ARL-TR-10219 ● Sep 2025

DEVCOM
ARMY RESEARCH
LABORATORY

Text-Enriched Hierarchical Graph Anomaly Detection with Uncertainty Detection

by Xiayan Ji, John Richardson, Adrienne Raglin, Insup Lee, and Oleg Sokolsky

DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer’s or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-10219 ● SEP 2025

Text-Enriched Hierarchical Graph Anomaly Detection with Uncertainty Detection

John Richardson and Adrienne Raglin
DEVCOM Army Research Laboratory

Xiayan Ji, Insup Lee, and Oleg Sokolsky
University of Pennsylvania

REPORT DOCUMENTATION PAGE

1. REPORT DATE September 2025	**2. REPORT TYPE** Technical Report	**3. DATES COVERED** **START DATE** 10/01/2024	 **END DATE** 05/31/2025

4. TITLE AND SUBTITLE
Text-Enriched Hierarchical Graph Anomaly Detection with Uncertainty Detection

5a. CONTRACT NUMBER	**5b. GRANT NUMBER**	**5c. PROGRAM ELEMENT NUMBER**
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
Xiayan Ji, John Richardson, Adrienne Raglin, Insup Lee, and Oleg Sokolsky

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) DEVCOM Army Research Laboratory ATTN: FCDD-RLA-IC Aberdeen Proving Ground, MD 21005	**8. PERFORMING ORGANIZATION REPORT NUMBER** ARL-TR-10219

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**	**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
ORCIDs: Xiayan Ji, 0000-0002-4896-7769; John Richardson, 0009-0003-3914-4765; Adrienne Raglin, 0000-0003-2147-8938; Insup Lee, 0000-0003-2672-1132; and Oleg Sokolsky, 0000-0001-5282-0658

14. ABSTRACT

Detecting anomalous nodes in hierarchical social networks is crucial for preventing corporate fraud and conducting forensic analysis of criminal organizations. Traditional approaches rely on network analysis but overlook rich textual interactions from emails, web content, and social media. While graph neural networks show promise, they often fail to capture hierarchical structures crucial for anomaly detection. Moreover, existing methods lack robust uncertainty quantification, essential for high-stakes decision-making. We propose a novel approach integrating lightweight language models to extract textual edge embeddings, transforming them into node embeddings via convolution operations. A hyperbolic graph convolutional network models the latent hierarchy of social networks, leveraging hyperbolic space for improved hierarchical representation. Additionally, we quantify uncertainty by calibrating thresholds on node anomaly scores to ensure reliable detection. Evaluated on the real-world Enron fraud dataset and an in-house synthetic criminal network dataset, our method achieves performance comparable to large language models while significantly reducing computational overhead.

15. SUBJECT TERMS
Military Information Sciences, graph neural network, anomaly detection, uncertainty quantification, social networks, decision-making

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**
a. REPORT UNCLASSIFIED	**b. ABSTRACT** UNCLASSIFIED	**C. THIS PAGE** UNCLASSIFIED	UU	31

19a. NAME OF RESPONSIBLE PERSON John Richardson	**19b. PHONE NUMBER** *(Include area code)* (520) 691-5682

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

1. Introduction

The detection of anomalous nodes within hierarchical social networks holds significant real-world implications. For instance, it plays a vital role in uncovering corporate fraud and advancing forensic analyses of criminal organizations. In corporate settings, fraudulent activities are often perpetrated by a few key individuals, leading to severe consequences like bankruptcy. Similarly, identifying suspicious nodes in criminal networks can help prevent illicit activities.

Existing approaches rely on traditional network analysis using centrality measures (Das and Sinha 2018) but overlook rich textual data from emails, web content, and social media, which could enhance anomalous node detection by revealing the interactions between nodes. Some works explore the use of sentiment analysis, entity recognition, and topic modeling (Celebi and Shashidhar 2022; Wen et al. 2023; Cheng and Cai 2024) in analyzing interaction texts, but they often fail to incorporate the structural dependencies of nodes within the network. While graph neural networks (GNNs) have shown promise in capturing the structural dependency (Kim et al. 2022), the hierarchy among nodes is often overlooked in existing works, which are crucial for identifying anomalies in a systematic manner.

Several challenges remain in effectively detecting anomalous nodes within hierarchical social networks. The vast volume of structured and unstructured web data necessitates efficient, scalable processing methods. Powerful techniques like large language models (LLMs) have considerable latency that may hinder timely analysis. Additionally, privacy and security concerns limit the adoption of LLMs and cloud-based AI, as sensitive data cannot always be processed externally due to regulatory and confidentiality constraints. Lastly, the lack of uncertainty quantification in existing models hinders their reliability in decision-making, as false positives can lead to unnecessary investigations while false negatives allow fraudulent activities to persist. Addressing these challenges requires an approach that is computationally efficient, privacy-preserving, and capable of providing robust confidence estimates.

To address these challenges, we propose Text-enriched Anomaly detection on hierarchical Graphs (TAG)—a text-enriched hierarchical graph anomaly detection framework, as shown in Figure 1. TAG integrates lightweight language models to extract edge embeddings from textual interactions, which are then incorporated into node embeddings via edge convolution. For anomaly detection, we use a hyperbolic graph convolutional network (HGCN)-based autoencoder (Chami et al. 2019) that effectively captures the latent hierarchy of social networks. Unlike Euclidean-based graph convolutional networks (GCNs), The HGCN is based on hyperbolic space, leading to a more expressive representation for hierarchical relationships. Each node is assigned an anomaly score based on the HGCN autoencoder, encoding both textual and structural information. We calibrate thresholds on these scores to ensure reliable detection with quantified uncertainty, providing theoretical guarantees on error rates.

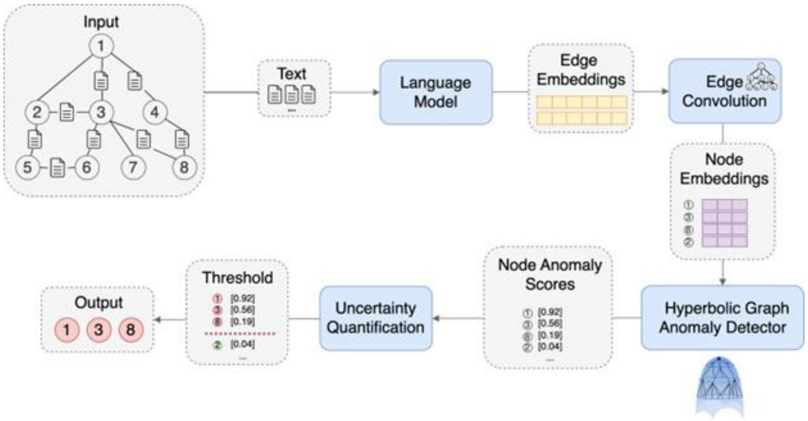


Figure 1. Overview of the TAG pipeline.

We evaluate the effectiveness of TAG on the Enron fraud dataset and a criminal network dataset, demonstrating comparable performance with LLM but with significantly lower computational cost, faster inference, and enhanced privacy through local deployment. The main contributions of this work are summarized as follows:

- Improved detection performance: TAG achieves 9.89% higher performance compared with LLM-based approaches by effectively capturing both structural and textual information.
- Optimized inference-time efficiency: TAG achieves 8.77 times faster inference than LLM-based models by leveraging a more efficient HGCN-based architecture.
- Lightweight and privacy-preserving deployment: with significantly fewer parameters than LLMs, TAG requires less computation, supports local deployment, and thereby enhances privacy.
- Reliable detection with quantifiable uncertainty: unlike existing graph anomaly detection methods that lack performance guarantees, we provide statistical guarantees on false positive (FPR) and negative rates (FNR) to ensure robust and trustworthy detection.
- Benchmark dataset for network analysis: we clean, structure, and open source the Enron fraud dataset and a criminal network dataset as graphs, consolidating ground truth labels for anomalies. These curated datasets

serve as valuable benchmarks for studying real-world anomaly detection in social networks with textual web data.

This report continues with the introduction of the problem statement in the following section. Section 3 describes TAG's details. Section 4 empirically demonstrates TAG's effectiveness on the Enron fraud and criminal network datasets. We list related works in Section 5 and conclude in Section 6.

2. Problem Statement

Let $G = (V, E)$ be a hierarchical social network with $|V| = N$ nodes, $|E| = M$ edges, and adjacency matrix $\mathrm{A} \in \{0, 1\}^{N \times N}$. Each node $i \in V$ is described by a feature vector $\mathrm{x}_i \in \mathbb{R}^d$. In addition, each edge $e = (i, j) \in E$ is endowed with textual content $\tau\,(e)$ that records communication between nodes i and j.

We define a binary label y_i for each node i, indicating whether it is anomalous:

$$y_i = \begin{cases} 1, & \text{if node } i \text{ is anomalous,} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The goal is to learn a function that predicts whether each node is anomalous, based on both the graph structure and the textual edge information.

$$f: (G, \{x_i\}, \{\tau(e)\}_{e\in \mathrm{E}}) \longrightarrow \{0,1\}^N \tag{2}$$

3. Textual Edge-Enriched HGCN Anomaly Detector

Here, we describe our text-enriched HGCN-based autoencoder for detecting anomalous nodes on a hierarchical social network, as discussed in previous problem statement (Section 2). First, we extract edge embeddings on the textual edges using a pretrained language model and integrate them to enrich node embeddings (Section 3.1). Next, we construct an autoencoder based on a HGCN to learn the latent hierarchy in the social network and perform anomaly detection on nodes (Section 3.2). Finally, we provide guarantees on the prediction results by calibrating thresholds on the node anomaly scores (Section 3.3).

3.1 Textual Edge-Enriched Node Embeddings

There are two core steps involved to derive the text-enriched node embeddings; namely, textual edge embedding extraction and integration to node embedding with edge convolution.

3.1.1 Textual Edge Embedding Extraction

To begin, we define a pretrained language model $\Phi : T \longrightarrow \mathbb{R}^{\mathrm{d}_e}$ that maps textual inputs in T to embeddings in $\mathbb{R}^{\mathrm{d}_e}$, where d_e is a fixed dimension depending on Φ. For each edge $e \in E$, the corresponding text is embedded as $\mathrm{u}_e = \Phi\ (\tau\ (e)) \in \mathbb{R}^{\mathrm{d}_e}$. This yields a semantic vector u_e for each edge e, capturing the content of its text.

3.1.2 Node Embeddings Using Edge Convolution

Let each node $i \in V$ have an initial feature vector $\mathrm{x}_i \in \mathbb{R}^{\mathrm{d}_n}$ that consists of d_n classical centrality features for network analysis; e.g., degree centrality (Park 2024), PageRank (Ekle and Eberle 2024), and betweenness centrality (Deprez et al. 2024). We define a set $E' = \{e = (i,j) \mid (A(i,j) = 1\}$ that consists of all the edges connecting to node i. We incorporate the edge embeddings $\{\mathrm{u}_e\}_{e \in E'}$through an edge convolution mechanism that is based on message-passing algorithms. We define three learnable functions for updating node representations by aggregating information from their neighbors:

$$\phi_{edge} : \mathbb{R}^{\mathrm{d}_e} \to \mathbb{R}^{\mathrm{d}_h}, \phi_{node} : \mathbb{R}^{(d_n + d_h)} \to \mathbb{R}^{\mathrm{d}_h}, \phi_{final} : \mathbb{R}^{\mathrm{d}_h} \to \mathbb{R}^{\mathrm{d}_o} \quad (3)$$

These functions transform edge embeddings to a hidden space of dimension d_h, update node embeddings with the edge information, and project the aggregated result to the final dimension d_o. We describe the details of the three functions as follows.

First, we perform edge attribute transformation with function ϕ_{edge}. For an edge e with embedding $\mathrm{u}_e \in \mathbb{R}^{\mathrm{d}_e}$, we apply $\mathrm{h}_e = {}_{\mathrm{edge}}(\mathrm{u}_e) \in \mathbb{R}^{\mathrm{d}_h}$. This step is to reduce the dimension of text embeddings from d_e to d_h prior to convolution, enhancing computational efficiency, mitigating overfitting, and improving the capture of relevant local patterns. Note that d_h is a tunable hyperparameter that balances model expressiveness and computational cost.

Second, we apply function ϕ_{node} to integrate the edge information to node embeddings. For a message in edge $e = (i, j\ )$, we concatenate the source node feature $\mathrm{x}_j \in \mathbb{R}^{\mathrm{d}_n}$ with h_e, forming a vector $[\mathrm{x}_j\ , \mathrm{h}_e] \in \mathbb{R}^{(d_n + d_h)}$ and compute $\mathrm{m}_e = \phi_{\mathrm{node}}([\mathrm{x}_j, \mathrm{h}_e]) \in \mathbb{R}^{\mathrm{d}_h}$. This vector m_e serves as the message of how the node i embeddings should be updated with the textual interactions from nodes j to i.

Third, we use the function ϕ_{final} to aggregate messages from all neighboring nodes. For each node i, let $\mathcal{N}(i)$ be the set of its in-neighbors. The messages $\{m_e: j \in \mathcal{N}(i)\}$ are combined via an aggregator AGG: $\mathcal{P}(\mathbb{R}^{d_h}\ ) \longrightarrow \mathbb{R}^{d_h}$, where

$\mathcal{P}\ (\mathbb{R}^{d_h})$ is the set of finite subsets of $\mathbb{R}^{d_h}$. In our work, we set AGG to be the elementwise mean, but we comment that other choices like sum and max are also valid. With the mean aggregation from all neighboring nodes, we have

$$\tilde{x}_i = AGG(\{m_e | j \in \mathcal{N}(i)\}) \in \mathbb{R}^{d_h}$$

Finally, we update the target node's embedding using the aggregated message. The aggregated vector $\tilde{x}_i$ is finally mapped to the output dimension d_o: $\acute{x}_i = \phi_{final}(\tilde{x}_i) \in \mathbb{R}^{d_o}$. To sum up, the EdgeConvolution mechanism for node i is

$$\acute{x}_i = \phi_{final}\left(AGG_{\ j \in \mathcal{N}(i)}\{\phi_{node}\left(\left[x_j, \phi_{edge}(u_e)\ \right]\right)\}\right) \tag{4}$$

We illustrate the feature extraction process explained above in Figure 2.

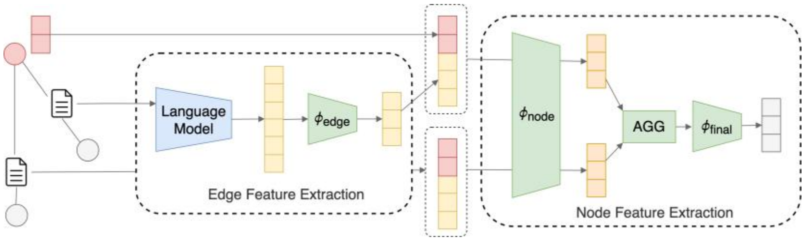


Figure 2. Edge and node feature extraction.

In fact, multiple EdgeConv layers may be stacked together to enhance the node representation. For example, in this work, we apply two successive layers with rectified linear unit (ReLU) layers after them, which yields

$$X^{(1)} = EdgeConv^{(1)}\left(X^{(0)}, \{u_e\}_{e\in E}\ \right),\ X^{(1)}_{act} = ReLU\left(X^{(1)}\right), \tag{5}$$

$$X^{(2)} = EdgeConv^{(2)}\left(X^{(1)}_{act}, \{u_e\}_{e\in E}\ \right),\ X^{(2)}_{act} = ReLU\left(X^{(2)}\right), \tag{6}$$

where $X^{(\ell)}, X^{(\ell)}_{act} \in \mathbb{R}^{N\times d_o}$ stores the node feature matrix and the updated node embeddings after the ℓ-th layer correspondingly. The activation function ReLU is applied between consecutive EdgeConv layers, ensuring nonlinearity and aiding the model in learning more expressive representations. This pipeline incorporates edge embeddings derived from textual information and node centrality features, capturing rich contextual relationships. A visualization of our EdgeConv layers can be found in Figure 3.

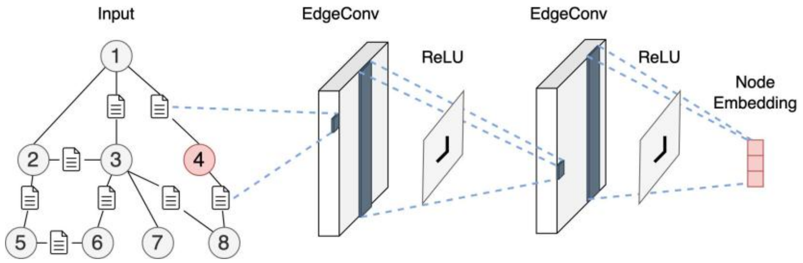


Figure 3. EdgeConv-based processing. Two stacked EdgeConv layers with ReLU activation to integrate textual edges and generate the final node embeddings.

3.2 Hyperbolic Graph Anomaly Detector

Hierarchical or tree-like patterns frequently arise in social networks, making it challenging for Euclidean embeddings to capture such structures effectively. To address this issue, we propose using a HGCN (Chami et al. 2019) within an autoencoder framework for anomaly detection. In hyperbolic geometry, negative curvature naturally accommodates hierarchical relationships, often resulting in more compact and faithful embeddings than those obtained in Euclidean space.

Our HGCN-based autoencoder operates by encoding and decoding node embeddings in a hyperbolic space (the Poincaré ball), while simultaneously imposing a distance-based hierarchy among node prototypes. By training only on on-anomalous data, the autoencoder yields higher reconstruction errors on anomalous inputs, thereby facilitating effective detection. We detail the model architecture in Section 3.2.1 and the training procedure in Section 3.2.2.

3.2.1 Model Architecture

This section explains the key components of our HGCN-based autoencoder, including the curvature parameter, class prototypes, encoder, hyperbolic mapping, and decoder. An overview of the model architecture is shown in Figure 4. The curvature parameter controls the negative curvature of the hyperbolic space, class prototypes serve as centroids in this space to represent hierarchical classes, the encoder transforms node embeddings from Section 3.1 into latent representations, the hyperbolic mapping projects those representations onto the Poincaré ball, and the decoder reconstructs the original features for anomaly detection. We provide the details of each component as follows.

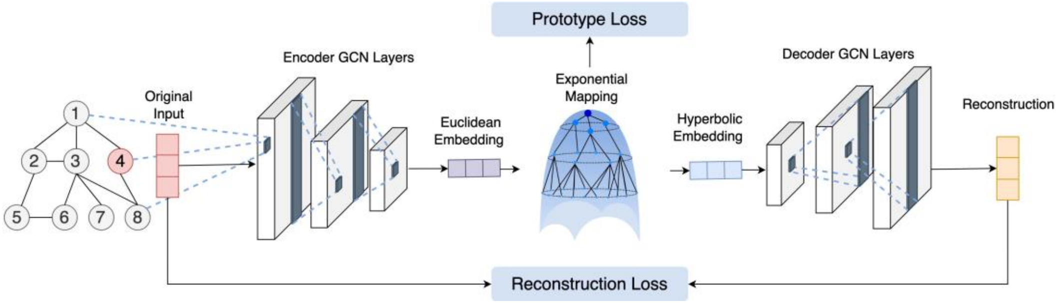


Figure 4. HGCN-based autoencoder architecture.

Curvature parameter. Let $\theta \in \mathbb{R}$ be a trainable parameter and define $c = \exp(\theta)$. This exponential mapping ensures $c > 0$ for all real θ, thereby preserving a valid curvature for hyperbolic geometry. The d-dimensional Poincaré ball of curvature c is given by

$$B_c^d = \left\{ z \in \mathbb{R}^d \mid \|z\| < \frac{1}{\sqrt{c}} \right\}, \tag{7}$$

where ‖ ▪ ‖ is the Euclidean norm.

Class prototypes. Suppose the network involves K class prototypes or centroids, denoted by $G_1, \ldots, G_K \in \mathbb{R}^d$. To exploit hyperbolic geometry, each G_k is constrained to lie in B_c^d . Collectively,

$$G = [G_1, \ldots, G_K]^{\mathrm{T}} \in (B_c^d)^K, \tag{8}$$

so that each prototype is a point in the Poincaré ball. This enables the model to represent different "levels" or clusters in the social hierarchy, with radial distance often correlating with a class's level of generality or specificity.

Encoder. We employ a sequence of GCN layers to transform node embeddings $\{x_i\}_{i=1}^N$ into latent representations $\{z_i\}_{i=1}^N \subset \mathbb{R}^d$. Let $H^{(0)}$ be the input feature matrix. Then, the output of the ℓ-th GCN layer is given by

$$H^{(\ell+1)} = \sigma\left(\widetilde{D}^{-\frac{1}{2}} A \widetilde{D}^{-\frac{1}{2}} H^{(\ell)} W^{(\ell)}\right), \tag{9}$$

where A is the adjacency matrix, $\widetilde{D}$ is its degree matrix, $W^{(\ell)}$ is a trainable weight matrix, and $\sigma(\,▪\,)$ is a nonlinear activation (e.g., LeakyReLU). After such layers, we obtain $Z \in \mathbb{R}^{Nxd}$, whose i-th row is z_i.

Exponential mapping. To capture hierarchical relationships more effectively than in Euclidean space, each z_i is mapped onto the Poincaré ball of curvature c via the exponential map at the origin:

$$z_i^{(hyp)} = exp_0(z_i) = \begin{cases} \tanh\left(\sqrt{c}\,\|z\|\right)\frac{z_i}{\sqrt{c}\,\|z\|}, & z_i \neq 0, \\ 0, \; z_i = 0 \end{cases} \tag{10}$$

By placing both the node embeddings $\left\{z_i^{(hyp)}\right\}$ and the class prototypes $\{G_K\}$ in B_c^d, we introduce a natural way to represent social hierarchies: the distance from the origin can reflect different depths or "levels," and distances among prototypes encode hierarchical relations. This negative curvature setting facilitates tree-like embedding with less distortion than a Euclidean space, making it particularly suited for social network data with pronounced or latent hierarchical structures. The distance-based constraints on $\{G_K\}$ further reinforce ordering relationships, ensuring that class prototypes and consequently the node embeddings associated with them respect the underlying hierarchy more faithfully than in standard Euclidean embedding.

Decoder. A decoder composed of additional GCN layers reconstructs the original features from the hyperbolic embeddings $\left\{z_i^{(hyp)}\right\}$. The decoder produces $\widehat{X} \in \mathbb{R}^{N\,x\,d}$, whose i-th row $\hat{x}_i$ is the reconstructed embedding for node i. During inference time, anomalous nodes are expected to exhibit higher reconstruction error, reflecting their deviation from the learned hyperbolic manifold.

Anomaly score. The anomaly score s_i is determined by computing the mean squared reconstruction error,

$$s_i \;=\; \mathbb{E}_{j\,\in[d_o]}\left[\left(\hat{x}_{i,j} - x_{i,j}\right)^2\right], \tag{11}$$

which measures the discrepancy between the original input and its reconstructed version for a node vector of dimension d_o. A higher anomaly score indicates a greater deviation from the expected benign patterns, suggesting a higher likelihood of an anomaly.

3.2.2 Model Training

This subsection describes the loss function and training strategy. The central objective is to minimize the reconstruction error, measured by the mean squared error

$$\mathcal{L}_{recon} = \frac{1}{N}\sum_{i=1}^{N}\|\hat{x}_1 - x_i\|^2, \tag{12}$$

which encourages each node's reconstructed features to match the original.

In many anomaly detection settings, there may be a single nonanomalous class or a dominant cluster. For instance, if G_1 is the prototype of this nonanomalous class, we can encourage node embeddings to remain near G_1 by adding a distance penalty

$$\mathcal{L}_{proto} = \frac{1}{N}\sum_{i=1}^{N} dist\left(z_i^{(hyp)}, G_1\right), \tag{13}$$

where $z_i^{(hyp)}$ is the hyperbolic embedding of node i. This term helps enforce that normal nodes cluster around the chosen prototype, making anomalous nodes more distinguishable if they lie farther away.

We combine the above components into a total loss function:

$$\mathcal{L}_{total} = \mathcal{L}_{recon} + \mathcal{L}_{proto} \tag{14}$$

By simultaneously minimizing reconstruction error and prototype distance, the model learns hyperbolic embeddings that capture nonanomalous behavior; nodes with high reconstruction error or large distance to G_1 are readily flagged as anomalies, while prototypes remain hierarchically ordered if multiple classes or subclusters exist.

3.3 Uncertainty Quantification with Guarantees

Real-world critical decision-making on social networks often demands not only node-level anomaly predictions but also theoretical assurances about false alarms (i.e., FPRs) and missed detections (i.e., FNRs). To provide such guarantees, we follow a probably approximately correct (PAC)-style calibration approach (Li et al. 2022). Specifically, we hold out a small calibration set of nodes with known labels of nonanomalous and anomalous classes and compute their anomaly scores. We then perform a binomial tail bound search for thresholds T_{fp} and T_{fn} that, with probability at least $1 - \delta$, constrain the false positive and false negative rates to be under ϵ. Formally, for an error parameter $_1$, $\epsilon_2 \in (0, 1)$ and confidence δ_1, $\delta_2 \in (0, 1)$, we have the following guarantees:

$$\mathbb{P}(FPR \leq \epsilon_1) \geq 1 - \delta_1, \mathbb{P}(FNR \leq \epsilon_2) \geq 1 - \delta_2 \tag{15}$$

T_{fp} limits alarms on nonanomalous nodes while T_{fn} safeguards against missing anomalies. Any scores between T_{fn} and T_{fp} may be flagged for additional human inspection. Thus, the resulting PAC certificates allow us to incorporate distribution-free uncertainty quantification, offering measurable performance guarantees for social network anomaly detection.

4. Experiments

Here, we present our experimental setup and evaluation results. We evaluate TAG on two datasets to assess its effectiveness; specifically, we use the Enron fraud dataset and a criminal network dataset for evaluation (see Section 4.1). We compare TAG, which is a text-enriched GNN-based paradigm, with different selection of language models (for text extraction) and GNN models (for graph learning). In addition, we compare TAG with a direct approach where all text is fed into an LLM without using graph topology. We aim to provide a thorough comparison across different baselines to study the performance and computational trade-offs (baseline models are detailed in Section 4.2). The key objectives of our experiments are to answer the following research questions:

- RQ1 model performance: what is the performance of TAG compared with baseline models? (Section 4.4.1)
- RQ2 computation overhead: what is the computation overhead of TAG compared with LLMs? (Section 4.4.2)
- RQ3 ablation study: what is the effect of embedding size on the model's performance? (Section 4.4.3)

To obtain robust and reliable results, we conduct each experiment using five different random seeds and report the average performance metrics across these runs. The following subsections detail the datasets, baselines, and metrics we use to evaluate our approach.

4.1 Datasets

The Enron fraud dataset serves as a well-established public benchmark for analyzing fraudulent email communication patterns. The criminal network dataset contains a synthetic criminal network with news reports on the interaction among members. Using this combination of datasets ensures that our evaluation covers hierarchical social networks from different domains.

Enron fraud dataset. The Enron fraud dataset captures email communications within the Enron Corporation, reflecting the hierarchical structure of the organization. The dataset consists of 4,775 nodes, each representing an employee, and 164,443 directed edges corresponding to mail exchanges between employees. Each edge is associated with the textual content of the emails, providing additional contextual information about interactions. The primary task formulated on this dataset is a binary classification problem, where the goal is to identify executives involved in the Enron scandal. According to publicly available records from news sources, 24 employees were implicated in the scandal and serve as the class one, while the remaining employees form the class zero.

Criminal network dataset. The criminal network dataset (Mittrick et al. 2012) represents a structured criminal network with a hierarchy based on subordinate relationships. It consists of 635 nodes with each node corresponding to an individual and 775 undirected edges representing the interactions between them. Each edge contains a news article describing the activities that occurred during the interaction. Among the 635 individuals, 18 are labeled as the class one due to their engagement in suspicious activities, while the remaining individuals are considered the class zero.

Dataset splitting strategy. To ensure a structured and realistic evaluation, the datasets are partitioned into training, calibration, and test sets. The training set consists of only class zero nodes to enable the model to learn benign behavioral patterns without exposure to anomalous instances. The calibration set contains both class zero and one nodes, allowing for the fine-tuning of detection thresholds. The test set includes both class zero and one nodes to evaluate the model's generalization to unseen anomalies.

To address the imbalanced nature of the dataset, we partition the data as follows: 10% of class zero nodes are randomly selected for calibration, another 10% for testing, and the remaining class zero nodes are used for training. For class one nodes, 50% are allocated to the calibration set, while the rest are assigned to the test set. A summary of the dataset statistics can be found in Table 1.

Table 1. Dataset statistics and splits.

Dataset	Anomaly ratio	Nodes	Edges	Train	Calibration	Test
Enron	0.5%	4775	164443	4267	266	242
Criminal	2.8%	635	775	564	36	34

4.2 Baseline Models

To evaluate the effectiveness of our approach, we compare it against multiple baselines that capture different aspects of textual and structural learning. TAG integrates both text and topology by using MiniLM for textual edge extraction and an HGCN-based autoencoder for learning hierarchical node representations in hyperbolic space. This combination allows efficient encoding of communication content while preserving the relational structure of the network.

We organize the baselines into two paradigms: 1) text-enriched GNN-based paradigm, where textual information is incorporated into graph-based models, and 2) LLM-as-predictor paradigm, where an LLM processes all text directly without explicit graph learning.

4.2.1 Text-Enriched GNN-based Paradigm

This paradigm integrates both structural and textual information for anomaly detection. Language models extract message content to enhance node representations, while GNNs are used to model network topology.

MiniLM–HGCN (TAG). Our proposed approach uses MiniLM (Wang et al. 2020), a lightweight transformer-based language model, to extract textual information from messages and emails, producing compact 384-dimensional embeddings. These embeddings serve as input features for the HGCN-based autoencoder that captures hierarchical structures in the network. The HGCN-based autoencoder is trained with an input dimension of 64, a hidden layer dimension of 64, and a latent embedding size of 64, effectively modeling both text and topology.

MiniLM–GCN. To evaluate the effect of hierarchical representation, we replace the HGCN with a GCN-based autoencoder. This model uses the same MiniLM-generated 384-dimensional embeddings but applies a GCN instead of a HGCN for learning node representations based purely on Euclidean graph topology.

Mistral–GCN. In this baseline, we replace MiniLM with Mistral-7B (Jiang et al. 2023), a larger-scale language model that generates 4096-dimensional embeddings, potentially providing richer textual representations with longer processing time. These embeddings are then used as input to a GCN-based autoencoder.

Mistral–HGCN. To analyze whether richer text embeddings improve hierarchical representation learning, we use Mistral-7B for textual feature extraction and a HGCN for structural learning.

4.2.2 LLM-as-Predictor Paradigm

In this paradigm, we assess a direct approach where an LLM is used to process entire interaction histories for node classification without explicitly modeling graph structure.

GPT-4o Mini. Rather than using a GNN, this baseline feeds all text into GPT-4o Mini (OpenAI 2024), allowing the LLM to infer node labels directly from raw textual content. This approach evaluates whether a purely text-based model can capture relational structures and anomalies without explicit graph learning.

4.3 Evaluation Metrics

We evaluate the anomaly detection model using key metrics that assess both detection effectiveness and reliability in imbalanced settings. The precision-recall area under the curve (PRAUC) is computed over all possible decision thresholds and quantifies the trade-off between precision and recall. This metric is particularly suited for anomaly detection (Saito and Rehmsmeier 2015), where the class one (anomalies) is rare. A larger value indicates better performance.

In addition, we also include several metrics that are evaluated at a specific decision threshold to evaluate the PAC-style guarantee in Section 3.3. The true positive rate (TPR), also known as recall or sensitivity, is given by TPR = $\frac{TP}{TP+FN}$ and measures the proportion of detected anomalies, and we want a higher value for this metric. The FPR, defined as $\frac{FP}{FP+TN}$, quantifies the proportion of normal instances misclassified as anomalies. A lower FPR reduces false alarms, improving system reliability. The FNR, given by $\frac{FN}{TP+FN}$, represents the proportion of missed anomalies, where a low FNR is critical to ensure that high-risk anomalies are detected. We evaluate on the minimal error rates guaranteed (ϵ) under the confidence at 95% level (δ = 0.05). A smaller guarantee value indicates tighter bound on FPR and FNR; hence, it is more desirable.

In summary, a good model should maximize PRAUC and optimize the threshold-dependent metrics (TPR, FPR, and FNR), while maintaining tight guarantees over misclassification rates.

4.4 Evaluation

We evaluate our approach across three key aspects: 1) model performance, comparing TAG against baseline methods and LLM; 2) computation overhead, analyzing efficiency in terms of parameters, memory, and throughput; and 3) ablation study, examining the impact of embedding size on detection performance. Our results demonstrate our MiniLM–HGCN combination achieves competitive accuracy while being significantly more lightweight and computationally efficient, making it well-suited for scalable anomaly detection.

4.4.1 RQ1 Model Performance

The evaluation results in Table 2 demonstrate that TAG achieves the best overall performance across datasets. Notably, TAG achieves the highest PRAUC, surpassing GPT-4o-mini by 11.64% on the Enron dataset and 8.13% on the criminal dataset. In terms of TPR, TAG exhibits a 36.36% improvement over GPT-4o-mini on the Enron dataset, indicating its enhanced ability to correctly identify anomalies. In addition, its TPR and FNR remains comparable to GPT-4o-mini on the criminal dataset, suggesting robustness across different datasets.

Table 2. Anomaly detection comparison.

Dataset	Model name	PRAUC (↑)	TPR (↑)	FPR (↓)	FNR (↓)	Guarantee (↓)
Enron	GPT-4o-mini	0.3566 ± 0.1065	0.5500 ± 0.1728	**0.0806 ± 0.0243**	0.4500 ± 0.1728	. . .
	Mistral-GCN	0.2444 ± 0.0750	0.4500 ± 0.1453	0.4395 ± 0.0545	0.5500 ± 0.1453	0.5020 ± 0.0384
	Mistral-HGCN	0.2292 ± 0.0580	0.4250 ± 0.1146	0.5403 ± 0.0544	0.5750 ± 0.1146	0.5870 ± 0.0600
	MiniLM-GCN	0.2923 ± 0.1189	0.5416 ± 0.2339	0.4590 ± 0.0743	0.4584 ± 0.2339	0.5200 ± 0.0732
	TAG	**0.3981 ± 0.0541**	**0.7500 ± 0.1086**	0.3908 ± 0.0536	**0.2500 ± 0.1086**	**0.4600 ± 0.0438**
Criminal	GPT-4o-mini	0.5780 ± 0.0861	**0.7778 ± 0.1646**	0.2742 ± 0.0872	**0.2222 ± 0.1646**	. . .
	Mistral-GCN	0.2798 ± 0.0615	0.3667 ± 0.1000	0.4790 ± 0.1587	0.6333 ± 0.1000	0.6090 ± 0.1469
	Mistral-HGCN	0.3800 ± 0.0833	0.5111 ± 0.1587	0.3355 ± 0.1330	0.4889 ± 0.1587	0.5010 ± 0.0951
	MiniLM-GCN	0.3986 ± 0.0965	0.5444 ± 0.1681	0.4000 ± 0.1863	0.4556 ± 0.1681	0.4850 ± 0.2157
	TAG	**0.6250 ± 0.0723**	**0.7778 ± 0.2183**	**0.1935 ± 0.1352**	**0.2222 ± 0.2183**	**0.3900 ± 0.1693**

Note: Best detection is shown in bold.

Furthermore, a key advantage of TAG over LLM is the ability to provide statistical guarantees on detection performance. In Table 2, the FPR and FNR columns have values smaller than that of the guarantee column, indicating our theoretical guarantees are empirically validated. The text-enriched GNN-based paradigm allows for more control on the node anomaly scores than LLM, allowing us to guarantee FPR and FNR for users. This is critical in applications where reliability and trustworthiness are important. In contrast, large-scale black-box models such as GPT-4o-mini lack such guarantees, making them less suitable for applications requiring explainability and robustness for decision-making.

Another finding is that increasing the language model size for textual-edge extraction does not necessarily yield better performance. TAG achieves a 66.49% higher PRAUC than Mistral–HGCN on average, despite Mistral being a larger language model. This result suggests that larger generative models like Mistral 7B are not inherently optimized for structured text representation in graph-based anomaly detection. MiniLM, a task-specific and lightweight model designed for efficient text embedding extraction, produces more discriminative and computationally efficient edge embeddings, making it better suited for this task.

HGCN-based autoencoders outperform GCN-based autoencoders by an average of 32.51% in PRAUC across both datasets. This improvement stems from the HGCN's ability to efficiently capture hierarchical structures using hyperbolic space, which better preserves distances and relationships in complex networks. Additionally, the HGCN requires fewer parameters to achieve expressive representations, reducing overfitting while maintaining anomaly separability. Its superior handling of sparse graph structures further enhances its ability to distinguish anomalous nodes. These advantages make the HGCN a more effective model for anomaly detection compared to traditional GCN-based approaches on hierarchical social networks.

Overall, TAG provides the tightest guarantee across all language model and GNN architecture choices.

4.4.2 RQ2 Computation Overhead

To evaluate the computational requirements of our approach, we compare TAG with baseline models in Table 3. TAG consists of approximately 22 million parameters and requires 90.05 MB of memory. In contrast, GPT-4o-mini is estimated to contain over billions of parameters, with a memory footprint exceeding 1,000 MB. This comparison highlights the difference in computational demands between TAG and large-scale language models. While our approach effectively leverages graph-based and language modeling techniques, it remains lightweight, making it feasible for deployment in resource-constrained environments.

Table 3. Model parameters and size.

Model	Total parameters	Size (MB)
MiniLM	22 million	Approx. 90
Mistral	7 billion	Approx. 13,000
GCN	17,006	0.06
HGCN	9698	0.04
TAG	Approx. 22 million	90.05
GPT-4o-mini	Not disclosed (>1 billion)	Not disclosed

In addition, the HGCN requires 86.87% fewer parameters than GCN, as shown in Table 3. This reduction is primarily due to the efficient representation capabilities of hyperbolic space, which allows the HGCN to encode hierarchical structures with fewer dimensions. Unlike standard GCNs that rely on large weight matrices for Euclidean feature transformations, the HGCN uses Möbius transformations, which require fewer learnable parameters while maintaining expressivity. Additionally, hyperbolic convolutions exploit the curvature of the space to naturally capture complex relationships, reducing the need for additional layers or large feature maps. Consequently, the HGCN achieves a smaller model size while preserving comparable performance with the GCN.

Table 4 presents the computational overhead of different language models used for textual-edge embedding extraction, while Table 5 compares the training and evaluation times for TAG and GPT-4o-mini. Our approach using MiniLM achieves the highest throughput of 200.30 inputs/s, compared with 22.17 for GPT-4o-mini and 3.43 for Mistral-7B. The total processing time for MiniLM is 0.23 h, while the other models require 2.06 and 13.33 h, respectively. Additionally, MiniLM operates at zero cost per token, whereas GPT-4o-mini incurs a cost of $0.02 per 1K tokens. Since GPT-4o-mini is a pretrained model, it does not require additional training time, whereas the HGCN and GCN require approximately 10.08 and 7.26 min, respectively. The inference time for GPT-4o-mini is around 8.77 times the time for the HGCN and 15.28 times for the GCN. TAG provides faster inference, making it suitable for time-sensitive decision-making. GPT-4o-mini, on the other hand, eliminates training time but has higher computational and financial costs. Each method has its strengths, and the choice depends on the specific requirements of the application.

Table 4. Computation overhead comparison between language models for extracting textual-edge embeddings, showing computation time per edge, total processing time, and throughput.

Method	Time per message (s)	Total time (h)	Throughput (input/s)	Coste per 1k token ($)
MiniLM	0.00499	0.23	200.30	0
Mistral-7B	0.292	13.33	2.43	0
GPT-4o-mini	0.0451	2.06	22.17	0.02

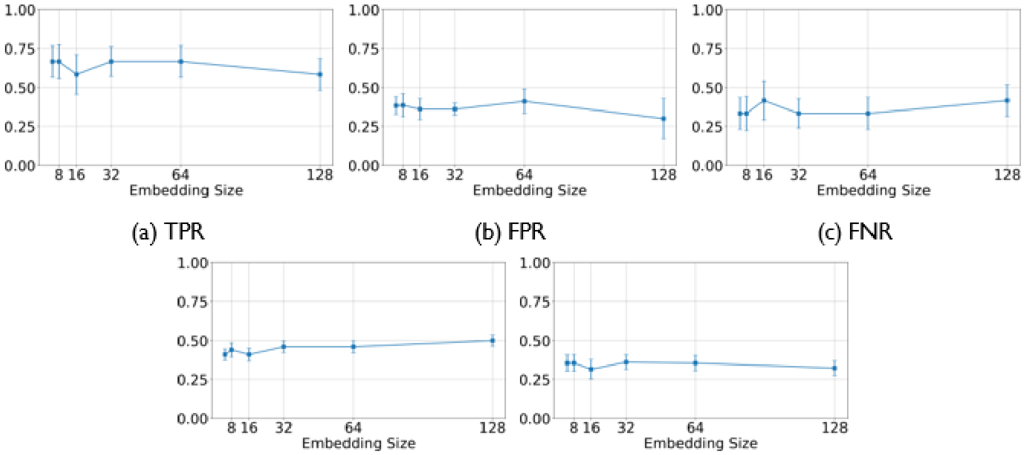
Table 5. Comparison of training and inference times for HGCN, GCN, and GPT-4o-mini.

Model	Training time (s)	Inference time (s)
HGCN	604.84 ± 43.25	11.81 ± 2.80
GCN	435.55 ± 28.11	6.78 ± 0.86
GPT-4o-mini	. . .	103.60 ± 8.11

4.4.3 RQ3 Ablation Study

This section investigates the impact of node embedding size (d_o from Section 3.1) on the performance of TAG. We analyze how different embedding sizes affect key evaluation metrics across both the Enron and criminal datasets to determine the most effective configuration for anomaly detection.

Enron dataset analysis. From Figure 5, our results indicate that embedding sizes 32 and 64 provide the best balance between stability and performance. The TPR remains stable at these sizes, ensuring effective anomaly detection. However, at smaller embedding sizes (8 and 16), TPR exhibits higher variance, suggesting unstable detection. Conversely, at 128, TPR declines slightly, indicating a potential loss of discriminative power.



(a) TPR

(b) FPR

(c) FNR

(d) Guarantee

(e) AUPRC

Figure 5. Performance metrics (TPR, FPR, FNR, Epsilon, and area under the precision-recall curve [AUPRC]) across different embedding sizes for the Enron dataset.

FPR shows a slight increase at 64 but decreases at 128, implying that larger embeddings may help reduce false positives. However, this comes at the cost of increased FNR at 128, meaning more anomalies are missed. FNR is lowest at 16 and 32, highlighting a balanced detection capability, but at 128, redundancy in the representation leads to a higher FNR.

Guarantee remains relatively stable across all embedding sizes, indicating that uncertainty is not significantly impacted. The area under the AUPRC is highest at 32 and 64, confirming these sizes yield the most effective anomaly detection. At 128, AUPRC declines slightly due to the increased FNR.

Criminal dataset analysis. A similar trend is observed in the criminal dataset in Figure 6, where embedding sizes 32 and 64 yield the most consistent results. TPR remains stable at these values, capturing anomaly patterns effectively. However, at smaller embedding sizes (8 and 16), TPR fluctuates, indicating less stable feature representation. At 128, TPR slightly decreases, suggesting that increasing embedding size beyond 64 does not provide additional benefits.

FPR increases slightly at 64 but drops at 128, suggesting that larger embeddings may reduce false alarms. However, this comes at the cost of increased FNR at 128, similar to the Enron dataset. The lowest FNR is observed at 16 and 32, balancing detection effectiveness, but at 128, excessive feature smoothing reduces recall.

Guarantee remains stable across all embedding sizes, indicating that model uncertainty is not significantly affected. AUPRC is highest at 32 and 64, reinforcing these sizes as the best choices. At 128, AUPRC declines due to increased FNR, demonstrating a trade-off between false alarms and missed detections.

Summary. Across both datasets, embedding sizes 32 and 64 provide the best balance between stability and detection performance. While larger embeddings (128) reduce FPR, they also increase FNR, leading to a decline in AUPRC.

Conversely, smaller embeddings (8 and 16) introduce higher variance, making anomaly detection less stable. Notably, guarantee remains consistent across all embedding sizes, indicating that model uncertainty is not significantly influenced by this parameter.

5. Related Works

This section provides a literature review on identifying anomalous nodes in social networks with hierarchical structures, focusing on the integration of language model and hyperbolic graph neural networks.

Early methods for anomaly detection in social networks primarily relied on traditional network analysis like centrality features (Das and Sinha 2018; Chang et al. 2021), PageRank (Azarm et al. 2024), and DeepWalk (Van Belle et al. 2023). While these approaches provided valuable insights into network structure, they often overlooked the rich textual data associated with nodes and edges, limiting their effectiveness in complex social networks with web data. Our approach integrates both structural and textual information on edges, thereby providing a more comprehensive analysis.

Recognizing the importance of textual data, researchers began to integrate natural language processing techniques into anomaly detection frameworks (Boulieris et al. 2024). For instance, some works proposed a topic modeling approach to detect security risks on websites (Carmichael et al. 2023) and social media networks (Celebi and Shashidhar 2022). Similarly, another work developed a method combining sentiment analysis and entity recognition for identifying unusual patterns in online communities (Wen et al. 2023). These approaches significantly improved the detection of content-based anomalies but often failed to fully leverage the structural dependencies within the network. In contrast, TAG seamlessly integrates textual and structural data, enhancing the detection of anomalies that manifest through both content and connections.

In addition, recent advancements in LLMs (Liu et al. 2024; Russell-Gilbert et al. 2024) show promise for processing textual data in graph-based applications. However, the inference latency and computational cost introduced by such models are often nonnegligible, hindering scalable deployment. Moreover, for sensitive information, it is preferable to keep the analysis local rather than relying on cloud-based solutions. TAG addresses these challenges by leveraging lightweight language models that can be kept locally for analysis, achieving comparable detection performance while ensuring efficiency and data privacy.

The advent of GNNs marked a significant advancement in capturing structural dependencies for anomaly detection (Kim et al. 2022; Tang et al. 2022; Qiao et al. 2024) in social network. Many works demonstrated the effectiveness of GNNs in identifying anomalous nodes by learning representations that incorporate both node features and graph structure (Wang and Yu 2022; Kisanga et al. 2023; Marfo et al. 2024; Zheng et al. c2024). However, many of these approaches overlook the hierarchical nature of social networks. Our research addresses this gap by employing hyperbolic geometry to naturally model hierarchical relationships within the network.

To address the limitations in modeling hierarchical structures, recent research has turned to hyperbolic geometry (Gu and Zou 2024). Hyperbolic graph neural networks (HGNNs) leverage the properties of hyperbolic space to better represent hierarchical data (Chami et al. 2019). Building on this, some works explored the use of HGNNs to improve the performance on graph data with inherent hierarchical structures across domains (Fu et al. 2024; Touahria et al. 2024). While these studies advanced the field, they did not incorporate textual edge information and uncertainty quantification in their models. As the field progresses, the importance of uncertainty quantification in anomaly detection has become increasingly apparent. We refer to the survey (Wang et al. 2024) that highlighted the need for uncertainty-aware GNNs in high-stakes decision-making scenarios. Our approach extends this line of work by showing how to construct a text-enriched HGCN and integrating uncertainty quantification to complete the pipeline.

In summary, unlike previous works that focused either solely on structural or textual information, our approach combines both aspects within a hyperbolic space, enabling more accurate modeling of hierarchical relationships. Furthermore, we address a critical gap in existing literature by incorporating uncertainty quantification, providing not just predictions but also confidence measures for those predictions. Our approach aims to enhance both the accuracy and reliability of anomaly detection in complex, hierarchical social networks.

6. Conclusion

We proposed a novel framework for anomalous node classification in hierarchical social networks by integrating textual information, HGCNs, and uncertainty quantification. TAG extracts edge embeddings from textual data, transforms them into node embeddings via convolution, and models hierarchical structures using HGCNs. Unlike existing methods that overlook either textual content or network hierarchy, TAG captures both, offering a more accurate anomaly detection system. Additionally, we provide PAC-guaranteed uncertainty quantification, ensuring reliable predictions for high-stakes decision-making. Empirical results on the Enron dataset and an in-house dataset demonstrate that TAG achieves comparable performance with LLM-based approaches while significantly reducing computational overhead. This makes TAG efficient, privacy-preserving, and well-suited for applications like corporate fraud detection and forensic analysis in criminal networks.

7. References

Azarm C, Acar E, van Zeelt M. On the potential of network-based features for fraud detection. arXiv; 2024 Feb 14. arXiv:2402.09495. https://doi.org/10.48550/arXiv.2402.09495

Boulieris P, Pavlopoulos J, Xenos A, Vassalos V. Fraud detection with natural language processing. Mach Learn. 2024;113(8):5087–5108.

Carmichael JJ, Eaton SE. Security risks, fake degrees, and other fraud: a topic modelling approach. Fake degrees and fraudulent credentials in higher education. Springer; 2023. p. 227–250.

Celebi N, Shashidhar N. Topic modeling in the Enron dataset. Big Data – BigData 2022, 11th International Conference, held as part of the Services Conference Federation, SCF 2022; 2022 Dec 10–14; Honolulu, HI. Springer; c2022. p. 27–34.

Chami I, Ying R, Ré C, Leskovec J. Hyperbolic graph convolutional neural networks. Adv Neural Inf Process Syst. 2019;32:4869–4880.

Chang YC, Lai KT, Chou SCT, Chiang WC, Lin YC. Who is the boss? Identifying key roles in telecom fraud network via centrality-guided deep random walk. Data Technol Appl. 2021;55(1):1–18.

Cheng C-H, Cai W-H. Double-weight LDA extracting keywords for financial fraud detection system. Multim Tools Appl. 2024;83(17):50757–50781.

Das K, Sinha SK. Centrality measure based approach for detection of malicious nodes in Twitter social network. Int J Eng Technol. 2018;7(4.5):518.

Deprez B, Vandervorst F, Verbeke W, Verdonck T, Baesens B. Network analytics for insurance fraud detection: a critical case study. Eur Actuar J. 2024;14(3): 965–990.

Ekle OA, Eberle W. Dynamic PageRank with decay: a modified approach for node anomaly detection in evolving graph streams. The International FLAIRS Conference Proceedings. 2024;37(1): 37.1.135553.

Fu Y et al. HC-GLAD: dual hyperbolic contrastive learning for unsupervised graph-level anomaly detection. arXiv; 2024 July 2. arXiv:2407.02057. https://doi.org/10.48550/arXiv.2407.02057

Gu J, Zou D. Three revisits to node-level graph anomaly detection: Outliers, message passing and hyperbolic neural networks. Proceedings of the Second Learning on Graphs Conference; 2023 Nov 27–30; virtual. PMLR; c2024.

Jiang AQ et al. Mistral 7B. arXiv; 2023 Oct 10. arXiv:2310.06825. https://doi.org/10.48550/arXiv.2310.06825

Kim H, Lee BS, Shin WY, Lim S. Graph anomaly detection with graph neural networks: current status and challenges. IEEE Access. 2022;10:111820–111829.

Kisanga P, Woungang I, Traore I, Carvalho GH. Network anomaly detection using a graph neural network. 2023 International Conference on Computing, Networking and Communications (ICNC); 2023 Feb 20–22; Honolulu, HI. IEEE; c2023. p. 61–65.

Li S, Ji X, Dobriban E, Sokolsky O, Lee I. PAC-wrap: semi-supervised PAC anomaly detection. Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining; 2022 Aug 14–18; Washington, DC. p. 945–955.

Liu S et al. AnomalyLLM: few-shot anomaly edge detection for dynamic graphs using large language models. arXiv; 2024 May 13. arXiv:2405.07626. https://doi.org/10.48550/arXiv.2405.07626

Marfo W, Tosh DK, Moore SV. Enhancing network anomaly detection using graph neural networks. 2024 22nd Mediterranean Communication and Computer Networking Conference (MedComNet); 2024. p. 1–10. IEEE.

Mittrick MR, Roy HE, Kase SE, Bowman EK. Refinement of the Ali Baba data set. Army Research Laboratory (US); 2012 Mar. Report No.: ARL-TN-0476.

OpenAI. GPT-4o mini: advancing cost-efficient intelligence. OpenAI; 2024 July 18. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

Park D-Y. Graph-theoretical approach to enhance accuracy of financial fraud detection using synthetic tabular data generation. Proceedings of the 33rd ACM International Conference on Information and Knowledge Management; 2024 Oct 21–25; Boise, ID. ACM; 2024. p. 5467–5470.

Qiao H et al. Deep graph anomaly detection: a survey and new perspectives. arXiv; 2024 Sep 16. arXiv:2409.09957. https://doi.org/10.48550/arXiv.2409.09957

Russell-Gilbert A et al. RAAD-LLM: adaptive anomaly detection using large language models. 2024 IEEE International Conference on Big Data (BigData); 2024 Dec 15–18; Washington, DC. IEEE; 2024. p. 4194–4203.

Saito T, Rehmsmeier M. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. PLoS One. 2015;10(3):e0118432.

Tang J, Li J, Gao Z, Li J. Rethinking graph neural networks for anomaly detection. International conference on machine learning. Proceedings of the 39th International Conference on Machine Learning; 2022 July 17–23; Baltimore, MD. PMLR; 2024.

Touahria Miliani MY, Sadat SA, Haddad M, Seba H, Amrouche K. Comparing hyperbolic graph embedding models on anomaly detection for cybersecurity. ARES '24: Proceedings of the 19th International Conference on Availability, Reliability and Security; 2024 July 30–Aug 2; Vienna, Austria. p. 1–11.

Van Belle R, Baesens B, De Weerdt J. CATCHM: a novel network-based credit card fraud detection method using node representation learning. Decis Support Syst. 2023;164:113866.

Wang F et al. Uncertainty in graph neural networks: a survey. arXiv; 2024 Mar 11. arXiv:2403.07185. https://doi.org/10.48550/arXiv.2403.07185

Wang S, Yu PS. Graph neural networks in anomaly detection. Graph neural networks: foundations, frontiers, and applications. Springer; 2022. p. 557–578.

Wang W et al. MiniLM: deep self-attention distillation for task-agnostic compression of pre-trained transformers. Adv Neural Inf Process Syst. 2020;33:5776–5788.

Wen X et al. An approach to internal threats detection based on sentiment analysis and network analysis. J Inf Sec Appl. 2023;77:103557.

Zheng X, Wu B, Zhang AX, Li W. Improving robustness of GNN-based anomaly detection by graph adversarial training. Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024); 2024 May 20–25, Torino, Italy. p. 8902–8912.

List of Symbols, Abbreviations, and Acronyms

AI	artificial intelligence
ARL	Army Research Laboratory
AUPRC	area under the precision-recall curve
DEVCOM	U.S. Army Combat Capabilities Development Command
FNR	false negative rate
FPR	false positive rate
GCN	graph convolutional network
GNN	graph neural network
HGCN	hyperbolic graph convolutional network
HGNN	hyberbolic graph neural network
LLM	large language model
PAC	probably approximately correct
PRAUC	precision-recall area under the curve
ReLU	rectified linear unit
TAG	Text-enriched Anomaly detection on hierarchical Graphs
TPR	true positive rate

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

1 DEVCOM ARL
(PDF) FCDD RLB CI
TECH LIB