Label-Aware Graph Attention Network for Enhanced Fraud Detection

Kaifeng Gao
Department of Statistics & Data Science
Yale University
New Haven, CT 06510
kaifeng.gao@yale.edu

1 Introduction

In the era of rapid digital expansion, fraudulent activities have become increasingly prevalent across various online platforms, including deceptive reviews, fake accounts, and malicious websites. Fraud detection has earned more and more attention in various fields including social networking, online payment, e-commerce, marketing etc.

Entity classification stands as one of the most critical tasks in fraud detection. The past few years have witnessed a significant increase in the utilization of graph-based techniques for identifying fraudulent activities. This approach involves representing entities as nodes, often accompanied by attribute information, while their interactions are depicted as edges within the graph structure.

Graph Neural Networks (GNNs) have become a powerful tool in entity level fraud detection, offering advantages over traditional methods through their ability to aggregate neighborhood information and learn node representations. This allows for semi-supervised learning with reduced need for feature engineering. However, fraudsters' camouflage tactics discussed in Dou et al. [2020b], Kaghazgaran et al. [2019] introduce noise into graph structures, challenging GNN effectiveness [Chen et al., 2019]. This camouflage often leads to low homophily, which impairs GNN performance. Ongoing research focuses on enhancing GNN architectures to better handle these challenges while maintaining end-to-end learning, aiming to improve robustness and accuracy in real-world fraud detection applications.

This project proposes an enhanced Graph Attention Network (GAT) Veličković et al. [2018] architecture that explicitly incorporates neighborhood label information to address the challenges posed by low homophily in fraud detection scenarios. Our method extends the traditional attention mechanism by computing attention scores based on a combination of node features, edge attributes, and known label distributions in local neighborhoods. Through comprehensive experiments on two real-world fraud detection datasets, we demonstrate that our approach outperforms standard GAT and other baseline methods. Specifically, our model achieves improvements of 3.03% and 2.97% in Average Precision score on datasets Amazon [Zhang et al., 2020] and Yelp [Weise, 2011], respectively. These results establish a promising foundation for future research in developing GNN architectures that maintain the advantages of graph structural learning while being specifically tailored to the unique challenges of fraud detection applications.

2 Related Works

Node-level Classification: Semi-supervised node classification aims to learn from labeled data and predict the properties of unlabeled nodes in a graph. Graph Attention Networks (GAT) [Veličković et al., 2018] introduce attention mechanisms that allow nodes to selectively attend to their neighbors' features. DeepGCN [Li et al., 2019] incorporates residual connections and message normalization layers to enable deeper Graph Convolutional Networks. GraphSAGE [Hamilton et al., 2018], Cluster-

Preprint. Under review.

GCN [Chiang et al., 2019], and GraphSAINT [Zeng et al., 2020] propose methods for constructing minibatches to scale up GNN representation learning.

GNN-based Fraud Detection: Several approaches have been developed to model relationships between nodes in fraud detection tasks. CARE-GNN [Dou et al., 2020a] addresses the camouflage problem in fraud graphs with a neighbor selection module. H2-FDetector [Shi et al., 2022] designs separate aggregation strategies for different connection types. FRAUDRE [Zhang et al., 2021] introduces a fraud-aware graph convolution module and an imbalance-oriented optimization objective.

Transformer-based Graph Fraud Detection: Researchers have leveraged Transformer models' sequence modeling capabilities for graph fraud detection. Graph Transformer [Cai and Lam, 2020] treats graphs as node sequences and uses GRU to encode node relationships. TADDY [Liu et al., 2023] proposes a Transformer-based detector for dynamic graphs with additional spatial and temporal encodings. GAGA [Wang et al., 2023] introduces group aggregation and additional encodings to incorporate structural and labeling information. RAGFormer [Li et al., 2024] addresses GAGA's limitations in capturing graph structure by combining it with a parallel GCN module and an attention fusion module.

Our work shares similarities with GAGA [Wang et al., 2023] in utilizing class labels to distinguish neighborhood information, which is beneficial in low homophily scenarios common in fraud detection. However, we address the potential loss of spatial information in GAGA's group aggregation, as noted by RAGFormer [Li et al., 2024]. Our approach aims to strike a balance between effective spatial information capture and model simplicity, differentiating it from the more complex multi-model approach of RAGFormer.

3 Methodology

We propose a Label-Aware Graph Attention Network (LAGAT) for enhanced fraud detection. Our approach extends GAT based on two key insights: (1) utilizing class labels to distinguish neighborhood information as proposed by Wang et al. [2023], which is particularly valuable in fraud detection scenarios where low homophily can impair GNN performance, and (2) preserving important spatial information that may be lost in group aggregation approaches, as highlighted by Li et al. [2024].

3.1 Background: Graph Attention Networks

The original GAT calculates attention coefficients as:

$$\begin{aligned}\alpha_{ij} &= \frac{\exp\left(\mathrm{LeakyReLU}\left(\mathbf{a}^T[\mathbf{\Theta}_t\mathbf{h}_i\|\mathbf{\Theta}_s\mathbf{h}_j\|\mathbf{\Theta}_e\mathbf{e}_{i,j}]\right)\right)}{\sum_{k\in\mathcal{N}(i)\cup\{i\}}\exp\left(\mathrm{LeakyReLU}\left(\mathbf{a}^T[\mathbf{\Theta}_t\mathbf{h}_i\|\mathbf{\Theta}_s\mathbf{h}_k\|\mathbf{\Theta}_e\mathbf{e}_{i,k}]]\right)\right)} && (1)\\ &= \frac{\exp\left(\mathrm{LeakyReLU}\left(\mathbf{a}_s^\top\mathbf{\Theta}_s\mathbf{x}_i+\mathbf{a}_t^\top\mathbf{\Theta}_t\mathbf{x}_j+\mathbf{a}_e^\top\mathbf{\Theta}_e\mathbf{e}_{i,j}\right)\right)}{\sum_{k\in\mathcal{N}(i)\cup\{i\}}\exp\left(\mathrm{LeakyReLU}\left(\mathbf{a}_s^\top\mathbf{\Theta}_s\mathbf{x}_i+\mathbf{a}_t^\top\mathbf{\Theta}_t\mathbf{x}_k+\mathbf{a}_e^\top\mathbf{\Theta}_e\mathbf{e}_{i,k}\right)\right)} && (2)\end{aligned}$$

3.2 Label-Aware Attention Networks

LAGAT extends the GAT architecture by incorporating label information into the attention mechanism while maintaining the ability to handle partially labeled graphs. For each target node, we categorize its neighbors into three distinct classes (This can be extended to multi-class):

- Benign nodes $(+)$: Confirmed legitimate transactions
- Fraudulent nodes $(-)$: Confirmed fraudulent transactions
- Unknown nodes $(*)$: Unlabeled or pending transactions

The enhanced attention coefficient $\alpha_{i,j}$ between a center node i and its neighbor j is computed as:

$$\alpha_{i,j} = \frac{\exp\left(\mathrm{LeakyReLU}\left(\mathbf{a}_s^\top\mathbf{\Theta}_s\mathbf{x}_i+\mathbf{a}_t^\top\mathbf{\Theta}_t\mathbf{x}_j+\mathbf{a}_e^\top\mathbf{\Theta}_e\mathbf{e}_{i,j}+\mathbf{a}_l^\top\mathbf{\Theta}_l\mathbf{l}_{i,j}\right)\right)}{\sum_{k\in\mathcal{N}(i)\cup\{i\}}\exp\left(\mathrm{LeakyReLU}\left(\mathbf{a}_s^\top\mathbf{\Theta}_s\mathbf{x}_i+\mathbf{a}_t^\top\mathbf{\Theta}_t\mathbf{x}_k+\mathbf{a}_e^\top\mathbf{\Theta}_e\mathbf{e}_{i,k}+\mathbf{a}_l^\top\mathbf{\Theta}_l\mathbf{l}_{i,k}\right)\right)} \qquad (3)$$

where:

- $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^d$ are node feature vectors
- $\mathbf{e}_{i,j} \in \mathbb{R}^{d_e}$ represents edge features (if applicable)
- $\mathbf{l}_{i,j} \in \mathbb{R}^{d_l}$ is the learnable label embedding
- $\mathbf{\Theta}_s, \mathbf{\Theta}_t \in \mathbb{R}^{d' \times d}$, $\mathbf{\Theta}_e \in \mathbb{R}^{d' \times d_e}$, $\mathbf{\Theta}_l \in \mathbb{R}^{d' \times d_l}$ are trainable transformation matrices
- $\mathbf{a}_s, \mathbf{a}_t, \mathbf{a}_e, \mathbf{a}_l \in \mathbb{R}^{d'}$ are learnable attention vectors

Additionally, to prevent label leakage during training and maintain model generalization, we implement the following safeguards:

- Self-connections are always treated as unknown labels
- Attention weights for unknown labels provide a fallback mechanism for unlabeled nodes

4 Experiments

4.1 Dataset

We conduct our experiments on Amazon and Yelp Fraud dataset. Table 1 presents the detailed statistics of the dataset.

4.1.1 Amazon

The Amazon dataset consists of product reviews in the Musical Instruments category. Users with more than 80% helpful votes are labeled as benign, while those with less than 20% are considered fraudulent. The dataset features 25 handcrafted node attributes as detailed by Zhang et al. [2020]. Users are represented as nodes with three types of connections:

1. **U-P-U**: Users who have reviewed at least one common product.
2. **U-S-U**: Users who have given at least one identical star rating within one week.
3. **U-V-U**: Users with mutual review text similarities in the top 5%, measured by TF-IDF.

4.1.2 Yelp

The Yelp dataset includes hotel and restaurant reviews from the Yelp platform. Yelp's filtering algorithm identifies potentially fake or suspicious reviews, classifying them as either recommended or filtered. Although not absolute, Yelp's anti-fraud filter is highly regarded and used as a benchmark in research [Weise, 2011]. The dataset includes 32 handcrafted node features as outlined in Rayana and Akoglu [2015]. Reviews are represented as nodes with three connection types:

1. **R-U-R**: Reviews posted by the same user.
2. **R-S-R**: Reviews for the same product with the same star rating.
3. **R-T-R**: Reviews for the same product posted in the same month.

Table 1: Dataset and graph statistics.

	#Nodes (Fraud%)	Relation	#Edges	Max Degree
Amazon	11,944 (6.8%)	U-P-U	175,608	511
		U-S-U	3,566,479	6311
		U-V-U	1,036,737	6477
		ALL	4,398,392	
Yelp	45,954 (14.5%)	R-U-R	49,315	46
		R-T-R	573,616	118
		R-S-R	3,402,743	465
		ALL	3,846,979	

4.2 Experimental Setup

4.2.1 Implementation Details

Our model is implemented in PyTorch Geometric [Fey and Lenssen, 2019]. The code is available at `https://github.com/Kaifeng-Gao/LAGAT`.

All experiments were conducted on a system equipped with a 13th Gen Intel(R) Core(TM) i7-13700K processor, 16GB RAM, and an NVIDIA GeForce RTX 4070 GPU with 12GB VRAM, running Ubuntu 22.04.3 LTS. The implementation uses PyTorch with CUDA 12.6 and NVIDIA driver version 560.94.

For the model architecture of GAT and LAGAT, we use 2 layers with 32 hidden channels, 2 attention heads per layer, and a dropout rate of 0.6. The label embedding dimension (only for LAGAT) is set to 8. The model is trained using the Adam optimizer with a learning rate of 0.005 and weight decay of 5e-4. We train for a maximum of 10,000 epochs with early stopping (patience of 300 epochs) based on validation average precision. For evaluation robustness, we report results averaged over 3 different random seeds (42, 43, 44) with a 60%/20%/20% train/validation/test split.

4.2.2 Data Preprocessing

For data loading and preprocessing, we implement a custom pipeline that handles heterogeneous fraud detection datasets. The data loader converts DGL graph formats to PyTorch Geometric's format using the `from_dgl` utility. A key preprocessing step is the label masking procedure, which simulates partial label observability - a common scenario in real-world fraud detection where ground truth labels are often limited.

Specifically, we implement a `mask_label` function that creates a controlled partially-labeled setting. For nodes in the validation and test sets, all labels are masked. For the training set, we randomly mask 40% of the labels, resulting in only 60% of training labels being observable. Labels are encoded using a three-value system (0,1,2), where 0 represents masked/unknown labels, and 1,2 represent the binary fraud classification. This masking strategy helps evaluate the model's capability to learn from partially labeled data while maintaining separate validation and test sets for proper evaluation.

4.3 Results

Table 2 presents the comparative results between GAT and LAGAT on Amazon and Yelp benchmark datasets. As mentioned in implementation details, all experiments are repeated three times with different random seeds, and we report the mean and standard deviation.

Table 2: Comparison between GAT and LAGAT

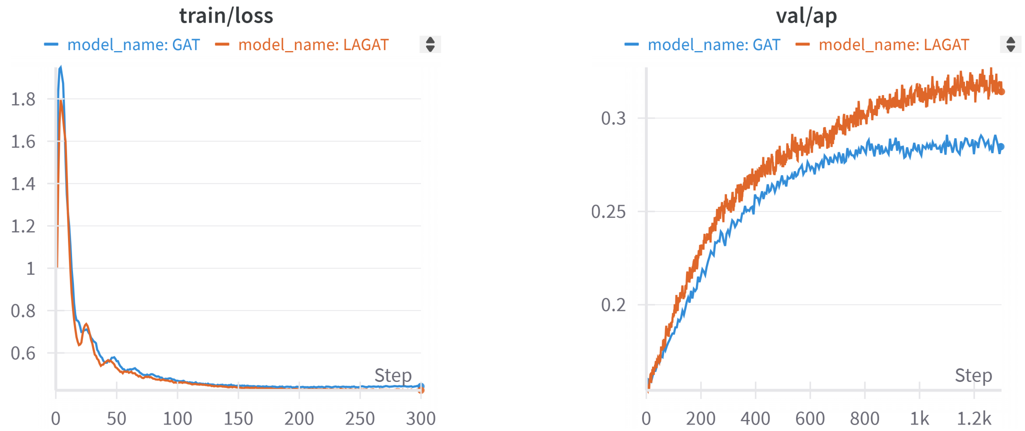
	Models	Test AP (%)	Test AUC (%)	Test F1 (%)
Amazon	GAT	11.56 ± 2.36	**55.23 ± 4.58**	44.28 ± 16.15
	LAGAT	**14.59 ± 3.54**	55.02 ± 3.51	**55.32 ± 3.69**
Yelp	GAT	28.59 ± 2.06	70.90 ± 3.28	46.12 ± 0.09
	LAGAT	**31.56 ± 1.38**	**73.45 ± 0.33**	**46.25 ± 0.21**

LAGAT consistently outperforms GAT across most metrics. On the Amazon dataset, LAGAT achieves a 3.03% improvement in AP and a substantial 11.04% gain in F1 score, while maintaining comparable AUC performance. For the Yelp dataset, LAGAT demonstrates superior performance across all metrics, with notable improvements of 2.97% in AP, 2.55% in AUC, and 0.13% in F1 score. These results demonstrate that incorporating label-aware attention mechanisms effectively enhances fraud detection performance, particularly in terms of precision and recall metrics.

4.4 Analysis

4.4.1 Learning Dynamics

Figure 1 shows the training and validation curves over epochs, demonstrating the learning behavior of our model. Curves are smoothed with time-weighted EMA (factor 0.3) and averaged across different trials.



(a) Train Loss by Model

(b) Validation AP by Model

Figure 1: Learning Dynamics for LAGAT and GAT (Yelp Dataset)

From the training loss curves (Figure 1a), we observe that both models exhibit similar convergence rates. However, the validation AP curves (Figure 1b) reveal that LAGAT achieves consistently better performance throughout training. This suggests that while the label-aware attention mechanism doesn't affect convergence speed, it enables more effective feature learning that translates to improved model performance.

4.5 Hyperparameter Analysis

We conduct extensive experiments to analyze the sensitivity of LAGAT to two key hyperparameters: the label embedding dimension and the ratio of observed labels in the training set. All experiments are performed on the Yelp dataset with results averaged across three runs.

Table 3: Influence of Label Embedding Dimension

Dataset	Label Embedding Dim	Test AP (%)	Test AUC (%)	Test F1 (%)
Yelp	4	31.35 ± 1.65	73.13 ± 0.78	46.07 ± 0.05
	8	**31.56 ± 1.38**	**73.45 ± 0.33**	**46.25 ± 0.21**
	16	29.92 ± 0.68	73.10 ± 0.61	46.07 ± 0.05
	32	30.67 ± 0.22	73.37 ± 0.71	46.10 ± 0.08

As shown in Table 3, a label embedding dimension of 8 achieves optimal performance across all metrics. Smaller dimensions may not capture sufficient label information, while larger dimensions potentially lead to overfitting. This suggests that a moderate embedding size strikes the right balance between expressiveness and model complexity.

Table 4 presents model performance under different ratios of observed training labels. Interestingly, the model achieves peak performance with 80% observed labels rather than full label observation. This phenomenon suggests that partial label masking during training may act as a form of regularization, helping the model better generalize to nodes with unknown labels at test time. The degraded performance at 100% observation could indicate that the model becomes overly dependent on label information and loses some ability to leverage structural features of the graph.

Table 4: Influence of Label Observation Ratio

Dataset	Observed Ratio	Test AP (%)	Test AUC (%)	Test F1 (%)
Yelp	0%	29.75 ± 1.06	73.48 ± 1.35	46.10 ± 0.03
	20%	30.39 ± 1.42	**73.65 ± 0.33**	46.10 ± 0.10
	40%	29.57 ± 0.42	72.59 ± 0.74	46.15 ± 0.02
	60%	28.44 ± 1.78	70.97 ± 3.96	46.10 ± 0.03
	80%	**30.79 ± 1.49**	72.66 ± 0.62	**46.23 ± 0.22**
	100%	28.90 ± 1.43	71.29 ± 1.98	46.07 ± 0.06

5 Conclusion

In this work, we presented LAGAT, a label-aware graph attention network designed specifically for fraud detection in low-homophily settings. By incorporating neighborhood label information into the attention mechanism, our model effectively captures both structural and label-based patterns in the data. The experimental results on two real-world fraud detection datasets demonstrate that LAGAT consistently outperforms traditional GAT, achieving significant improvements in detection accuracy with gains of 3.03% and 2.97% in Average Precision on Amazon and Yelp datasets respectively.

However, our approach has several limitations that warrant further investigation. A primary constraint is that LAGAT is currently limited to transductive learning scenarios where partial labels are available in the graph. In scenarios without any known labels, the model essentially reduces to a standard GAT, limiting its applicability in fully inductive settings. Additionally, while our results show consistent improvements across multiple runs, the performance gains, though promising, would benefit from validation across a broader range of experimental settings and datasets to establish more robust statistical significance.

Looking forward, several promising directions could extend this work. One potential avenue is enhancing model interpretability through the analysis of cumulative attention distributions. By examining how the model weighs information from nodes with different label states, we could gain valuable insights into its decision-making process and potentially identify key patterns in fraud detection. Another interesting direction would be exploring alternative mechanisms for label information integration and extending the attention mechanism to capture multi-hop neighborhood information. These improvements could further enhance the model's effectiveness in real-world fraud detection applications.

References

D. Cai and W. Lam. Graph transformer for graph-to-sequence learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7464–7471, Apr. 2020. doi: 10.1609/aaai.v34i05.6243. URL `https://ojs.aaai.org/index.php/AAAI/article/view/6243`.

D. Chen, Y. Lin, W. Li, P. Li, J. Zhou, and X. Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view, 2019. URL `https://arxiv.org/abs/1909.03211`.

W.-L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C.-J. Hsieh. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '19. ACM, July 2019. doi: 10.1145/3292500.3330925. URL `http://dx.doi.org/10.1145/3292500.3330925`.

Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 315–324, New York, NY, USA, 2020a. Association for Computing Machinery. ISBN 9781450368599. doi: 10.1145/3340531.3411903. URL `https://doi.org/10.1145/3340531.3411903`.

Y. Dou, G. Ma, P. S. Yu, and S. Xie. Robust spammer detection by nash reinforcement learning. *CoRR*, abs/2006.06069, 2020b. URL `https://arxiv.org/abs/2006.06069`.

M. Fey and J. E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

W. L. Hamilton, R. Ying, and J. Leskovec. Inductive representation learning on large graphs, 2018. URL `https://arxiv.org/abs/1706.02216`.

P. Kaghazgaran, M. Alfifi, and J. Caverlee. Wide-ranging review manipulation attacks: Model, empirical study, and countermeasures. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 981–990, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450369763. doi: 10.1145/3357384.3358034. URL `https://doi.org/10.1145/3357384.3358034`.

G. Li, M. Müller, A. Thabet, and B. Ghanem. Deepgcns: Can gcns go as deep as cnns?, 2019. URL `https://arxiv.org/abs/1904.03751`.

H. Li, S. Jiang, L. Zhang, S. Du, G. Ye, and H. Chai. Ragformer: Learning semantic attributes and topological structure for fraud detection, 2024. URL `https://arxiv.org/abs/2402.17472`.

Y. Liu, S. Pan, Y. G. Wang, F. Xiong, L. Wang, Q. Chen, and V. C. Lee. Anomaly detection in dynamic graphs via transformer. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12081–12094, Dec. 2023. ISSN 2326-3865. doi: 10.1109/tkde.2021.3124061. URL `http://dx.doi.org/10.1109/TKDE.2021.3124061`.

S. Rayana and L. Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 985–994, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336642. doi: 10.1145/2783258.2783370. URL `https://doi.org/10.1145/2783258.2783370`.

F. Shi, Y. Cao, Y. Shang, Y. Zhou, C. Zhou, and J. Wu. H2-fdetector: A gnn-based fraud detector with homophilic and heterophilic connections. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 1486–1494, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450390965. doi: 10.1145/3485447.3512195. URL `https://doi.org/10.1145/3485447.3512195`.

P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks, 2018. URL `https://arxiv.org/abs/1710.10903`.

Y. Wang, J. Zhang, Z. Huang, W. Li, S. Feng, Z. Ma, Y. Sun, D. Yu, F. Dong, J. Jin, B. Wang, and J. Luo. Label information enhanced fraud detection against low homophily in graphs. In *Proceedings of the ACM Web Conference 2023*, WWW '23. ACM, Apr. 2023. doi: 10.1145/3543507.3583373. URL `http://dx.doi.org/10.1145/3543507.3583373`.

K. Weise. A lie detector test for online reviewers. `http://bloom.bg/1KAxzhK`, 2011. Accessed: [Insert access date here].

H. Zeng, H. Zhou, A. Srivastava, R. Kannan, and V. Prasanna. Graphsaint: Graph sampling based inductive learning method, 2020. URL `https://arxiv.org/abs/1907.04931`.

G. Zhang, J. Wu, J. Yang, A. Beheshti, S. Xue, C. Zhou, and Q. Z. Sheng. Fraudre: Fraud detection dual-resistant to graph inconsistency and imbalance. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 867–876, 2021. doi: 10.1109/ICDM51629.2021.00098.

S. Zhang, H. Yin, T. Chen, Q. V. N. Hung, Z. Huang, and L. Cui. Gcn-based user representation learning for unifying robust recommendation and fraudster detection. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 689–698, 2020.