Journal of Computer Technology and Software

ISSN:2998-2383

Vol. 3, No. 5, 2024

Adaptive Spatio-Temporal Aggregation for Temporal Dynamic Graph-Based Fraud Risk Detection

Wenjun Gu[1], Mengfang Sun[2], Bingyao Liu[3], Ke Xu[4], Mingxiu Sui[5]

[1]Johns Hopkins University, Maryland, USA

[2]Stevens Institute of Technological, Hoboken, USA

[3]University of California, Irvine, Irvine, USA

[4]Columbia University, New York, USA

[5]University of Iowa, Iowa City, USA

Correspondence should be addressed to Mingxiu Sui; suimingx@gmail.com

Abstract: This paper introduces an advanced fraud detection algorithm, AT-GCN, tailored for temporal dynamic graphs frequently encountered in financial domains such as money laundering and financial fraud. Traditional graph neural networks (GNNs) have been predominantly successful in static graph analysis, lacking the capability to capture the temporal dynamics of transactions. To address this, the proposed AT-GCN algorithm integrates three key innovations: adaptive parameter updates using LSTM to reflect the temporal evolution of graph structures, a resampling method across time steps to balance label distribution and leverage temporal correlations, and a novel similarity-based weighted aggregation approach that enhances the differentiation of node importance within the graph. The LSTM component allows the model to dynamically adjust to changes in graph topology, capturing temporal dynamics effectively. The oversampling strategy mitigates label imbalance by connecting nodes across various time steps using Euclidean distance, enriching the model's fraud detection accuracy. The aggregation method, underpinned by a machine learning perceptive model, assigns weights based on node similarity, thereby tackling the disguise problem posed by fraudulent entities. Empirical results demonstrate AT-GCN's superiority over existing methods, showcasing its potential in real-world dynamic graph applications.

Keywords: Financial Fraud Detection, Graph Neural Networks (GNN), Spatio-temporal Aggregation

1. Introduction

Financial fraud refers to the act of illegally obtaining benefits from financial institutions or others through deception, false statements, manipulation of information, and other means. The research was conducted on the disguise problems of fraudsters and the aggregation process of graph neural networks, successfully resolving the existing issues by setting up a neighbor noise filter and a central node enhancer, ultimately achieving accurate identification of fraudulent behavior. However, the research on financial fraud is more focused on static graphs where nodes and edges do not change. The traditional graph representation method is inspired by the learning of word vector representation in Word2Vec, designing random walk strategies in the graph to obtain a sequence of nodes, and then inputting the node sequence into the SkipGram model to learn the low-dimensional vector representation of nodes. The emergence of GNN is mainly to learn the neighborhood information in the graph by designing graph convolutional operators, but these methods are all learning of node representation in static graphs and lack consideration of the dynamics of the graph.

Dynamic graphs appear in many places in the issue of financial fraud, such as money laundering, financial fraud, etc. Transactions in the financial field are often real-time and dynamically changing, and the algorithms in static graphs do not learn the temporal relevance in transactions very well.

Graph neural networks are a general model capable of performing various learning tasks on graphs. Currently, graph neural networks have achieved great success in the learning of static graphs [1]. However, in a financial transaction network where users are nodes and transactions between users are edges, users and transactions change over time, and the structure of the graph also changes. Therefore, the identification and processing of static graphs can no longer meet the new needs of current fraud detection. Although graph neural networks are actively exploring applications in the direction of dynamic graphs, there are still some issues. Currently, GNNs based on dynamic graphs have not combined advanced methods of GNNs based on static graphs, which still have limitations in the process of dynamic graph evolution over time and aggregation methods. Overcoming these limitations is crucial for dynamic graph applications in the real world.

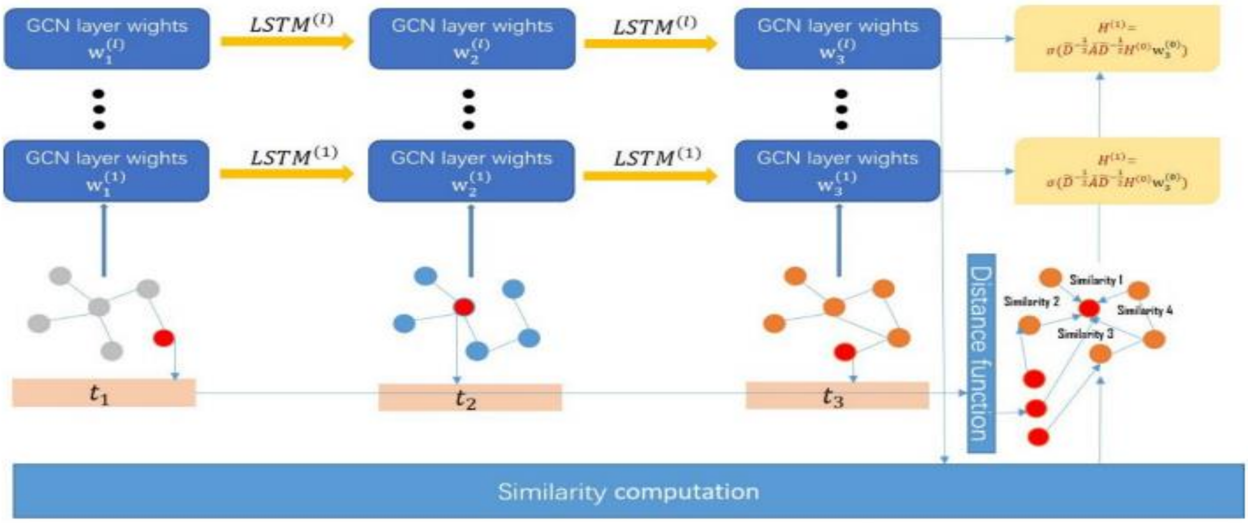

Figure 1. AT-GCN model architecture with time step set to 3 as an example

This paper proposes a fraud detection algorithm based on temporal dynamic graphs: Spatio-temporal and similarity aggregation based GCN (AT-GCN), addressing the temporal relevance in dynamic graphs and the aggregation method of graph neural networks. The main contributions of this paper include:

(1) The method uses LSTM to evolve the parameters of GCN, enabling GCN to adaptively perceive changes in graph structure and capture dynamic information in the evolving network parameters, thus allowing for more flexible graph sequences to be processed.

(2) A resampling method across time steps is proposed, using Euclidean distance as the distance function, establishing connections between nodes across different time steps by comparing the distances between nodes. This method can address the issue of label imbalance within time steps.

(3) In terms of aggregation, we proposes a new aggregation method that weights based on similarity, which can consider the similarity between the central node and neighbor nodes of GCN and assign different weights to alleviate the disguise problem of fraudsters.

2. Background

2.1. Related Work

The field of financial fraud detection using graph neural networks (GNNs) has experienced notable advancements, particularly in addressing the dynamic nature of transaction networks. Traditional GNNs, while successful in static graph analysis, often struggle with the evolving structure of temporal dynamic graphs. The proposed AT-GCN model directly tackles these limitations by integrating LSTM for adaptive parameter updates, a resampling method to balance label distribution across time steps, and a similarity-based weighted aggregation approach, drawing from various deep learning advancements. Cheng et al. [2] contributed significantly with their GNN-CL model, emphasizing the capture of both global and local patterns within static graphs, which sets the stage for extending these principles to dynamic graphs as done in AT-GCN. In the broader deep learning context, Zhong et al. [3] explored generative adversarial networks (GANs) and traditional methods for image recognition, with their findings relevant to enhancing feature extraction processes in fraud detection models.

Further, Cheng et al. [4] investigated ELMo word embeddings and multimodal transformers, primarily in natural language processing (NLP) and image analysis, but their techniques inform how AT-GCN aggregates information from different temporal snapshots in dynamic graphs. Similarly, Yang et al. [5] advanced emotional analysis with large language models, showcasing deep learning's potential to grasp complex patterns in unstructured data, akin to how AT-GCN addresses temporal and spatial complexities in financial transactions. Wang et al. [6], in their work on LSTM networks for predicting stock market trends, addressed limitations of recurrent neural networks (RNNs), directly influencing the adaptive parameter updates in AT-GCN. Additionally, Wang et al. [7] explored anomaly detection and risk assessment in financial markets using deep neural networks, emphasizing robust detection mechanisms necessary in financial contexts, aligning with AT-GCN's goals. Lastly, Zheng et al. [8] introduced adaptive friction in deep learning optimizers, which enhances the training process stability for models like AT-GCN, dealing with the intricacies of dynamic graphs. Together, these contributions underscore the AT-GCN model's position as a cutting-edge tool for detecting and mitigating fraudulent behavior in dynamic financial networks.

2.2. Parameter Definition

A graph can be represented as $G = \{V, E\}$, where $V = \{v_1, \dots, v_n\}$ represents the set of nodes, and $E = \{e_1, \dots, e_n\}$ represents the set of edges. Each node has a feature set that characterizes the features of the node, denoted as FvFv.

For a dynamic graph, each node v has a temporal feature t_v,. A dynamic graph $\mathcal{G} = \{G_t\}_{t=1}^{T}$ can be represented as a sequence of graph snapshots, where each snapshot is a static graph $G_t = \{V_t, E_t\}$, $V_t = \{v \in V \mid t_v = t\}, E_t = \{e \in E \mid t_v = t\}$. Different snapshots may have different sets of nodes. To better understand dynamic graphs.

Graph neural networks learn node embeddings through iterative aggregation of messages from the network neighborhood, with the embedding matrix $H^{(l)} = \{h_v^{(l)} \mid v \in V\}$ representing all the node embeddings at layer I. The essence of GNN layers is message passing and message aggregation. The process of GNN can be written as:

$$\text{message}_{u \to v}^{(l)} = MSG^{(l)}\left(h_u^{(l-1)}, h_v^{(l-1)} \mid u \in \mathcal{N}(v)\right)$$

$$h_v^{(l)} = AGG^{(l)}\left(\{\text{message}_{u \to v}^{(l)}\}, h_v^{(l-1)} \mid u \in \mathcal{N}(v)\right)$$

Where $h_v^{(l)}$ represents the node embedding of node v after the I-th layer of GNN, $h_v^{(0)} = x_v, message_{u \to v}^{(l)}$ represents the information embedding of node v, and $\mathcal{N}(v)$ represents the set of neighbors of node v. Different GNNs can have different definitions for message passing functions $MSG^{(l)}(\cdot)$ and aggregation functions $AGG^{(l)}(\cdot)$,. Typically, in GNN, average aggregation functions or max-pooling aggregation can be used. The GNN defined by equations is called static GNN because it is designed to capture certain features of static graphs without capturing the dynamic information of the graph.

Graph convolutional networks make some improvements based on graph neural networks. For traditional graph neural networks, only the node feature matrix and the adjacency matrix are needed as inputs to perform aggregation operations. However, graph convolutional networks also require a degree matrix, which represents how many nodes are associated with the central node. At the same time, the adjacency matrix of the graph convolutional network not only considers the features of its neighbor nodes but also considers its own weighting. The implementation method of GCN is as shown in Equation:

$$h_v^{(l+1)} = \sigma\left(D^{-\frac{1}{2}} \tilde{A} D^{-\frac{1}{2}} h_v^{(l)} W^{(l)}\right)$$

where $\tilde{A} = A + I$ represents the sum of the adjacency matrix A and the identity matrix I, allowing GCN to consider its own weighting while considering the neighbors. D represents the degree matrix, σ represents the activation function, and $W^{(l)}$) is the learnable parameter matrix.

3. Method

3.1.Model Structure

The model mainly consists of three parts: First, an LSTM dynamic update module; second, a time window oversampling; and third, a similarity-based aggregation module.

Figure 1. provides a schematic diagram of the model framework. Existing methods require consideration of global information to obtain the embedding of a node, but they do not handle the temporal changes of the node information well. The dynamic update module uses LSTM to capture the evolution process of the GCN's learnable weight matrix, fully considering the temporal characteristics of the dynamic graph. GCN can consider the spatial features of the dynamic graph at a certain moment, so the combination of GCN and LSTM can fully consider the spatiotemporal characteristics of the dynamic graph. Due to label imbalance, oversampling of fraudulent nodes can balance the labels. Using a similarity-based aggregation method can make the aggregation phase pay more attention to the features of neighbors similar to the central node.

Although GCN can consider the spatial structure of the graph, too much consideration of the global structure may actually lead to a decrease in performance. Therefore, adopting a similarity-based aggregation method allows GCN to pay more attention to neighbors similar to the central node during the aggregation process, which can enhance the detection effect.

3.2. GCN Adaptive Parameter Update Method

In the process of money laundering detection, since the graph has temporal relationships and spatial structural relationships, it is considered to use GCN for the spatial structure learning of dynamic graphs. Taking into account the temporal relationships of dynamic graphs, when using GCN for the learning of dynamic graphs, the weights of GCN at each moment also have certain temporal relevance [9]. LSTM is a neural network with the ability to remember long-term and short-term information. LSTM introduces gating mechanisms to control the flow and loss of features, and designs a memory cell with selective functionality, which can choose to remember important information and filter out useless information.

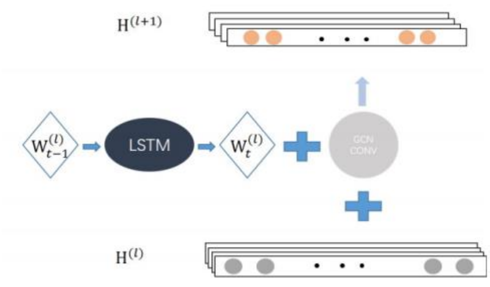

Figure 2. Illustration of the adaptive parameter update

3.3. Resampling Methods Across Time Steps

In the context of dynamic graph fraud detection, each snapshot of the dynamic graph at each time step is often imbalanced in terms of data, and there is typically temporal correlation within the dynamic graph. That is, the fraud patterns from the previous time period may also appear in the next time period and can have a positive impact on the predictions for the subsequent time period. Therefore, a resampling method across time steps is proposed. The schematic diagram of the resampling is shown in Figure 3:

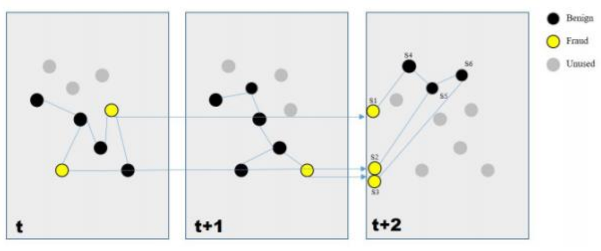

Figure 3. Oversampling methods across time steps

Due to the imbalance in the labels within each dynamic graph snapshot at each time step, this can have a negative impact on the experimental results. Therefore, it is important to balance the labels reasonably. However, dynamic graphs are often temporally correlated, so incorporating fraud nodes from the previous time period into the dynamic graph snapshot of the subsequent time step can have a positive effect.

The first step of resampling involves selecting fraud nodes from each time step within the time window, and the resampling formula is shown below:

$$C_t\ Fsample = \sum_{t}^{t+Tw} Sample_{\ t}$$

$$N = N.add(Fsample)$$

By changing different time windows, you can control the number of time steps and the amount of resampling needed.

Since fraud nodes from the previous time window are added to the graph of the subsequent time step, it is necessary to consider how to connect these fraud nodes with the nodes of the subsequent time period. Therefore, a measurement method is needed. The widely used distance function is the Euclidean distance function, which is:

$$D = \|x_v - x_u\|$$

After calculating the distances, the resampled nodes need to be added to the current time step and connected with the nodes in the current time step. By calculating the difference between the Euclidean distance of each node in the current time step and the Euclidean distance of the resampled nodes, and establishing connections with the resampled nodes that are close in distance, nodes from different time steps can also participate in the training and have a positive impact on the training of the current time step.

$$Top = Select(\{Sort(D_v - D_u) | u \in Fsample\})$$

$$E = E.add(v, u) where u \in Top, v \in V$$

where the Select function represents the selection of the resampled node uu with the smallest distance difference from node v.

4. Experiment

4.1. Dataset

This paper uses the Elliptic[10] dataset to verify the effectiveness of the model, which is currently the largest labeled Bitcoin transaction dataset in the world. The Elliptic dataset constructs and labels a graph from the original Bitcoin transactions, where nodes represent transactions and edges represent the flow of bitcoins from one transaction to the next. If the entity initiating the transaction (i.e., the node) belongs to the legal category, then the given transaction is classified as a legal category. The Elliptic dataset consists of 203,769 transaction nodes and 234,355 directed transaction payment edges. The dataset also includes 49 different steps, with an average time interval of about 2 weeks between time steps, and there are no edges connecting different time steps. Each node in the dataset has 166 features, and each node contains a label with a total of three categories: "Legal," "Illegal," and "Unknown." Of the 166 features, the first 94 features represent local information of the transaction, including the time step, transaction fees, etc., and the last 72 features are aggregated features obtained by aggregating information one hop forward or backward from the central node. This includes information such as the number of adjacent transactions of the node. Among these nodes, 21% are labeled as legal, only 2% are labeled as illegal, and the rest are labeled as unknown. The dataset is processed using the Linked Data methodology, which integrates various data formats, a critical aspect in scholarly research[11]. This structured approach enhances data cross-referencing, thereby improving interoperability between different datasets. This capability is especially valuable in fields like machine learning and artificial intelligence, where the quality of data is paramount for training models effectively and achieving accurate results.

4.2. Baselines

The experiments selected GCN[12] and GAT[13] as conventional baseline algorithms, as well as EvolveGCN, which is an advanced algorithm currently available, for comparison to demonstrate the effectiveness of the algorithm proposed in this paper in the problem of fraud detection based on dynamic graphs.

4.3. Experiment Results Analysis

To verify the effectiveness of the proposed method, ablation experiments were conducted using the first 94 local features and all features for different methods. GCN+W is without resampling and similarity-based aggregation methods, only using LSTM for adaptive parameter updates. GCN+A is a simplified version of AT-GCN that only uses similarity-weighted aggregation methods. GCN+O only uses resampling methods across time steps. The experimental results show that even using a single module in the spatiotemporal graph convolutional fraud detection strategy is better than GCN.
When only using 94 local features, after using the GCN adaptive parameter update method proposed in this paper, Precision increased by 14.2%, Recall increased by 22.6%, and F1 increased by 17.9%. When only using the resampling method across time steps, the effect of GCN also improved, with Precision increasing by 10%, Recall

increasing by 20.6%, and F1 increasing by 15%. When only using the similarity-weighted aggregation method, Precision increased by 11.4%, Recall increased by 13.9%, and F1 increased by 12.7%. The experimental comparison is shown in Figure 4.

When using all features, due to the addition of more information, the effect of all models is improved. In this case, using a single method in the spatiotemporal graph convolutional money laundering strategy still greatly improves the effect compared to GCN.

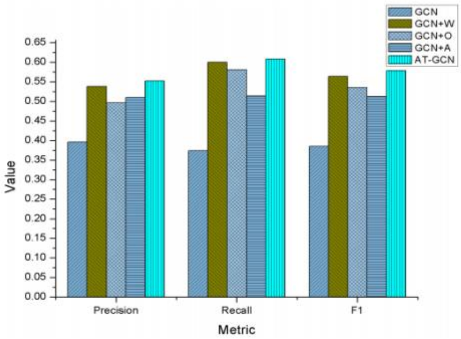

Figure 4. Performance comparison using 94 local features

5. Conclusion

This study presents the Adaptive Temporal Graph Convolutional Network (AT-GCN), a pioneering fraud detection model designed for the dynamic landscapes of financial transactions. Our approach addresses critical shortcomings in existing graph neural networks by incorporating three novel components: LSTM-based adaptive parameter updates, a strategic oversampling across temporal steps, and a similarity-based weighted aggregation technique. These elements collectively enhance the model's capacity to adapt to and accurately identify evolving fraudulent behaviors within financial networks. Empirical evaluations of the AT-GCN have demonstrated its superiority over traditional methods, particularly in its robust detection capabilities and adaptability to complex transaction patterns. The ablation studies further corroborate the individual contributions of each component to the overall effectiveness of the system. Future research will explore scaling the AT-GCN for broader applications and further refining the model to accommodate the rapid evolution of fraud tactics. The potential extension of this model to other sectors with dynamic graph requirements underscores its versatility and the promising direction for subsequent advancements. In summary, the AT-GCN sets a new benchmark for fraud detection in financial systems, offering a sophisticated tool that is both adaptable and rigorously validated through extensive testing.

References

[1] Kwok, Y. K., & Ahmad, I. (1999). Static scheduling algorithms for allocating directed task graphs to multiprocessors. ACM Computing Surveys (CSUR), 31(4), 406-471.

[2] Cheng Y, Guo J, Long S, Wu Y, Sun M, Zhang R. Advanced Financial Fraud Detection Using GNN-CL Model. arXiv preprint arXiv:2407.06529. 2024.

[3] Zhong Y, Wei Y, Liang Y, Liu X, Ji R, Cang Y. A comparative study of generative adversarial networks for image recognition algorithms based on deep learning and traditional methods. arXiv preprint arXiv:2408.03568. 2024.

[4] Cheng X, Mei T, Zi Y, Wang Q, Gao Z, Yang H. Algorithm Research of ELMo Word Embedding and Deep Learning Multimodal Transformer in Image Description. arXiv preprint arXiv:2408.06357. 2024.

[5] Yang H, Zi Y, Qin H, Zheng H, Hu Y. Advancing Emotional Analysis with Large Language Models. Journal of Computer Science and Software Applications. 2024;4(3):8-15.

[6] Wang J, Hong S, Dong Y, Li Z, Hu J. Predicting stock market trends using LSTM networks: overcoming RNN limitations for improved financial forecasting. Journal of Computer Science and Software Applications. 2024;4(3):1-7.

[7] Wang B, Dong Y, Yao J, Qin H, Wang J. Exploring Anomaly Detection and Risk Assessment in Financial Markets Using Deep Neural Networks. International Journal of Innovative Research in Computer Science & Technology. 2024;12(4):92-98.

[8] Zheng H, Wang B, Xiao M, Qin H, Wu Z, Tan L. Adaptive Friction in Deep Learning: Enhancing Optimizers with Sigmoid and Tanh Function. arXiv preprint arXiv:2408.11839. 2024.

[9] Pareja A, Domeniconi G, Chen J, et al. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(04): 5363-5370.

[10] Weber, M., Domeniconi, G., Chen, J., Weidele, D. K. I., Bellei, C., Robinson, T., & Leiserson, C. E. (2019). Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv preprint arXiv:1908.02591*.

[11] Li, Y., Yan, X., Xiao, M., Wang, W., & Zhang, F. (2023, December). Investigation of Creating Accessibility Linked Data Based on Publicly Available Accessibility Datasets. In Proceedings of the 2023 13th International Conference on Communication and Network Security (pp. 77-81).

[12] Dudziak, L., Chau, T., Abdelfattah, M., Lee, R., Kim, H., & Lane, N. (2020). Brp-nas: Prediction-based nas using gcns. Advances in Neural Information Processing Systems, 33, 10480-10490.

[13] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903.