacency matrix of the original graph $G = (V, E)$ be $\mathbf{A} \in \{0, 1\}^{N \times N}$, where N denotes the number of nodes. After performing SMOTE oversampling, N_{syn} synthetic nodes are generated, resulting in the expanded graph $G' = (V', E')$, whose adjacency matrix $\mathbf{A}_{\text{ext}} \in \{0, 1\}^{(N+N_{\text{syn}}) \times (N+N_{\text{syn}})}$ has the following block structure:

$$\mathbf{A}_{\text{ext}} = \begin{bmatrix} \mathbf{A}_{\text{orig}} & \mathbf{A}_{\text{orig, syn}} \\ \mathbf{A}_{\text{syn, orig}} & \mathbf{A}_{\text{syn, syn}} \end{bmatrix} \tag{11}$$

where $\mathbf{A}_{\text{orig}}$ represents the adjacency relationships among the original nodes; $\mathbf{A}_{\text{orig, syn}}$ and $\mathbf{A}_{\text{syn, orig}}$ represent the connections between the original and synthetic nodes; $\mathbf{A}_{\text{syn, syn}}$ reflects the internal connection structure among the synthetic nodes.

Subsequently, to ensure the plausibility and interpretability of the expanded graph structure, this paper adopts an Inheritance-based Connectivity Strategy. This strategy is centered on the "seed nodes" of each synthetic node. By inheriting and reconstructing the adjacency relationships of the original graph, it simultaneously preserves both topological consistency and local connectivity. The specific rules are as follows:

- Direct Inheritance: Each synthetic node v_i^{syn} is derived from a corresponding seed node v_j^{seed}. It first inherits all the adjacency relationships of its seed node from the original graph. Formally, if an edge exists between v_j^{seed} and another node v_k in the original graph, then a new connection (v_i^{syn}, v_k) is established in the expanded graph:

$$(v_j^{\text{seed}},\ v_k) \in E \quad \Rightarrow \quad (v_i^{\text{syn}},\ v_k) \in E' \tag{12}$$

where E and E' denote the edge sets of the original and expanded graphs, respectively.

- Symmetry Preservation: For an undirected graph, the adjacency matrix $\mathbf{A}_{\text{ext}}$ must satisfy $\mathbf{A}_{\text{ext}} = \mathbf{A}_{\text{ext}}^{\top}$, which implies that $\mathbf{A}_{ij} = \mathbf{A}_{ji}$ holds for all node pairs. During the connection generation process, this property is enforced through a forced symmetric update operation, ensuring that the expanded graph remains strictly undirected. Formally, for every assigned edge:

$$\mathbf{A}_{\text{ext}}(i,\ j) = 1 \quad \Rightarrow \quad \mathbf{A}_{\text{ext}}(j,\ i) = 1 \tag{13}$$

- Seed-dependent Linking: An edge is created between two synthetic nodes if their respective seed nodes are adjacent in the original graph. This rule forms "inheritance-based edges" among synthetic nodes, thereby preserving local structural coherence.

This "connection replication" strategy is based on a core assumption: nodes that are similar in feature space (i.e., a seed node and its derived synthetic node) should also play similar roles in the graph structure. However, the connections generated by such a heuristic-based rule are inherently static and suboptimal. While it provides a set of plausible candidate connections for the newly generated nodes, it fails to fully leverage the data's own distributional properties to validate the quality of these connections. In other words, the strategy's effectiveness is highly dependent on the strength of its core assumption. In complex graphs, this assumption may not fully hold, and it could even lead to the indiscriminate propagation of structural noise from seed nodes to synthetic nodes.

To learn the intrinsic relationship between node features and the graph topology, we designed a graph structure decoder, $D()$. This decoder aims to reconstruct the graph's adjacency relationships solely from node features.The structure of the decoder D is as follows:

$$D(\mathbf{X}_{\text{new}}) = \sigma\left(\mathbf{X}_{\text{new}}\mathbf{W}_d\left(\mathbf{X}_{\text{new}}\mathbf{W}_d\right)^{\top}\right) \tag{14}$$

where $\mathbf{W}_d \in \mathbb{R}^{F\times F}$ is a learnable parameter matrix, and $\sigma(\cdot)$ denotes the Sigmoid activation function. The output of the decoder, $\mathbf{P} \in \mathbb{R}^{(N+N_{\text{syn}})\times(N+N_{\text{syn}})}$ is a probabilistic adjacency matrix, where each entry P_{ij} represents the predicted probability that an edge exists between nodes i and j based on their feature similarity.

To train the decoder, we introduce a Structure Reconstruction Loss (L_{rec}). In this loss function, instead of using the expanded adjacency matrix—which may contain noisy or unreliable edges—as the supervision signal, we utilize the adjacency matrix of the original graph $\mathbf{A}$ to guide the learning process. This ensures that the decoder focuses on reconstructing the intrinsic and stable structural patterns of the original network. The loss function is defined as:

$$L_{\text{rec}} = \text{MSE}\left(\mathbf{P}_{1:N,\ 1:N}, \mathbf{A}\right) \tag{15}$$

where, $\mathbf{P}_{1:N,\ 1:N}$ denotes the submatrix of the predicted probability matrix that corresponds exclusively to the nodes of the original graph. By minimizing the reconstruction loss L_{rec}, the decoder learns an effective mapping from the node feature space to the structural space of the original graph. This enables the decoder to evaluate the plausibility of a connection between any pair of nodes.

Once the decoder is trained, we utilize it to optimize the initial expanded graph. The decoder's output probability matrix $\mathbf{P}$ is converted into a binary decision matrix $\mathbf{P}_{\text{filter}}$ using a decoder filtering threshold τ:

$$(\mathbf{P}_{\text{filter}})_{ij} = \begin{cases} 1, & \text{if } P_{ij} \geq \tau \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

Afterward, this filter is multiplied element-wise with the expanded graph's adjacency matrix (denoted as $\mathbf{A}_{\text{template}}$) to adjust and optimize the connections generated during the oversampling stage:

$$\hat{\mathbf{A}} = \mathbf{A}_{\text{template}} \odot \mathbf{P}_{\text{filter}} \tag{17}$$

where $\odot$ denotes the Hadamard product. This operation ensures that a final edge must satisfy two conditions simultaneously it must be a candidate edge proposed by the initial oversampling strategy, and it must also be considered structurally plausible by our trained, data-driven decoder.

The resulting $\hat{\mathbf{A}}$, together with the augmented features $\mathbf{X}_{\text{new}}$, jointly constitute the final augmented graph used for downstream GNN model training.

3.6 Dual Hypergraph Neural Network

In communication networks, edges are the core elements that carry key interaction semantics, playing a crucial role in downstream tasks such as fraud detection. However, existing Graph Neural Networks (GNNs) tend to be node-centric in their design, which leads to implicit and inefficient edge representation learning. Therefore, we employ a Dual Hypergraph Transformation (DHT) mechanism, a paradigm shift that redefines edges as the primary learning entities rather than mere connectors. DHT redefines the roles of network elements by mapping the edges of the original graph to nodes in a dual hypergraph, and the original nodes to hyperedges. The core advantages of this mechanism are: First, it allows mature node-centric message-passing mechanisms to be directly applied to edge representation learning, thereby efficiently capturing high-order semantic associations. At the same time, as an information-lossless

bijective transformation, DHT guarantees the topological integrity of the original interaction patterns, providing an ideal structural input for subsequent graph convolution operations.

To facilitate the understanding of this paradigm shift, Figure 3 illustrates the Dual Hypergraph Transformation process. As shown in the figure, transposing the incidence matrix **H** naturally induces the dual hypergraph structure, where the semantically rich edges in the original graph view are transformed into learnable nodes in the dual view.

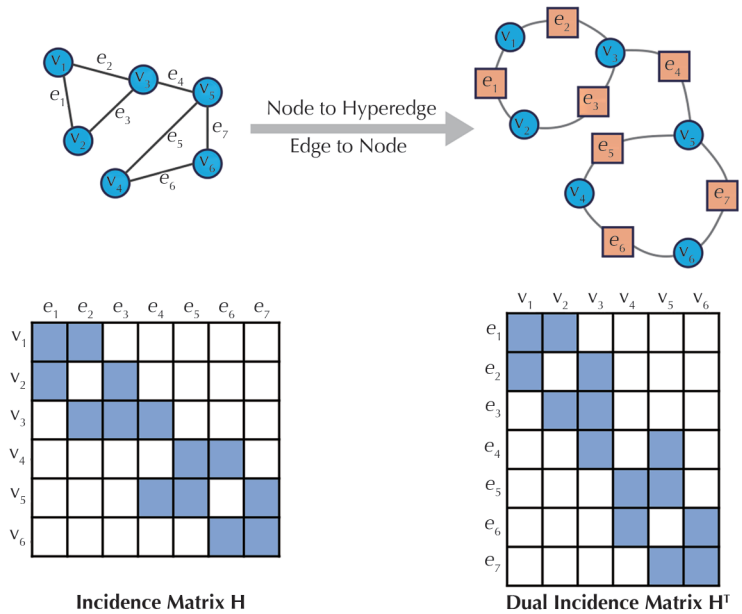


Figure 3. Illustration of the Dual Hypergraph Transformation (DHT). The original graph and its incidence matrix **H** (left) are transformed into a dual hypergraph and dual incidence matrix $\mathbf{H}^T$ (right). Edges in the original view become nodes (orange squares) in the dual view.

Formally, given an original graph $G = (V, E)$, the Dual Hypergraph Transformation (DHT) maps it to a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, with the mapping relationships defined as follows:

Vertex Mapping: The vertex set $\mathcal{V}$ of the hypergraph is constructed from the edge set E of the original graph. That is, each edge $e \in E$ in the original graph corresponds to a unique vertex $v_e \in \mathcal{V}$ in the hypergraph. Therefore, the vertex set can be expressed as:

$$\mathcal{V} = \{v_e \mid e \in E\} \tag{18}$$

Hyperedge Mapping: The hyperedge set $\mathcal{E}$ of the hypergraph is induced by the vertex set V of the original graph. Specifically, each vertex $v \in V$ in the original graph corresponds to a hyperedge $h_v \in \mathcal{E}$. This hyperedge consists of all the vertices in $\mathcal{V}$ that correspond to the edges incident to v in G. It is formally defined as:

$$h_v = \{v_e \in \mathcal{V} \mid e \text{ is incident to } v \text{ in } G\} \tag{19}$$

On the hypergraph structure constructed via the Dual Hypergraph Transformation (DHT), we employ a widely validated two-stage message-passing paradigm from the field of hypergraph learning to achieve efficient edge representation learning. This paradigm is fundamentally different from the direct "node-to-node" information propagation model in traditional Graph Neural Networks (GNNs). On our dual hypergraph, a complete hypergraph convolution operation is naturally decomposed into a two-stage message-passing process.

The core objective of this design is to migrate and adapt mature node-level message-passing mechanisms to the task of edge representation learning, thereby generating richer and more context-aware representations for each edge in the original graph.

Formally, let the node feature matrix of the dual hypergraph at layer l be $\mathbf{X}_H^{(l)} \in \mathbb{R}^{|E| \times d_l}$, $|E|$ is the number of edges in the original graph, and d_l is the feature dimension at layer l.

Node-to-Hyperedge Aggregation: In the dual hypergraph, the feature of each hyperedge e_j^H (corresponding to a node in the original graph) is not predefined. Instead, it is obtained by aggregating the features of all dual-hypergraph nodes contained in it. Formally, this process is defined as:

$$m_{e_j}^{(l)} = \sum_{v_i^H \in e_j^H} \frac{1}{e_j^H d_{v_i^H}} \mathbf{X}_H^{(l)}(i) \mathbf{W}^{(l)} \tag{20}$$

where $W^{(l)} \in \mathbb{R}^{d_l \times d_{l+1}}$ is the learnable linear transformation for layer l, used to project features from dimension d_l to d_{l+1}. Here, $|e_j^H|$ is the number of nodes contained in the hyperedge e_j^H, and $d_{v_i^H}$ is the number of hyperedges associated with node v_i^H. The normalization factor $|e_j^H| \cdot d_{v_i^H}$ balances contributions from hyperedges of different sizes and nodes with varying degrees, preventing numerical bias and improving stability.

Hyperedge-to-Node Propagation: After obtaining the aggregated representation $m_{e_j}^{(l)}$ for each hyperedge, this information is propagated back to the nodes of the dual hypergraph to achieve higher-order information fusion. For any dual hypergraph node v_i^H (which corresponds to an edge in the original graph), its update at layer $l+1$ is:

$$X_H^{(l+1)}(i) = \sigma\left(\sum_{e_j^H \ni v_i^H} \frac{1}{d_{v_i^H}} m_{e_j}^{(l)} \right) \tag{21}$$

This propagation step implements a bidirectional "edge-node-edge" information flow, allowing the representation of each original edge to both contain its own initial features and fuse with related local, higher-order structural information.

After obtaining edge-centric representations from the dual hypergraph branch, we aggregate the embeddings of edges incident to each original user node to form an edge-aware node representation. This representation is then fused with the node representation learned by the GCN branch and fed into a softmax classifier for fraud prediction.

3.7 Complexity Analysis and Scalability Discussion

To theoretically evaluate the efficiency of the proposed framework and clarify our architectural design choices, we analyze its time and space complexity. Let $N = |V|$ and $M = |E|$ denote the number of nodes and edges in the original graph, respectively. Let I = nnz (H) be the number of nonzero elements in the sparse incidence matrix, and d be the hidden dimension size.

Time Complexity: The computational cost is meticulously optimized via sparse matrix operations. The Dual Hypergraph Transformation (DHT) construction is linear with respect to the original edges, i.e., $O(M)$. For message passing, each Hypergraph Convolution layer performs two incidence-based propagations (node-to-hyperedge and hyperedge-to-node), resulting in a complexity of $O(I \cdot d)$ per layer. Concurrently, the standard GCN branch contributes $O(M \cdot d)$ per layer. For L layers, the dominant forward-pass cost is approximately $O(L(Md + Id))$. Furthermore, if the SMOTE-style graph augmentation is triggered, the pairwise-distance computation strictly operates on the synthesized minority subset rather than the full graph, taking $O(rm^2 D)$, where m is the minority class size and D is the feature dimension. Consequently, the time complexity scales linearly with the graph's structural density, ensuring computational efficiency.

Space Complexity and Hardware Bottleneck: Rather than materializing a dense $N \times M$ incidence matrix, our implementation utilizes sparse incidence indices. Therefore, the dominant memory footprint arises from storing node embeddings, edge embeddings, sparse graph structures, and intermediate activations: $O(Nd + Md_e + I)$, where d_e is the edge embedding dimension. While the sparse implementation mitigates topological memory overhead, the inherent paradigm of Dual Hypergraph learning elevates edges to primary learning subjects. In a massive telecommunications network comprising hundreds of thousands of nodes and millions of edges, caching large-scale edge-level continuous activations and gradients across L layers for backpropagation requires massive VRAM, inevitably leading to Out-of-Memory (OOM) errors on standard single-GPU hardware.

Justification for Subgraph Evaluation and Extreme Label Scarcity: To strike a balance between the higher-order semantic extraction capabilities of the dual hypergraph neural network and the aforementioned hardware memory constraints, training directly on the full graph at scale is computationally infeasible. This led us to adopt a strategy of extracting local communication subgraphs. Crucially, the extracted subgraphs preserve the extreme class imbalance characteristic of telecommunications networks. Consequently, the

extreme label scarcity phenomenon observed in our evaluationwhere only a dozen or so hidden fraudsters were identified among thousands of ordinary usersis not an artifact of the computational process, but rather a true reflection of the real-world data distribution. This setup perfectly mirrors actual anti-fraud operations: law enforcement must precisely locate highly covert, small-scale criminal groups (micro-clusters) within dense and vast local communities.

4. EXPERIMENTS

4.1 Dataset

We validate our proposed method on a real-world telecommunications fraud detection dataset. This dataset, collected and anonymized by a local telecommunications company, covers one month's worth of complete call detail records (CDRs) for 186,228 users in a city. All samples are classified into two categories: fraudsters and normal users. The dataset contains rich user attributes and behavioral information. Specific fields include basic user information such as: MD5-encrypted phone number, user ID, age, home city, home district/county, channel information, group information, ID document type, MD5-encrypted ID document number, network registration time, service plan information, total number of cards under the same ID document, user status, SIM card type, whitelist status, and disposition status. Simultaneously, the dataset also includes detailed call records, for example: event type, caller/callee numbers (encrypted), call duration, call type, caller/callee home location, roaming information, and the International Mobile Equipment Identity (IMEI) of the caller/callee devices.

For the data in each dimension of the dataset, we performed feature extraction using the data processing methods introduced in the experimental methods section. In terms of statistical features, we calculated a series of statistical functions, including sum, mean, maximum, minimum, median, quantiles, skewness, variance, and standard deviation. For pattern features, we focused on the users' calling behavior and extracted key metrics such as the number of communication partners and the total number of calls. Ultimately, a 30-dimensional feature vector was constructed for each user.

Furthermore, we abstract each user as a node in a graph and their communication behaviors (such as calls or SMS) as edges connecting the nodes, thereby constructing a large-scale, undirected communication network. This network comprises 186,228 nodes (representing individual users) and 4,098,898 edges (representing communication relationships). Among these users, 1,272 are labeled as fraudsters, with an imbalance ratio of approximately 0.0069. This indicates a significant class imbalance problem in the dataset. This data allows us to analyze the topological structure of the communication network, with graph sparsity being a key metric. In a network of 186,228 nodes, if a connection existed between every pair of nodes, it would form a Complete graph with a total of 1.734×10^{10} possible undirected edges. However, our constructed network has only 4,098,898 edges, meaning the actual number of connections is a minuscule fraction of the theoretical maximum. Its graph density is approximately

0.024%. This extremely low density indicates that the communication network is highly sparse. This sparsity is further evidenced by node connectivity. Among all users, the number of nodes with internal connections is 153,261. This means that approximately 17.7% of users did not establish any communication with other users in the network during the observation period, appearing as isolated nodes in the graph. The presence of these isolated nodes further enhances the sparse nature of the network.

4.2 Baselines

To comprehensively evaluate the performance of our proposed model, especially its learning capability in handling the inherent multi-dimensional sparsity of telecommunications data, we conducted experimental comparisons against three categories of representative models under a semi-supervised learning scenario.

Classic Graph Neural Networks: This category includes foundational and widely used general-purpose GNN models in the field of graph learning. They serve as our baseline for evaluating the fundamental performance of our model on the current task.

- **GCN**: A classic graph convolutional network that effectively learns local structural information by defining convolution operations in the graph's Fourier domain [39]. In node classification tasks, it updates a central node by aggregating the feature representations of its neighboring nodes.
- **GAT**: This model introduces the attention mechanism into the neighborhood aggregation process [40]. By calculating attention coefficients between neighboring nodes and the central node, GAT is able to assign different weights to different neighbors, thereby aggregating information more selectively.
- **GraphSAGE**: An inductive learning framework designed for large-scale graphs [41]. It generates node embeddings by sampling the neighbors of each node and then using a learnable aggregation function (e.g., mean and max-pooling), which provides excellent scalability.

GNNs with Imbalance Handling: Considering the inherent data imbalance problem in the telecommunications fraud detection task, we design this set of comparative experiments to validate the effectiveness of our model component.

Domain-Specific Models: To compare our model with advanced methods specifically designed for the telecommunications fraud detection task, we selected relevant models from the domain as benchmarks.

- **GAT-COBO**: A cost-sensitive graph attention network model for telecommunications fraud detection [16]. This model introduces cost-sensitive learning to adjust its focus on the minority class (fraudsters) and has achieved good results on some publicly available telecommunications fraud datasets.
- **CARE-GNN**: A well-established camouflage-resistant GNN for fraud detection [13]. It utilizes reinforcement learning to adaptively select informative intra-class neighbors and filter out deceptive inter-class edges to defend against structural camouflage.

- **BWGNN**: A state-of-the-art spectral-based graph neural network proposed for anomaly detection [43]. It designs a Beta Wavelet Graph Neural Network to capture high-frequency anomalous signals (i.e., the 'right-shift' phenomenon) in the graph spectral domain.
- **BAT**: A recent state-of-the-art topological augmentation framework for class-imbalanced graph learning [42]. Unlike traditional re-weighting or SMOTE-based resampling, it mitigates class-imbalance bias by rectifying topological asymmetries directly without class rebalancing.

4.3 Experimental Implementation

In our experiments, we used a real-world telecommunications fraud detection dataset to evaluate model performance. We randomly split the dataset into training, validation, and test sets in a 60%, 20%, and 20% ratio, respectively.

Hyperparameter settings: To ensure strict reproducibility, we explicitly specified the core thresholds applied within the framework. For initial graph structure enhancement, the behavioral similarity threshold θ (Equation 5) was empirically set to 0.65 to ensure high-confidence densification while preventing heterogeneous noise. During subgraph extraction, we adopted a highly reproducible pruning strategy, discarding the bottom 10% of edges in the relationship strength distribution, rather than using complex dynamic thresholds. In the graph augmentation module, the target imbalance ratio ρ is set to 0.80 to balance the minority class, and the decoder filtering threshold τ (Equation 16) is set to 0.75 to retain only highly credible synthetic edges. Training configuration: The entire training process runs for up to 3,000 epochs, with an early stopping mechanism applied using a patience value of 325 to mitigate overfitting. We employ the Adam optimizer with an initial learning rate of 0.01 and a weight decay of 5×10^{-4}. We apply a dropout rate of 0.5 for regularization. We use a joint optimization strategy combining classification loss and graph reconstruction loss, where the weight coefficient of the reconstruction loss (rec_weight) is set to 0.1. All experiments were conducted on a Linux server equipped with an Intel(R) Xeon(R) Gold 5218 CPU, 64 GB of memory, and an NVIDIA GeForce RTX 3090 GPU. The framework is implemented using PyTorch on Python 3.9.19.

4.4 Experimental Results and Analysis

We trained and tested our proposed graph neural network model and all baseline models on a real-world telecommunications fraud detection dataset. First, following the subgraph extraction method described in the experimental methods section above, we constructed a moderately-sized and representative communication subgraph for this experiment. This subgraph contains a total of 10,517 user nodes, of which 517 nodes are authentically labeled as suspected fraudsters, and the remaining 10,000 are normal users. For model training and evaluation, we randomly partitioned this subgraph dataset into training, validation, and test sets according to a 60%, 20%, and 20% ratio, respectively.

To evaluate the true performance of the models on this dataset, we adopted multiple metrics, including Accuracy, macro-F1, fraud-class recall, fraud-class F1-score, and fraud-class precision. As shown in Table 1, we can observe that, firstly, standard graph neural network models exhibit significant limitations when faced with highly imbal-anced data. Classic models like GCN, GAT, GraphSAGE all achieve misleadingly high accuracy rates of over 95%. However, their Recall, F1-score, and Precision metrics for the fraudster class are all 0. Meanwhile, although BAT is an advanced topological augmentation method, it fails to work under extreme label scarcity due to the lack of initial homophilic connections, leading to final results comparable to those of classical models such as GCN. This indicates that under conditions of extreme data imbalance, these models are completely biased towards the majority class and have degenerated into trivial classifiers that predict all nodes as normal users, thereby completely losing their ability to identify the critical minority class, although some studies report validation of these methods on balanced datasets.

Table 1. Performance Comparison of Different Models.

Method	Accuracy	Macro F1	Fraud Recall	Fraud F1	Fraud Precision
GCN	95.10%	48.75%	0.0000	0.0000	0.0000
GAT	95.10%	48.75%	0.0000	0.0000	0.0000
GraphSAGE	95.10%	48.75%	0.0000	0.0000	0.0000
GAT-COBO	81.30%	53.97%	42.27%	18.51%	11.85%
BAT	95.10%	48.75%	0.0000	0.0000	0.0000
BWGNN	89.62%	51.95%	10.97%	9.41%	8.23%
CARE-GNN	46.65%	37.18%	72.95%	12.32%	6.73%
Ours	64.64%	46.90%	69.90%	16.22%	9.17%

Furthermore, state-of-the-art domain-specific baselines still exhibit obvious drawbacks. Taking GAT-COBO as an example, this cost-sensitive graph attention network tailored for telecommunications fraud detection introduces heavy penalty terms for minority-class misclassification to boost fraud detection performance. Nevertheless, its recall only reaches 42.27% on the real-world dataset, while our proposed model attains a recall of 69.90%, ranking second across all compared approaches. More importantly, our method yields much higher precision and F1 scores than CARE-GNN despite competitive recall performance.

CARE-GNN achieves the maximum recall for fraud samples, yet this advantage comes at the expense of drastically degraded overall accuracy and precision. By contrast, our model strikes the optimal trade-off among all high-recall algorithms: it elevates the precision and F1 metric for fraud identification while retaining competitive recall, making it more applicable to preliminary anti-fraud triage scenarios. Overall, comprehensive evaluation metrics consistently demonstrate the remarkable performance superiority of our proposed method.

4.5 Model Diagnostic Analysis

To provide a deeper diagnostic insight into the model's classification behavior on the test set, Figure 4 presents the confusion matrix of our proposed model. The test set comprises 2,104 nodes in total, consisting of 2,001 normal users and only 103 verified fraudsters. As clearly illustrated in the matrix, our model successfully identified 72 out of the 103 fraud nodes (true positives), directly accounting for the 69.90% Fraud Recall. Simultaneously, the model misclassified 713 normal users as fraudsters (false positives). While this results in a precision of 9.17%, a comparative analysis with the data in Table 1 reveals the significant superiority of our model in controlling misclassification costs. This demonstrates that our model does not simply rely on aggressive flagging to improve recall; instead, it breaks the zero-recall dilemma while preserving the normal users from false accusations far more effectively than oversampling strategies.

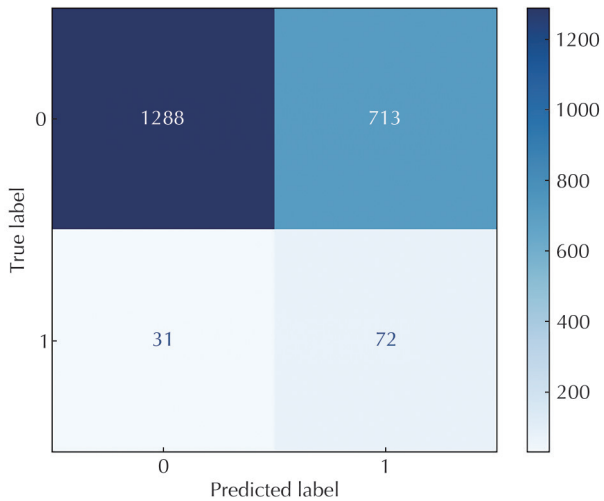


Figure 4. Confusion matrix on the localized subgraph. The model successfully identifies 69.90% of the highly camouflaged fraudsters while maintaining a controlled false positive rate within the extreme label scarcity setting.

4.6 Ablation Study

To further verify the effectiveness of the proposed modules, we conducted ablation experiments by progressively removing key components from the complete model. Specifically, /KNN removes behavioral-similarity-based KNN edges while retaining real communication edges; /CALL removes real communication edges and relies only on behavioral similarity for graph construction; /SMOTE removes the improved minority-class feature interpolation and structural enhancement module; and /HyperDrop removes the dual hypergraph learning mechanism and reduces the model to a conventional GNN.

As shown in Figure. 5, the complete model achieves the best or jointly best performance in terms of AUC, Fraud Recall, and Precision, demonstrating the effectiveness of the proposed collaborative framework. Removing KNN similarity edges leads to the largest overall degradation, with AUC and Recall decreasing from 0.72 and 0.69 to 0.62 and 0.62, respectively. This result confirms that behavioral-similarity-based structural completion is essential for recovering latent relationships and alleviating graph fragmentation in sparse communication networks.

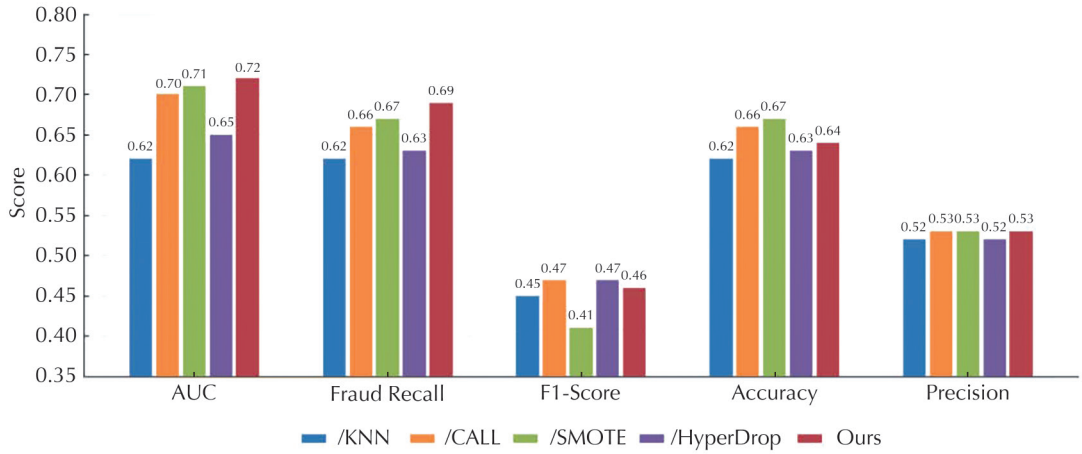


Figure 5. Ablation Experiment Results Comparison.

Removing real communication edges also decreases AUC and Recall to 0.70 and 0.66, respectively, indicating that explicit communication topology remains important for preserving direct interaction patterns and local structural constraints. However, its smaller performance drop compared with /KNN suggests that real communication relationships alone are insufficient to reveal concealed fraud associations. The fusion of explicit communication links and implicit behavioral-similarity links therefore provides a more comprehensive representation of user risk relationships.

After removing the SMOTE module, the F1-score decreases markedly to 0.41, indicating that minority-class augmentation is important for mitigating class bias and improving the stability of fraud-node discrimination. Finally, removing the dual hypergraph learning mechanism decreases AUC and Recall to 0.65 and 0.63, respectively. This verifies that high-order edge semantics are necessary for capturing collaborative fraud patterns that cannot be fully modeled by conventional pairwise message passing.

Overall, the results demonstrate that KNN-based structural completion, real communication topology, minority-class augmentation, and dual hypergraph learning are complementary components. Their joint use enables the proposed model to address structural sparsity, extreme class imbalance, and insufficient high-order semantic information in telecom fraud detection.

4.7 Parameter Sensitivity Analysis

To evaluate the robustness of our framework and explicitly address the impact of key hyperparameters related to our multi-sparsity mitigation strategies, we conduct a comprehensive parameter sensitivity analysis. We focus on three crucial thresholds governing the structural, sample, and semantic dimensions: the behavioral similarity threshold (θ), the target imbalance ratio (ρ), and the decoder filtering threshold (τ). The experimental results are summarized in Figure 6.

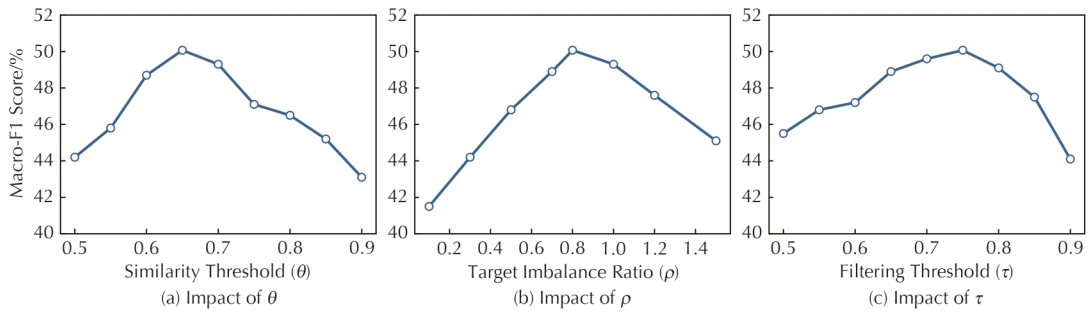


Figure 6. Parameter sensitivity analysis evaluating the impact of (a) the behavioral similarity threshold θ, (b) the target imbalance ratio ρ, and (c) the decoder filtering threshold τ on the overall model performance (Macro-F1 score).

1) Impact of Behavioral Similarity Threshold (θ): The threshold θ (as defined in Eq. 5) controls the strictness of the initial graph densification, directly targeting structural sparsity. As shown in Figure 6(a), setting θ too low (< 0.55) indiscriminately establishes connections between behaviorally disparate nodes, injecting severe heterophilic noise that triggers over-smoothing and degrades the Macro-F1 score to below 46%. Conversely, an overly strict threshold (> 0.80) severely prunes the network, leaving critical fraud nodes isolated and unable to aggregate neighborhood information. The empirical optimal balance is achieved around $\theta = 0.65$, which provides a sufficiently dense yet highly reliable homophilic topology for subsequent message passing.
2) Impact of Target Imbalance Ratio (ρ): Rather than a simple oversampling multiplier, we evaluate the impact of sample augmentation through the target imbalance ratio ρ, defined as the ratio of the synthesized minority class size to the majority class size within the training set. Given the extreme initial disparity (517 fraud nodes vs. 10000 normal users in the training split), a small ratio ($\rho < 0.50$) is vastly insufficient to rescue the minority features from being overwhelmed during neighborhood aggregation. As shown in Figure 6(b), the model performance peaks when ρ is between 0.75 and 0.85. Interestingly, forcing a strict 1:1 balance ($\rho = 1.0$) slightly degrades performance. This occurs because synthesizing over a thousand virtual nodes from a mere 4 seed nodes inevitably introduces overlapping features and topological redundancy, leading to overfitting. Thus, an empirical optimum is reached around $\rho \approx 0.80$, where the minority class is sufficiently augmented to resist over-smoothing but not over-generated.

3) Impact of Decoder Filtering Threshold (τ): The threshold τ (as defined in Eq. 20) in the bilinear decoder strictly dictates the structural plausibility of inherited edges for synthetic nodes, addressing potential semantic noise. As depicted in Figure 6(c), a relaxed threshold ($\tau < 0.50$) admits structurally implausible connections, injecting topological noise into the final augmented graph. In contrast, setting $\tau > 0.85$ becomes too rigid, discarding valid homophilic connections and leaving synthetic nodes semantically isolated. The fact that peak performance is achieved around $\tau = 0.75$ validates the effectiveness of the learnable decoder, which mitigates structural and semantic sparsity while preserving high-confidence connections.

4.8 Evaluation on Another Highly Imbalanced Network

To address potential concerns regarding the limited sample size in our preliminary evaluation (as described in Section 4.4) and to more rigorously validate the statistical validity and practical applicability of the framework, we conducted an expanded evaluation on a significantly larger telecommunications subgraph.

1) Evaluation on another Imbalanced Network: In this extended experiment, we extracted a more comprehensive network subset comprising a total of 6,920 user nodes. This dataset preserves the severe class imbalance characteristic of the global carrier network, containing 6,874 normal users (99.34%) and 46 authenticated fraudsters (0.66%). The dataset was partitioned into training, validation, and test sets following the same 60:20:20 ratio. Crucially, the independent test set now encompasses 1,375 nodes, including 9 verified fraudsters, providing a foundation for evaluating minority-class detection performance.
2) Practical Utility in Triage Operations: Figure 7 presents the confusion matrix of our proposed framework on this expanded test set. As clearly illustrated, the model successfully identified 5 out of the 9 elusive fraud nodes, directly achieving a robust Fraud Recall of 55.56%. Simultaneously, the model flagged 203 normal users as suspected fraudsters (False Positives), resulting in a Precision of 2.40%.

While the absolute precision appears numerically low, this is an inherent mathematical consequence of the overwhelming 99.34% negative class majority; this performance translates to profound practical significance in real-world, cost-sensitive triage systems. In conventional telecommunications anti-fraud operations, identifying concealed syndicates among vast populations is akin to finding a needle in a haystack; standard baseline GNNs typically collapse to a 0% recall, offering zero assistance to investigators. In stark contrast, our framework successfully narrows down the entire 1,375-user search space to a highly concentrated suspect pool of merely 208 individuals (5 True Positives + 203 False Positives).

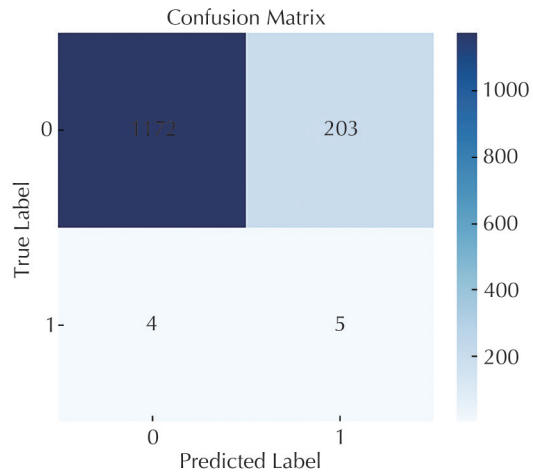


Figure 7. Confusion matrix of the proposed model on the extended imbalanced network. Evaluated on a substantially larger test set (1,375 nodes), the model retrieves 55.56% of the verified fraudsters (5 out of 9). This demonstrates its practical utility as an effective preliminary filter by reducing the investigative search space by approximately 85%.

By achieving an 85% reduction in manual screening workload while successfully trapping more than half of the highly camouflaged targets, the model operates as a highly effective preliminary filter. Given that the cost of a False Negative (missing a fraudster) is substantially higher than investigating a False Positive, this massive reduction in investigative search space validates the framework's practical value in large-scale security deployments.

5. CONCLUSION

This paper proposes a novel telecommunications fraud detection framework based on a Dual Hypergraph Neural Network, aimed at systematically addressing the challenges of multi-dimensional sparsity prevalent in real-world communication networks. By decomposing sparsity into three dimensions—structure, sample, and semantics—our framework provides a unified solution that integrates relationship-aware graph structure enhancement, imbalanced learning, and higher-order semantic modeling. Specifically, we enhance topological connectivity by reconstructing latent co-occurrence links and extracting high-quality subgraphs. To mitigate extreme class imbalance, we design an improved graph augmentation mechanism that not only synthesizes minority-class samples in the feature space but also reconstructs their structural connections via a learned decoder. Furthermore, the Dual Hypergraph Transformation enables explicit edge-level representation learning, allowing the model to capture higherorder communication behaviors. Experiments were conducted on a real-world telecommunications fraud dataset with extreme class imbalance (approximately 0.69%). Given the computational complexity of higher-order modeling, the evaluation focused on local subgraphs representing extremely label-scarce scenarios. On an independent test set, our model successfully broke through the zero recall dilemma

commonly faced by baseline and state-of-the-art models. Although absolute accuracy and F1 scores are inherently limited by the extreme class imbalance and data fragmentation in real-world datasets, the model's ability to achieve a 69.90% fraud recall rate—that is, identifying highly concealed fraudsters missed by traditional methods—fully validates its practical value. Overall, the proposed framework serves as an effective preliminary screening tool for real-world telecommunications anti-fraud systems, helping to narrow the scope of investigations. Future work will focus on optimizing the space complexity of hypergraph convolutions to extend this framework to large-scale full-graph scenarios.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the anonymous reviewers for their constructive comments. This work is supported in part by funds from the National Science Foundation of China (No. 62367005), 2024 Central Government Guides Local Science and Technology Development Fund Project (No. 24ZYQA051), Major Cultivation Project of Scientific Research and Innovation Platforms in Gansu Provincial Colleges and Universities (No. 2024CXPT-17). All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] X. Ma *et al.*, "A comprehensive survey on graph anomaly detection with deep learning," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12012–12038, 2023.

[2] N. F. Ryman-Tubb, P. Krause, and J. Garn, "How artificial intelligence and machine learning research impacts payment card fraud detection: A survey and industry benchmark," *Eng. Appl. Artif. Intell.*, vol. 76, pp. 50–57, 2018.

[3] P. Burge and J. Shawe-Taylor, "Detecting cellular fraud using adaptive prototypes," in *Proc. AAAI Workshop AI Approaches Fraud Detect. Risk Manag.*, 1997, pp. 9–13.

[4] C. S. Hilas and P. A. Mastorocostas, "An application of supervised and unsupervised learning approaches to telecommunications fraud detection," *Knowledge-Based Systems*, vol. 21, no. 7, pp. 721-726, 2008.

[5] R. K. Gopal and S. K. Meher, "A rule-based approach for anomaly detection in subscriber usage pattern," *International Journal of Computer and Information Engineering*, vol. 1, no. 10, pp. 3002-3005, 2007.

[6] P. A. Estvez, C. M. Held, and C. A. Perez, "Subscription fraud prevention in telecommunications using fuzzy rules and neural networks," *Expert Syst. Appl.*, vol. 31, no. 2, pp. 337–344, 2006.

[7] S. Rajani and M. Padmavathamma, "A model for rule based fraud detection in telecommunications," *International Journal of Engineering Research & Technology*, vol. 1, no. 5, pp. 1-4, 2012..

[8] C. S. Hilas, "Designing an expert system for fraud detection in private telecommunications networks," *Expert Syst. Appl.*, vol. 36, no. 9, pp. 11559–11569, 2009.

[9] R. Andreswari, D. A. Rahmani, and R. Rahmawati, "Comparative analysis of K-means and K-nearest neighbor algorithm for telecom fraud detection," in *Proc. 2nd Int. Conf. Inf. Technol. Educ. (ICIT&E)*, Malang, Indonesia, 2022, pp. 107–111.

[10] M. Kashir and S. Bashir, "Machine learning techniques for SIM box fraud detection," in *Proc. Int. Conf. Commun. Technol. (ComTech)*, 2019, pp. 4–8.

[11] M. Taniguchi, M. Haft, J. Hollmen, and V. Tresp, "Fraud detection in communication networks using neural and probabilistic methods," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '98)*, vol. 2, 1998, pp. 1241-1244.

[12] C. Phua, V. Lee, K. Smith, and R. Gayler, "A comprehensive survey of data mining-based fraud detection research," 2010, arXiv:1009.6119.

[13] 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 *Proc. 29th ACM Int. Conf. Inf. Knowl. Manag. (CIKM)*, 2020, pp. 315–324.

[14] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: A survey," *Data Mining Knowl. Discov.*, vol. 29, no. 3, pp. 626–688, 2015.

[15] D. Wang *et al.*, "A semi-supervised graph attentive network for financial fraud detection," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Beijing, China, 2019, pp. 598–607.

[16] X. Hu *et al.*, "GAT-COBO: Cost-sensitive graph neural network for telecom fraud detection," *IEEE Trans. Big Data*, vol. 10, no. 4, pp. 528–542, 2024.

[17] G. Chu, J. Wang, Q. Qi, H. Sun, S. Tao, H. Yang, J. Liao, and Z. Han, "Exploiting spatial-temporal behavior patterns for fraud detection in telecom networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 6, pp. 4564-4577, 2023.

[18] X. Hu, H. Chen, S. Liu, H. Jiang, G. Chu, and R. Li, "BTG: A bridge to graph machine learning in telecommunications fraud detection," *Future Gener. Comput. Syst.*, vol. 137, pp. 274–287, 2022.

[19] T. N. Kipf and M. Welling, "Variational graph auto-encoders," in *Proc. NIPS Workshop Bayesian Deep Learn.*, 2016.

[20] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Deep graph structure learning for robust representations: A survey," 2021, arXiv:2103.03036.

[21] T. Zhao, X. Zhang, and S. Wang, "GraphSMOTE: Imbalanced node classification on graphs with graph neural networks," in *Proc. 14th ACM Int. Conf. Web Search Data Mining (WSDM)*, Jerusalem, Israel, 2021, pp. 833–841.

[22] L. Qu, H. Zhu, R. Zheng, Y. Shi, and H. Yin, "ImGAGN: Imbalanced network embedding via generative adversarial graph networks," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21)*, Virtual Event, Singapore, 2021, pp. 1390-1398.

[23] Y. Liu, X. Ao, Z. Qin, J. Chi, J. Feng, H. Yang, and Q. He, "Pick and choose: A GNN-based imbalanced learning approach for fraud detection," in *Proceedings of the Web Conference 2021 (WWW '21)*, Ljubljana, Slovenia, 2021, pp. 3168-3177.

[24] L. Gong and Q. Cheng, "Exploiting edge features for graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, 2019, pp. 9211–9219.

[25] J. Jo, J. Baek, S. Lee, D. Kim, M. Kang, and S. J. Hwang, "Edge representation learning with hypergraphs," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, 2021, pp. 7534–7546.

[26] G. Salha, R. Hennequin, V. Tran, and M. Vazirgiannis, "A degeneracy framework for graph generation with normalizing flows," 2019, arXiv:1907.02586.

[27] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially regularized graph autoencoder for graph embedding," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2018, pp. 2609–2615.

[28] J. You, R. Ying, X. Ren, W. Hamilton, and J. Leskovec, "GraphRNN: Generating realistic graphs with deep auto-regressive models," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 5708–5717.

[29] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proc. Web Conf. (WWW)*, 2020, pp. 2704–2710.

[30] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 3558–3565.
[31] M. A. Stricker and M. Orengo, "Similarity of color images," in *Proc. SPIE Storage Retrieval Image Video Databases III*, 1995, pp. 381–392.
[32] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high-dimensional space," in *Proc. 8th Int. Conf. Database Theory*, 2001, pp. 420–434.
[33] A. A. Nanavati *et al.*, "On the structural properties of massive telecom call graphs: Findings and implications," in *Proc. 15th ACM Int. Conf. Inf. Knowl. Manag. (CIKM)*, 2006, pp. 435–444.
[34] C. Cortes, D. Pregibon, and C. Volinsky, "Communities of interest," in *Proc. Int. Symp. Intell. Data Anal. (IDA)*, Berlin, Heidelberg, 2001, pp. 105–114.
[35] J.-P. Onnela *et al.*, "Structure and tie strengths in mobile communication networks," *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 18, pp. 7332–7336, 2007.
[36] R. Xiang, J. Neville, and M. Rogati, "Modeling relationship strength in online social networks," in *Proc. 19th Int. Conf. World Wide Web (WWW)*, 2010, pp. 981–990.
[37] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manag.*, vol. 24, no. 5, pp. 513–523, 1988.
[38] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.
[39] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations (ICLR)*, Toulon, France, 2017.
[40] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations (ICLR)*, Vancouver, BC, Canada, 2018.
[41] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, Long Beach, CA, USA, 2017, pp. 1024-1034.
[42] Z. Liu, R. Qiu, Z. Zeng, H. Yoo, D. Zhou, Z. Xu, Y. Zhu, K. Weldemariam, J. He, and H. Tong, "Class-imbalanced graph learning without class rebalancing," in *Proceedings of the 41st International Conference on Machine Learning (ICML)*, Vienna, Austria, 2024, vol. 235, pp. 31747-31772.
[43] J. Tang, J. Li, Z. Gao, and J. Li, "Rethinking graph neural networks for anomaly detection," in *Proc. 39th Int. Conf. Mach. Learn. (ICML)*, 2022, pp. 21076–21089.
[44] Y. Duan *et al.*, "CaT-GNN: Enhancing credit card fraud detection via causal temporal graph neural networks," 2024, arXiv:2402.14708.