A Novel Parallel Graph Computing Model for Unsupervised Fraud Detection

Jiahui Wang[1], Rui Sun[1], Fangshu Chen (✉)[1], Wei Zhang[1], Panpan Feng[2], and Chengcheng Yu[1]

[1] Shanghai Polytechnic University, Shanghai, China
{wangjh,20231513016,fschen,20211510135, ccyu}@sspu.edu.cn
[2] iQIYI Inc, Shanghai, China
fengpanpan@qiyi.com

Abstract. Graph computing models show great popularity in tracking the fraud detection problem defending merchant losses with illegal behaviors. However, current solutions mainly focus on capturing features of user historical behavior sequences with supervised learning, which makes it difficult to cope with the vagaries of fraud patterns. To gain a lot of traction, this paper proposes a novel PARallel Graph Computing model for Unsupervised Fraud Detection (PAR-GCUFD). The unsupervised graph construction and graph contrastive learning are developed to effectively identify unseen fraud patterns. Furthermore, we innovatively design a parallel strategy to accelerate the speed of training and inference. Extensive experiments on real-world industrial datasets also prove the superiority of the proposed PAR-GCUFD framework.

Keywords: Fraud Detection · Parallel Computing · Graph Neural Network · Unsupervised Learning.

1 Introduction

With the explosive growth of Internet users, platforms inevitably face more and more complex security challenges. Massive fraudulent behaviors, such as malicious batch account registration and fake reviews, seriously threaten the security, reputation, and economic efficiency of online platforms. Therefore, it is urgent to effectively assess the risk level of users and detect fraudulent users in real time, which can help to identify and prevent potential security threats, protect users' rights and interests, and maintain social public order. To deal with this issue, a myriad of feature-based and graph-based fraud detection models achieve eye-catching performance. Generally, these methods rely on large amounts of annotated data to learn potential characteristics from the historical behavior of fraudulent users and then make predictions on the unknown. This inevitably makes it difficult for models to cope with the vagaries of fraud patterns, especially those that have never been seen before. However, unsupervised fraud detection has not received sufficient attention in the literature. Besides, graph-based fraud detection models have become the mainstream paradigm due to their powerful representation capability. Widely recognized, the computational complexity

of graph models is significantly higher than that of traditional models. However, the task of fraud detection has a high requirement for timeliness, which is a significant contradiction. Therefore, this paper is willing to propose a novel PARallel Graph Computing model for Unsupervised Fraud Detection (PAR-GCUFD). PAR-GCUFD, a novel unsupervised approach, can effectively identify unseen fraud patterns. Furthermore, the method innovatively designs a parallel strategy to construct graph structures and unsupervised strategies, which can effectively accelerate the speed of training and inference. Consequently, PAR-GCUFD breaks the defect of the traditional model in unsupervised prediction and parallel acceleration. The primary contributions of this paper are outlined as follows:

1. An unsupervised graph fraud detection method is delivered based on the user-event bipartite graph, which fully considers the outliers of fraudulent user behavior events.
2. A parallel computing strategy is introduced to accelerate the efficiency in both the pre-graph topology construction and post-graph prediction.
3. Extensive experiments are conducted on real-world industrial datasets to show the superiority of our proposal.

2 Related Work

Along with the explosive growth of online information, the frequency of Internet risk activities has also increased significantly, which has caused huge reputational and economic losses. This has attracted many research teams to actively invest in the field of user risk prediction.

2.1 Feature-based Fraud Detection Models

Conventional feature-based fraud detection models are mainly based on the user's historical behavior data, such as IP address, device type, behavior timestamp, etc. using machine algorithms for risk assessment and prediction such as support vector [20, 13], logistic regression [21] and decision tree [4]. Xie [22] proposed the spam payload and spam server traffic attributes to generate high-quality regular expressions for spam identification. Cao[1] proposed an unsupervised machine learning framework to promote the use of aggregated behavior patterns to better identify fraudulent users. Taha et al. [18] proposed an optimized LightGBM algorithm for detecting credit card transaction fraud and adjusted the parameters of LightGBM through a Bayesian-based hyperparameter optimization algorithm, thereby improving the performance of the model.

However, fraudulent users disguise themselves so that their features are similar to those of normal users, which makes it difficult to distinguish them based on features alone.

2.2 Graph-based Fraud Detection Models

Graph data, the non-Euclidean form of data organization, can not only express and store massive data information in graph structure but also make the relationship between data more intuitive and easy to understand. Graph neural network (GNN) is the mainstream paradigm for processing this kind of data [11, 17, 14, 6]. Liu[7] introduced user motivation when identifying user behavior. In the construction of graph data for fraud detection, by modeling user motivation, the graph neural network can not only detect fraudsters but also give a good explanation. Li[5] proposed a new node aggregation method for heterogeneous information networks. It proposed aggregators for user nodes, comment nodes, and commodity nodes, aggregated the information of their respective neighbors, and learned three different expressions. Finally, an isomorphic comment graph was constructed by the similarity between comments. Graph Neural Networks [23, 9, 12] can effectively capture the graph structure and complex relationships between nodes through information transmission and aggregation.

2.3 Parallel Computing

When processing the massive data, the traditional calculation method may encounter the problem of low efficiency. In response to this challenge, the use of parallel computing technology can accelerate the data processing process and improve the training and prediction efficiency of prediction models [24, 3].

Parallel computing divides computing tasks into multiple sub-tasks and executes them simultaneously on multiple processing units to achieve the purpose of accelerating computing. User risk prediction based on a knowledge graph needs to construct and analyze the graph structure of massive data. Parallel computing technology can accelerate the construction process of the knowledge graph. By dividing the construction process of the graph into multiple sub-tasks and executing them in parallel on different computing nodes, the efficiency and speed of graph composition can be improved [15]. At the same time, for the graph algorithm applications of the knowledge graph, such as label propagation, community discovery, and graph neural network, the parallel computing technology can accelerate the calculation process of the algorithm and improve the training and prediction speed of the prediction model [16, 2]. Ensemble learning plays an important role in user risk prediction. However, the training and optimization of ensemble learning models also require a lot of computing resources. Through parallel computing technology, the training process of the ensemble learning model can be accelerated, and the performance and robustness of the model can be improved [19, 8]. For example, using the distributed computing framework and parallel algorithm design, the training process of the ensemble learning model can be parallelized, making full use of the computing resources of multiple computing nodes, accelerating the training process of the model and improving the accuracy and generalization ability of the model. Therefore, the application of parallel computing technology can accelerate the training and prediction process of user risk prediction models based on knowledge graphs and improve the efficiency and performance of prediction models.

3 Methodology

3.1 Graph Construction Framework

First, the user behavior log must be converted into a graph structure, providing the data foundation for subsequent fraud detection. PAR-GCUFD designs the unsupervised graph construction procedure with full consideration of the actual industrial scenario, as shown in Fig 1.

Initially, users and events are abstracted into two types of node entities in a heterogeneous bipartite graph, which can be easily extracted from user behavior logs. Additionally, we design an unsupervised node weight initialization strategy based on outliers in behavior events. Afterward, PAR-GCUFD converts heterogeneous graphs into homologous graphs based on topological connectivity and edge weights, which are then fed into the graph model for detection.

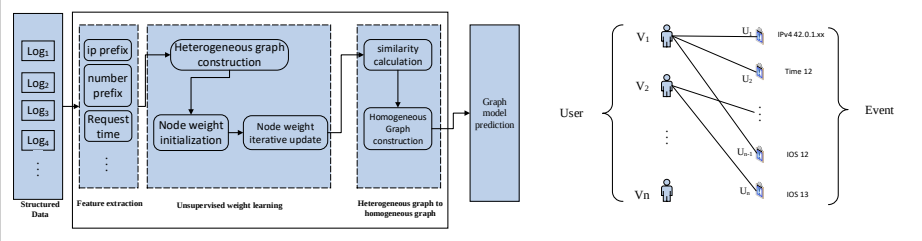

Fig. 1. User behavior event graph construction framework.

3.2 Weight Initialization and Updating

The heterogeneous graph is defined as H=(V, U, E), where V and U represent user nodes and event nodes, respectively. Additionally, user nodes and event nodes are connected by the edge set E in the bipartite graph. Assuming an event occurs after 12 PM, the device system version is IOS 12, and the IPv4 prefix is 43.0.1.xxx, then the corresponding user node v1 and event nodes u1, u2, u3 are created in the heterogeneous graph. Node v1 is connected to these event nodes to represent their relationships. Next, we construct unsupervised weight initialization and updating strategies. Generally, the number of values for behavior events is finite and fixed. For example, the request time (hour) has 24 possible values, from 0 to 23. The ratio of an event is defined as the probability of its occurrence, as shown in Eq.(1). For example, if there are 1000 logins using IOS devices and 100 are IOS12, then the ratio(IOS12) is 0.1.

$$ratio\,(u) = \frac{freq\,(u)}{\sum_{u' \in pre(u)} freq\,(u')} \tag{1}$$

In addition, the event mode is defined as the maximum ratio in Eq.(2).

$$u' = mode\,(\,pre\,(\,u\,)\,) = \; arg\; max_{u' \in pre(\,u\,)}\; ratio\,(\,u'\,) \tag{2}$$

For example, if there are three event nodes: IOS 7, IOS 8, and IOS 9, with ratios of 0.25, 0.25, and 0.5, respectively, then the mode of the IOS version event is IOS 9. However, this definition is not perfect, as the difference in event ratios between nodes under different categories can be significant. In reality, the distribution of device system versions is often concentrated, as most users use newer versions. The distribution of mobile phone number prefixes may be relatively uniform, as phone numbers are scattered across different segments.

Therefore, it is necessary to reconsider how to define and compare the weights of different event nodes to ensure more meaningful comparisons. More detailed methods can consider the different characteristics of event distributions to more accurately reflect node abnormalities. To address this issue, [10] proposed an event coupling technique for categorical events, which enables the comparison of different event types. Based on EQ.(1), the following equations are obtained through optimization and redefinition as Equation 3-5.

$$w_u = \frac{1}{2}\left(dev\left(u\right) + base\left(u\right) \right) \tag{3}$$

$$dev(x) = \begin{cases} 1 - \frac{ratio(u)}{ratio(mode(pre(u)))} & if\ pre(u) \in Pre\_A \\ \frac{\text{ratio}(u)}{ratio(\text{mode}(pre(u)))} & if\ pre(u) \in Pre\_B \end{cases} \tag{4}$$

$$base(x) = \begin{cases} 1 - ratio\left(mode\big(pre(u)\big)\right) & if\ pre(u) \in Pre\_A \\ ratio\left(mode\big(pre(u)\big)\right) & if\ pre(u) \in Pre\_B \end{cases} \tag{5}$$

Where $dev(u)$ describes the anomaly of an event node u, and $base(u)$ describes the anomaly of the event class $pre(u)$.

3.3 Converting into Homogeneous Graph

Considering that the core focus of fraud detection is the user, PAR-GCUFD designs strategies to eliminate event entities from the user-event heterogeneous bipartite graph, transforming it into a graph containing only user entities while retaining crucial information. Specifically, the similarity between two users is calculated, and user nodes with similarity above a threshold are aligned. The similarity between users is defined in Equation 6-7, where $con(v)$ represents the set of event nodes connected to the user node v in the heterogeneous graph H. The formula sums the weights of event nodes shared by the node pair (v, v').

$$sim\left(v, v^{'}\right) = \sum\nolimits_{s \in con(v) \cap con(v')} w_s \tag{6}$$

$$x_v = [\, w_{u_1}, w_{u_2}, \ldots, w_{u_F} \mid u_1, u_2, \ldots u_F \in con\left(v\right)] \tag{7}$$

After calculating the similarity between node pairs using Eq (6), edges are evaluated based on the threshold, and if they meet the criteria, they are appended to E. Once all node pairs have been traversed, the transmission of topological structure information from the heterogeneous graph to the homogeneous graph is complete.

3.4 Parallel Accelerated Graph Construction Optimization

In transforming heterogeneous graphs into Homogeneous graphs, the time complexity for calculating similarity between node pairs at the topological level is $O(n^2)$. Thus, as the number of nodes increases, the computation time grows exponentially. To mitigate this, the paper proposes dividing event node subsets based on feature nodes and then calculating similarity within these subsets to construct the isomorphic graph.

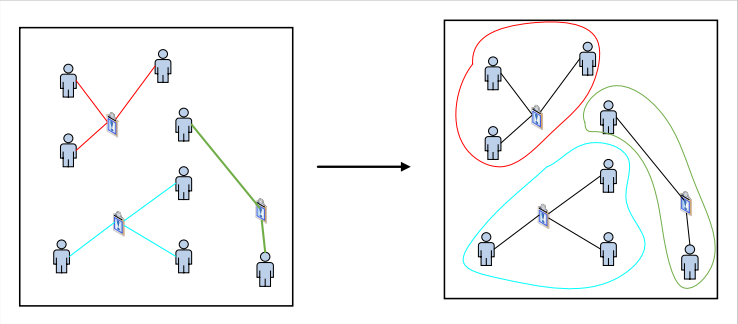

Fig. 2. Parallel accelerated graph construction.

The graph is divided into three subsets using colored dashed lines, and similarity is compared within these subsets. For example, there are 8 user nodes requiring 28 similarity calculations under normal conditions in Fig.2. However, only user nodes sharing common event nodes are likely to have potential relationships. This method incurs minimal time cost for dividing subsets but significantly reduces the overall time complexity of the graph construction process. In practical tasks, where converting heterogeneous graphs to isomorphic graphs often involves tens of thousands of nodes, this method can substantially save computing time and accelerate graph construction.

3.5 Graph Contrastive Learning for Fraud Detection

PAR-GCUFD is an unsupervised learning method based on graph contrastive learning, designed to enhance model performance by leveraging both topological and attribute information of homogeneous knowledge graphs. This method enhances data by altering graph structures and attributes, allowing the model to learn critical features effectively.

For an existing homogeneous knowledge graph G, where X and A represent the feature matrix and adjacency matrix, respectively, we define $I(G) = (X, A)$ to encapsulate the graph's information. A GNN encoder f_{GNN} processes this information: $f_{GNN}(X, A)$.Two random augmentation functions $t \sim T$and $t' \sim T$ are applied to the graph G, resulting in two new graph representations $G1 = t(G)$ and $G2 = t'(G)$. These graphs are processed by the same GNN encoder to yield new feature matrices $M = f_{GNN}\,(\,X_1, A_1\,)$ and $N = f_{GNN}\,(\,X_2, A_2\,)$, referred

to as views M and N. The goal of augmentation functions is to preserve important structures and node features while perturbing less critical edges and features.

To quantify the similarity between the same nodes in different views and the dissimilarity among different nodes, we define a contrastive objective. For any node i, let m_i and n_i be its representations in views M and N, respectively. The objective is to maximize the similarity between positive pairs (m_i,n_i) and minimize it for negative pairs (m_i,n_i) and(m_i,n_i):

$$\mathit{\Gamma}(u_i, m_i) = \log \frac{e^{\theta(m_i,n_i)/\tau}}{e^{\theta(m_i,n_i)/\tau} + \sum_{j\neq i} e^{\theta(m_i,n_j)/\tau} + \sum_{j\neq i} e^{\theta(m_i,m_j)/\tau}} \tag{8}$$

where τ is a temperature parameter, and $\theta(m, n) = s(g(m), g(n))$ with s as the cosine similarity function and g as a nonlinear projection. The overall loss function is:

$$Loss = -\frac{1}{2N}\sum_{i-1}^{N}[\ \mathit{\Gamma}\left(u_i, m_i\right) + \mathit{\Gamma}\left(m_i, u_i\right)\ ] \tag{9}$$

3.6 Parallel Accelerated Graph Fraud Detection Optimization

The basic concept of multi-GPU parallel training is to divide the entire graph into subgraphs and assign each to different GPUs for processing. Each GPU independently performs forward propagation, backpropagation, and parameter updates, periodically synchronizing model parameters to ensure consistency. The general steps of the multi-GPU parallel training process in DGL are shown in Fig 3.

Graph partitioning: First, the entire graph is divided into multiple subgraphs, each with some nodes and edges. Partitioning is usually based on node or edge features to make sure each subgraph has roughly the same size and complexity.

Subgraph assignment: Subgraphs are evenly distributed among GPUs, enabling each GPU to process one or more subgraphs. This enables each GPU to calculate independently and achieve parallel processing.

Parallel computing: Each GPU independently conducts forward and backpropagation calculations. During forward propagation, each node gets messages from its neighbors and updates its representation accordingly. In backpropagation, each node's gradient is calculated and sent back to its neighboring nodes. These computations are done in parallel on multiple GPUs.

Parameter synchronization: Model parameters are regularly synchronized among GPUs to ensure consistency during training. Typically, the All-reduce operation is used for parameter synchronization. This operation effectively accumulates gradients from each GPU and calculates their average to update model parameters.

Model Update: Model parameters on each GPU are updated with the synchronized gradient. This ensures each GPU has the latest model parameters for the next round of calculations. This process ensures model consistency in multi-GPU training and enables more data processing in less time, thus accelerating graph model training.

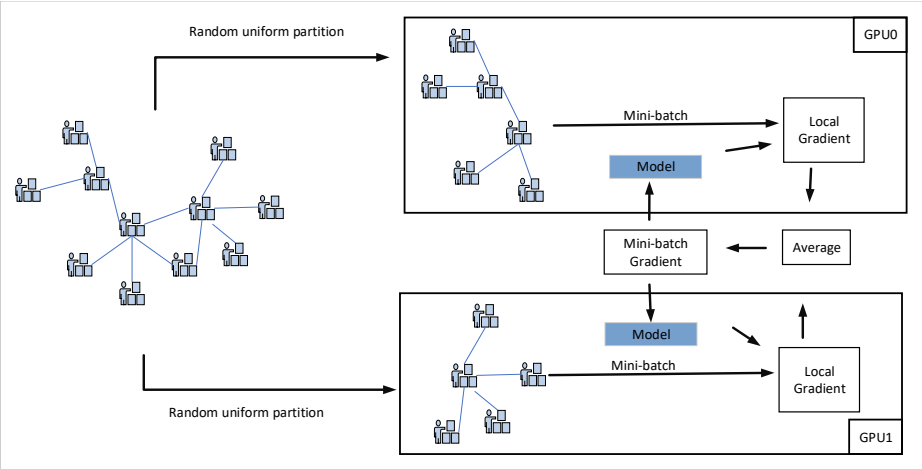

Fig. 3. Parallel multi-GPU training of graph model.

4 Experimental Analysis and Discussion

For evaluation, we testify the PARallel Graph Computing model for Unsupervised Fraud Detection (PAR-GCUFD) on the real-world industrial datasets provided by our cooperative company, Beijing iQiyi Technology Co., LTD, the video industry leader in China. The raw data includes 49.13 million access log data for login and registration scenarios between March 2021 and February 2022. All experiments were performed on Intel(R) Xeon(R) Gold 6248R, 512GB of RAM, 2000GB hard disk, and 6 Nvidia-A100 graphics cards.

4.1 Effectiveness of Fraud Detection

Since PAR-GCUFD is a unified graph-based fraud detection model, the prediction phase can utilize arbitrary graph neural networks. Table 1 testifies the effectiveness of fraud detection of PAR-GCUFD and other mainstream graph models. The experiments reveal that PAR-GCUFD achieves relatively higher performance than other frontier methods, which can effectively detect fraudulent users.

Table 1. Performance of different Methods

Algorithm	Accuracy	Precision	Recall	F1 Score
GCN	0.9075	0.8311	0.6457	0.7267
GAT	0.9300	0.7946	0.8530	0.8228
GraphSAGE	0.9400	0.8462	0.8373	0.8417
SGC	0.8915	0.8694	0.5066	0.6401
GIN	0.9400	0.8271	0.8661	0.8462
APPNP	0.9330	0.8034	0.8583	0.8299
GatedGraph	0.8915	0.8694	0.5066	0.6401
PAR-GCUFD	**0.9445**	**0.8462**	**0.8661**	**0.8560**

Furthermore, Fig 4 compares the Par-GCUFD with the classical GCN and GraphSAGE models under different data scales and imbalances. It is indicated that the Par-GCUFD algorithm maintains a high level of prediction precision at any number of nodes, reaching more than 90%, which is 15%-20% ahead of GCN and GraphSAGE, and only 10% of the samples are fraudulent users. Also, the average Recall of Par-GCUFD is still 10%-20% better than that of GCN and GraphSAGE.

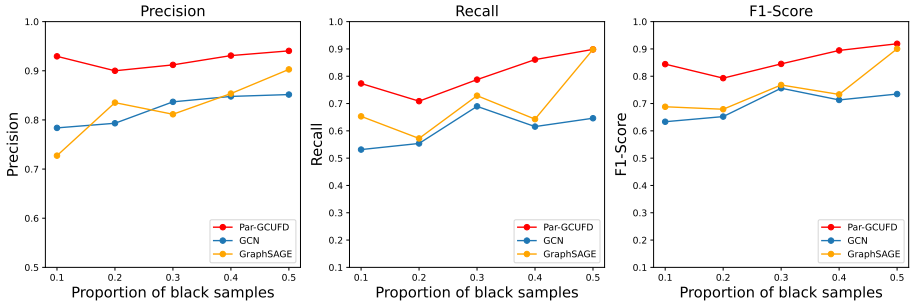

Fig. 4. Evaluation of each algorithm with different proportions of black samples.

4.2 Parallel Acceleration of Graph Construction

Fig 5 compares the Traversal Graph Construction (TGC), GCUFD, and the parallel version Par-GCUFD under different training scales. It can be seen that Par-GCUFD has the best performance under all the given data amounts, followed by GCUFD, and Par-GCUFD consumes the longest time. When the amount of data is 40000, the composition time of TGC will increase significantly, reaching 10^4 orders of magnitude. However, GCUFD is still within the acceptable time range of nearly 1000.

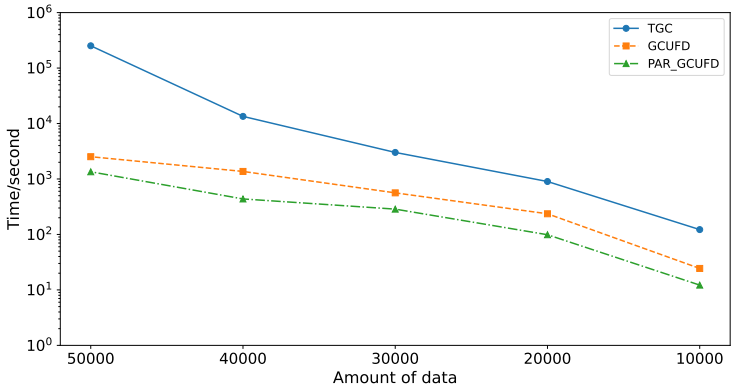

Fig. 5. Parallel acceleration of graph construction.

4.3 Parallel Acceleration of Fraud Detection

To explore the acceleration effect of our proposal, Table 1 illustrates the performance of the original version of GCN, GraphSAGE, and GCUFD. The experiments are implemented on three Tesla A100 for training with 512 batch size. The multi-GPU parallel accelerated version of PAR-GCN can reduce training time by 25%-52% compared to the original GCN. Similarly, Par-GCUFD training time was reduced by 31% to 53%. In summary, using the DGL framework for parallel training can significantly reduce the training time of graph algorithms and increase the timeliness and feasibility of the model in practical projects.

Table 2. Performance of different methods at different number of points

Method/points	10000	20000	30000	40000	50000
GraphSAGE	60.847	252.431	518.904	1010.539	2450.558
GCN	131.289	531.239	991.447	1795.806	4221.938
GCUFD	115.394	479.234	754.656	1375.165	3254.164
Par-GraphSAGE	34.032	180.284	342.741	723.483	1792.828
Par-GCN	62.364	321.575	712.813	1262.145	3132.192
Par-GCUFD	53.634	271.053	468.327	943.541	2035.480

5 Conclusion

This paper presents a novel PARallel Graph Computing model for Unsupervised Fraud Detection (PAR-GCUFD). As an unsupervised learning model, the recognition efficiency of unknown crime patterns has been significantly enhanced. Meanwhile, the parallel graph computing framework is introduced to accelerate the two stages of graph construction and graph prediction simultaneously. The comprehensive experiments demonstrate that PAR-GCUFD can reduce the training time consumption by 31% to 53% under the premise of maintaining high fraud detection performance.

Acknowledgments. This work is supported by the Natural Science Foundation of Shanghai(Grant No. 24ZR1425500), Shanghai University Young Teacher Training funding Program (Grant No. ZZEGD202414).

References

1. Cao, Q., Yang, X., Yu, J., Palow, C.: Uncovering large groups of active malicious accounts in online social networks. In: Proceedings of ACM SIGSAC Conference on Computer and Communications Security. pp. 477–488 (2014)

2. Deng, L., Wu, C., Lian, D., Wu, Y., Chen, E.: Markov-driven graph convolutional networks for social spammer detection. Institute of Electrical and Electronics Engineers Transactions on Knowledge and Data Engineering **35**(12), 12310–12322 (2023)
3. Han, Y., Liao, W., Wang, J.: Recognition of macrofungi by convolutional neural networks with attention mechanism. In: International Conference on Automation, Robotics and Computer Engineering. pp. 1–4 (2022)
4. Khare, N., Viswanathan, P.: Decision tree-based fraud detection mechanism by analyzing uncertain data in banking system. In: Emerging Research in Data Engineering Systems and Computer Communications. pp. 79–90 (2020)
5. Li, A., Qin, Z., Liu, R., Yang, Y., Li, D.: Spam review detection with graph convolutional networks. In: Proceedings of ACM International Conference on Information and Knowledge Management. pp. 2703–2711 (2019)
6. Liang, X., Yang, Z., Wang, B., Hu, S., Yang, Z., Yuan, D., Gong, N.Z., Li, Q., He, F.: Unveiling fake accounts at the time of registration: An unsupervised approach. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 3240–3250 (2021)
7. Liu, C., Sun, L., Ao, X., Feng, J., He, Q., Yang, H.: Intention-aware heterogeneous graph attention networks for fraud transactions detection. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 3280–3288 (2021)
8. Liu, Y., Zhu, L., Ding, L., Huang, Z., Sui, H., Wang, S., Song, Y.: Selective ensemble method for anomaly detection based on parallel learning. Scientific reports **14**(1), 1420–1420 (2024)
9. Murnane, D., Thais, S., Wong, J.: Semi-equivariant gnn architectures for jet tagging. Journal of Physics: Conference Series **2438**(1) (2023)
10. Pang, G., Cao, L., Chen, L.: Outlier detection in complex categorical data by modelling the feature value couplings. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. p. 1902–1908 (2016)
11. Pei, Y., Huang, T., van Ipenburg, W., Pechenizkiy, M.: Resgcn: attention-based deep residual modeling for anomaly detection on attributed networks. Machine Learning **111**(2), 519–541 (2022)
12. Perera R., A.V.: Multiscale graph neural networks with adaptive mesh refinement for accelerating mesh-based simulations. Computer Methods in Applied Mechanics and Engineering **429** (2024)
13. Rtayli, N., Enneya, N.: Enhanced credit card fraud detection based on svm-recursive feature elimination and hyper-parameters optimization. Journal Of Information Security And Applications **55**, 102596 (2020)
14. Sarma, D., Alam, W., Saha, I., Alam, M.N., Alam, M.J., Hossain, S.: Bank fraud detection using community detection algorithm. In: International Conference on Inventive Research in Computing Applications. pp. 642–646 (2020)
15. Schwing, G., Grosu, D., Schwiebert, L.: Parallel maximum cardinality matching for general graphs on gpus. In: IEEE International Parallel and Distributed Processing Symposium. pp. 880–889 (2024)
16. Song, Y., Li, X., Li, F., Yu, G.: Learning from feature and global topologies: Adaptive multi-view parallel graph contrastive learning. Mathematics **12**(14), 2277–2277 (2024)
17. Sserwadda, A., Ozcan, A., Yaslan, Y.: Structural and topological guided GCN for link prediction in temporal networks. Journal Of Ambient Intelligence And Humanized Computing **14**(7), 9667–9675 (2023)

18. Taha, A.A., Malebary, S.J.: An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine. Institute of Electrical and Electronics Engineers Access **8**, 25579–25587 (2020)
19. Tang, J., Su, Q., Su, B., Fong, S., Cao, W., Gong, X.: Parallel ensemble learning of convolutional neural networks and local binary patterns for face recognition. Computer Methods And Programs In Biomedicine **197**, 105622 (2020)
20. Thennakoon, A., Bhagyani, C., Premadasa, S., Mihiranga, S., Kuruwitaarachchi, N.: Real-time credit card fraud detection using machine learning. In: International Conference on Cloud Computing, Data Science Engineering. pp. 488–493 (2019)
21. Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., Anderla, A.: Credit card fraud detection - machine learning methods. In: International Symposium INFOTEH-JAHORINA. pp. 1–5 (2019)
22. Xie, Y., Yu, F., Achan, K., Panigrahy, R., Hulten, G., Osipkov, I.: Spamming botnets: signatures and characteristics. In: Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. pp. 171–182 (2008)
23. Zhe, H., Kexin, C., Xiaofei, X.: A graph neural network (gnn) algorithm for constructing the evolution process of rural settlement morphology. Security and Communication Networks **2022** (2022)
24. Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.: Graph neural networks: A review of methods and applications. Artificial Intelligence Open **1**, 57–81 (2020)