# LEX-GNN: Label-Exploring Graph Neural Network for Accurate Fraud Detection

CIKM 2024

**Woochang Hyun[1], Insoo Lee[2], Bongwon Suh[1]**

[1] Human Centered Computing Lab, Seoul National University
[2] Digital Forensic Center, Supreme Prosecutors' Office, Republic of Korea

HCC LAB SNU

## Introduction

### Background

- GNNs are promising for fraud detection due to their ability to capture relational data. However, they face challenges such as class imbalance and heterophilic connections.

### Existing Approaches

- **Node-based:** Focuses on similar nodes but struggles when fraudsters resemble legitimate ones.
- **Edge-based:** Discerns homophilic from heterophilic edges but can be computationally expensive as #edge ≫ #node.
- **Label-based:** Incorporates label information but has limitations in properly handling unlabeled nodes, as it treats them equally.

### Proposed Framework

- **LEX-GNN:** Label-Exploring Graph Neural Network
- Predicts the soft labels of nodes in advance utilizing a semi-supervised learning approach.
- Adjusts message passing and aggregation pipeline dynamically based on the predicted labels.

## Method

### Label Exploration

- We predict the soft (probable) label of nodes using a double-layer MLP at the beginning of each layer.

$$p_v^{(l)} = \Phi_{\text{pre}}^{(l)}(h_v^{(l-1)})$$

### LEX Aggregation

- **Differentiated message passing:** We incorporate the label embedding into the node representation and interpolate the outgoing message according to the fraudulent probability.

$$\tilde{m}_v^{(l)} = h_v^{(l-1)} + \Psi^{(l)}(p_v^{(l)}), \quad \Psi^{(l)}(p_v^{(l)}) = p_v^{(l)}\Psi_1^{(l)} + (1 - p_v^{(l)})\Psi_0^{(l)}$$

$$m_v^{(l)} = \mathcal{W}_{\text{s},v}^{(l)}(\tilde{m}_v^{(l)}), \quad \mathcal{W}_{\text{s},v}^{(l)} = p_v^{(l)}\mathcal{W}_{\text{s},1}^{(l)} + (1 - p_v^{(l)})\mathcal{W}_{\text{s},0}^{(l)}$$

- **Attention-based aggregation:** The messages are aggregated through multi-head dynamic attention.

$$\tilde{h}_u^{(l)} = \sigma\left(\frac{1}{H}\sum_{h=1}^{H}\sum_{v\in\mathcal{N}_u} \alpha_{uv}^{(l,h)}\mathcal{W}_a^{(l,h)} m_v^{(l)}\right), \quad \alpha_{uv}^{(l,h)} = \frac{\exp\left(a^{(l,h)}\cdot\sigma\left(\mathcal{W}_a^{(l,h)}\left(m_u^{(l)} + m_v^{(l)}\right)\right)\right)}{\sum_{v'\in\mathcal{N}_u}\exp\left(a^{(l,h)}\cdot\sigma\left(\mathcal{W}_a^{(l,h)}\left(m_u^{(l)} + m_{v'}^{(l)}\right)\right)\right)}$$

- **Differentiated message reception:** Destination nodes also incorporate the aggregated messages based on their own predicted labels to make final representations.

$$h_u^{(l)} = \mathcal{W}_{\text{self}}^{(l)}(h_u^{(l-1)}) + \mathcal{W}_{\text{d},u}^{(l)}(\tilde{h}_u^{(l)}), \quad \mathcal{W}_{\text{d},u}^{(l)} = p_u^{(l)}\mathcal{W}_{\text{d},1}^{(l)} + (1 - p_u^{(l)})\mathcal{W}_{\text{d},0}^{(l)}$$

### Optimization

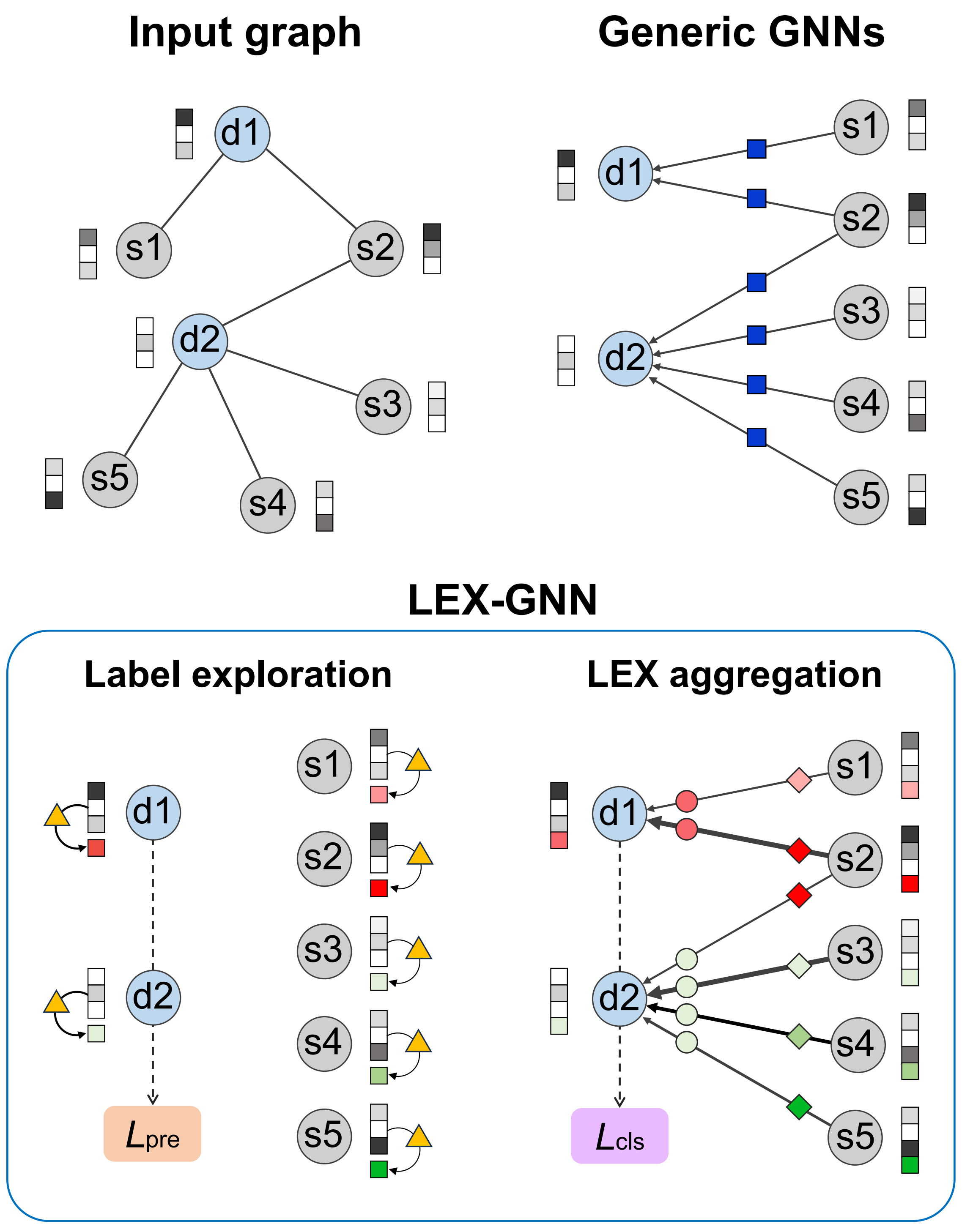
- The overall loss is formulated reflecting the accuracy of label exploration.

$$\mathcal{L}_{\text{cls}} = -\sum_{u\in\mathcal{V}_{\text{t}}}[y_u\log(q_u) + (1 - y_u)\log(1 - q_u)], \quad q_u = \Phi_{\text{cls}}(h_u^{(L)})$$

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \beta\mathcal{L}_{\text{pre}}^{(L)}, \quad \mathcal{L}_{\text{pre}}^{(l)} = -\sum_{v\in\mathcal{V}_{\text{t}}}[y_v\log(p_v^{(l)}) + (1 - y_v)\log(1 - p_v^{(l)})]$$

## Conclusion

- LEX-GNN effectively detects fraud by integrating label predictions into the message passing process, even when dealing with unlabeled data.
- The dual transformation on both the source and destination ends with attention effectively captures the relational characteristics of the graph.
- The superior accuracy and robustness of the proposed framework have been validated through experiments on real-world datasets.



In generic GNNs, source nodes pass their representations as messages toward destination nodes, applying a learnable weight (■) uniformly.

LEX-GNN explores the probability of a node being fraudulent through a neural classifier (▲).

Based on the prediction, source nodes transform (●) and send messages, while destination nodes adjust their reception adaptively (◆).

**Figure 1: A conceptual diagram of LEX-GNN.**

## Results

### Dataset

- Yelp: 6,677 fraud reviews / 45,954 (14.53%)
- Amazon: 821 fraudsters / 11,944 (6.87%)

### Performance Comparison

- LEX-GNN outperforms state-of-the-art baselines in all metrics, except for F1-macro on Amazon with 1% training condition.
- The performances on Yelp with 40% training are significant, showing improvements of over 3.05% in AUC and 5.29% in F1-macro.

**Table 1: Performance comparison on *Yelp* and *Amazon* under 40% and 1% training conditions.**

| Method | Dataset | *Yelp* (40%) | | *Amazon* (40%) | | *Yelp* (1%) | | *Amazon* (1%) | |
|---|---|---|---|---|---|---|---|---|---|
| | Metric | AUC | F1-macro | AUC | F1-macro | AUC | F1-macro | AUC | F1-macro |
| Baselines | CARE-GNN | 80.15±1.13 | 67.41±1.61 | 94.18±0.85 | 89.35±0.72 | 72.91±0.86 | 59.51±1.27 | 89.35±2.37 | 65.34±1.68 |
| | PC-GNN | 81.64±1.08 | 65.72±1.52 | 94.42±0.75 | 89.82±1.64 | 74.12±1.51 | 57.63±1.54 | 90.70±2.16 | 72.18±2.47 |
| | H$^2$-FDetector* | 89.48±1.26 | 74.38±2.42 | 96.03±0.69 | 86.91±1.01 | 74.19±0.52 | 57.42±0.45 | 83.26±0.17 | 67.60±0.31 |
| | GHRN* | 90.57±0.36 | 77.54±1.02 | 97.07±0.73 | 92.36±0.97 | 76.76±0.37 | 64.30±0.61 | 90.27±0.30 | **89.16±0.89** |
| | GTAN | 92.86±0.60 | 80.12±0.26 | 96.65±0.41 | 92.53±0.48 | 81.05±1.02 | 66.10±0.77 | 91.70±1.54 | 85.67±1.34 |
| | PMP | 93.54±0.25 | 82.01±0.70 | 97.37±0.18 | 92.29±0.27 | 81.58±0.57 | 67.15±0.56 | 91.55±0.80 | 82.49±1.26 |
| Ours | **LEX-GNN** | **96.40±0.28** | **86.35±0.39** | **97.91±0.15** | **93.48±0.25** | **83.14±0.53** | **69.73±0.68** | **93.02±0.21**† | 87.33±1.76† |

* Results of H$^2$-FDetector and GHRN are obtained from [28]. † Results on *Amazon* of 1% learning are derived from a single-layer LEX-GNN with $\beta = 1$.

### Ablation Study

- All components are beneficial, the exclusion of attention on Yelp and the source-side transformation on Amazon has the most detrimental impact.

### Effect of Label Exploration

- When $\beta > 0$, label exploration is activated, and the accurate $p$ is fed into the LEX aggregation, which enhances the overall performance.
- In addition to conventional pseudo-labeling which primarily aims at data augmentation, our approach adjusts the message passing based on predicted labels, thus enabling more precise relational representation.

**Table 2: Ablation study. All components are beneficial.**

| Method | *Yelp* (40%) | | *Amazon* (40%) | |
|---|---|---|---|---|
| | AUC | F1-macro | AUC | F1-macro |
| **LEX-GNN** | **96.40±0.28** | **86.35±0.39** | **97.91±0.15** | **93.48±0.25** |
| – label emb. | 96.08±0.26 | 85.89±0.27 | 97.75±0.24 | 93.24±0.19 |
| – src. trans. | 96.30±0.15 | 86.16±0.14 | 97.16±0.15 | 92.74±0.67 |
| – dst. trans. | 95.87±0.39 | 85.17±0.08 | 97.78±0.10 | 93.22±0.23 |
| – attention | 94.61±0.25 | 82.85±0.53 | 97.54±0.01 | 93.10±0.27 |

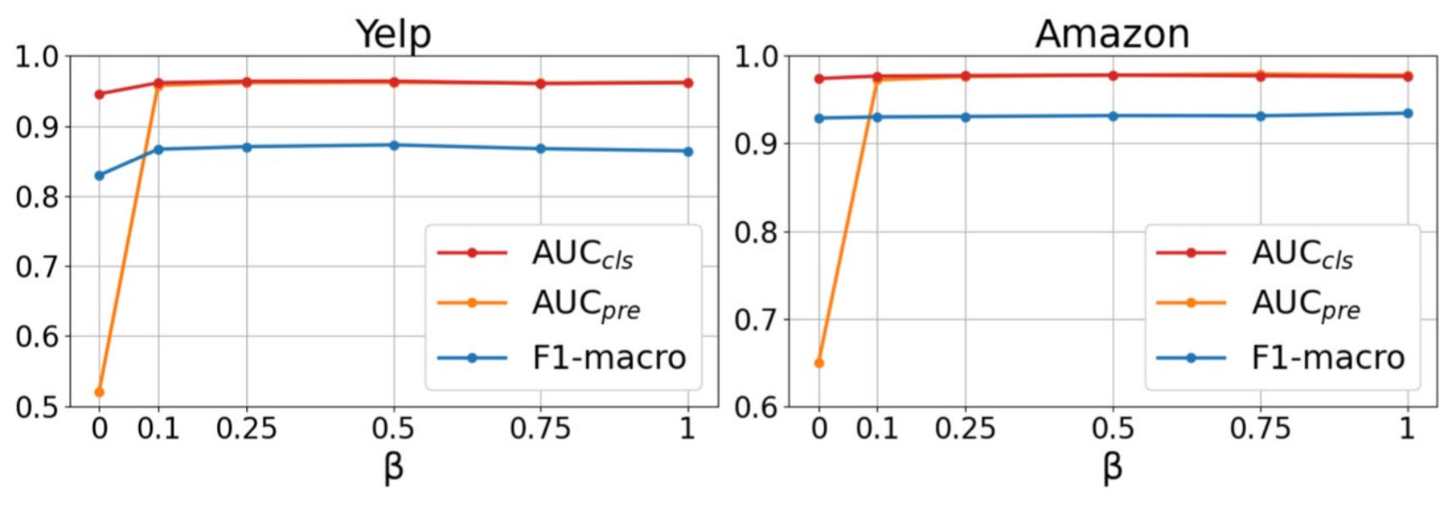


**Figure 2: Performance w.r.t $\beta$. When $\beta > 0$, label exploration is activated, which enhances overall model accuracy.**

Our code is available at https://github.com/wdhyun/LEX-GNN