Federation University ResearchOnline

https://researchonline.federation.edu.au

Copyright Notice

This is the peer-reviewed version of the following article:

Liu, J., Xia, F., Feng, X., Ren, J., & Liu, H. (2022). Deep Graph Learning for Anomalous Citation Detection. *IEEE Transaction on Neural Networks and Learning Systems, 33*(6), 2543–2557.

Available online: https://doi.org/10.1109/TNNLS.2022.3145092

See this record in Federation ResearchOnline at:
http://researchonline.federation.edu.au/vital/access/HandleResolver/1959.17/186627

Deep Graph Learning for Anomalous Citation Detection

Jiaying Liu, Feng Xia, *Senior Member, IEEE*, Xu Feng, Jing Ren, and Huan Liu, *Fellow, IEEE*

Abstract—Anomaly detection is one of the most active research areas in various critical domains, such as healthcare, fintech, and public security. However, little attention has been paid to scholarly data, i.e., anomaly detection in a citation network. Citation is considered as one of the most crucial metrics to evaluate the impact of scientific research, which may be gamed in multiple ways. Therefore, anomaly detection in citation networks is of significant importance to identify manipulation and inflation of citations. To address this open issue, we propose a novel deep graph learning model, namely GLAD (Graph Learning for Anomaly Detection), to identify anomalies in citation networks. GLAD incorporates text semantic mining to network representation learning by adding both node attributes and link attributes via graph neural networks. It exploits not only the relevance of citation contents but also hidden relationships between papers. Within the GLAD framework, we propose an algorithm called CPU (Citation PUrpose) to discover the purpose of citation based on citation texts. The performance of GLAD is validated through a simulated anomalous citation dataset. Experimental results demonstrate the effectiveness of GLAD on the anomalous citation detection task.

Index Terms—Anomalous citation, Deep graph learning, Network representation, Scholarly network analysis

I. INTRODUCTION

AN objective and fair evaluation of the impact of published research is essential for science itself. Since citations are important for assessing the scholarly impact of academic entities (e.g., publications, scholars, and institutions), it becomes essential to avoid making citation-based bibliometrics strongly flawed and defective. A branch of Science of Science (SciSci) [1] entitled Scholarly Network Analysis (SNA) [2] has emerged intending to discover meaningful insights to implement data-driven research decisions. Citation network analysis is one of the important points of interest in SNA, aiming to find relationships between cited papers and citing papers. Expansion of large open databases such as Web of Science (WOS), Scopus, and Microsoft Academic Graph (MAG) has opened up opportunities for multi-perspective and systematic analysis of citation networks [3].

Traditionally, the quantity of citations is one measure of the quality of a paper, or the impact of academic entities [4]. For

J. Liu is with School of Economics and Management, Dalian University of Technology, Dalian 116024, China.

F. Xia and J. Ren are with School of Engineering, IT and Physical Sciences, Federation University Australia, Ballarat, VIC 3353, Australia.

X. Feng is with School of Software, Dalian University of Technology, Dalian 116620, China.

H. Liu is with School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85281, USA.

Corresponding author: Feng Xia; email: f.xia@ieee.org.

example, existing bibliometric indicators such as h-index, g-index, and Journal Impact Factor (JIF) all focus on quantitative evaluation aspect while ignoring qualitative aspects. However, with the explosive growth of publications, the number of citations increases fast. As citation-based metrics are transformed into "academic currency", the credit of citations also implies economical gains. Suspicious citations are one of many other issues in the context of broader unethical practices that affect science academic integrity and fairness. Specifically, some citations are solely intended to boost the unwarranted impact of the publication, rather than disseminate scientific advancements. Although researchers have realized that not all citations are equal and tried to assign different weights to citations with different types, most studies have focused on distinguishing and weighting citations of a single type. In fact, as Prabha [5] illustrated, more than 2/3 references in a paper are unnecessary, which also confirms the existence of dubious citations.

These misused citations can cause a series of side effects. On the one hand, they affect citation-based measurement indices related to scholar promotion and reward in bibliometrics. On the other hand, they may mislead readers about the paper correct value. Worst of all, a large number of meaningless citations can negatively impact academic integrity. Therefore, anomaly detection in the citation network is critical to identify manipulation and inflation of citations. To the best of our knowledge, there exists no related method in the relevant literature that is directly applicable to automatic detection of anomalous citations.

Conventional anomaly detection techniques cannot tackle this problem well because of the complexity of graph data (e.g., irregular structures and relational dependencies). Compared with the traditional conventional detection methods, anomaly detection methods based on graph learning can preserve the node attributes and network structures at the same time during the learning process. However, most of the anomaly methods based on graph learning focus on detecting anomalous nodes in the network, such as GCNSI [6] and GCAN [7]. To date, only a few works have put their efforts into utilizing graph learning techniques for anomalous edge detection. At the same time, most of the anomalous edge detection methods are task-dependent and are designed for specific tasks such as spam review detection and rumor detection [8], [9], [10]. These methods cannot be directly applied to anomalous citation detection. The major issue of direct usage of graph learning in anomalous citation detection lies in the fact that they can't integrate edge feature learning process and semantic analysis such as citation purpose, which cannot be ignored in

the learning process. It drives us to propose a new method that considers both anomalous citation features and the structure of the citation network to identify anomalous citations.

In this work, as a first step, we define a novel class of citations, namely anomalous citations, which contains two aspects: semantic information and relational information. We say that a citation is anomalous if i) the cited paper and the citing paper are disparate from the perspective of contents, and there is no clear purpose in the scientific citation text, and ii) the citing paper contains excessive relational citations. We then propose a framework based on deep graph learning, namely GLAD (Graph Learning for Anomaly Detection), to detect the anomalous paper pairs. GLAD incorporates text semantic mining to network representation learning by adding both node attributes and link attributes via graph neural networks. In addition, we propose an algorithm called CPU (Citation PUrpose) to identify the purpose of citation based on citation context. To validate the performance of GLAD, we generate a simulated dataset for anomalous citations. We compare GLAD's performance with that of state-of-the-art approaches for network embedding. GLAD outperforms baseline approaches by up to 37% in the anomalous citation detection task. Our major contributions can be summarized as follows:

- We formally define a novel class of relationships introducing both semantic information and relational information, which covers universal real-world issues.
- We provide an effective algorithm named CPU based on transfer learning to solve the problem of unmarked citation purposes. It can automatically judge the relationship between the cited paper and the citing paper based on the citation context.
- We propose a deep graph learning framework, namely GLAD, for anomalous citation detection. GLAD can model rich information with integrating attributes such as text content and author relationship.
- Extensive experiments have been conducted. The results verify the effectiveness and superiority of GLAD as compared against state-of-the-art baselines.

The rest of this paper proceeds as follows. Section II briefly reviews the related work the related work of SNA and GNNs. In Section III, we introduce mathematical preliminaries and formally formulate the problem of anomalous citation detection. Section IV illustrates our optimal solution, GLAD and describes its architecture in detail. Experimental settings and results are described in Section V. Finally in Section VI, we conclude the paper.

II. RELATED WORK

Although numerous techniques have been developed for anomalies detection in different areas such as financial security and image/video surveillance [11], limited work has paid attention to anomalous citation detection in academia. Hence, in this part, we introduce topics that are most relevant to our research topics, including scholarly network analysis and graph neural networks.

A. Scholarly Network Analysis

It is well known that citation count is an important index to measure academic entities such as papers, journals, and scholars [12], [13]. Indeed, references in the paper are often overlooked and there is also randomness to a certain extent in citation behavior. Prabha [5] has pointed out that only less than 1/3 references in each paper are mandatory. Moustafa [14] introduces the idea that there are multiple inherent biases in citation practice, which will make citation-based bibliometric measures strongly flawed and defective. For example, authors can easily inflate their h-indices and distort scientific knowledge towards more conformism by manipulating self-citations [15]. Inappropriate co-author self-citations and collaborative self-citations can also mislead and distort scientific literature, thereby challenging scientific fairness. Reciprocal citations indicate that authors prefer to cite publications of people who cite their own work rather than those who do not [16]. Moreover, to inflate the journal impact factor, some editors have encouraged authors to cite the journal's own papers, which will lead to journal citation stacking. Previous studies have explored the relationships between journal impact factors and self-citations [17]. Another similar kind of citations is called coercive induced self-citation [18], which may occur in reviewers, grants, advisors, etc.

Under pressure to ensure that journals/authors maintain high-value bibliometric indicators, i.e., JIF and h-index, some editors have artificially generated citations for their journals/papers by forcing authors to submit papers or by writing reviews. "Citation cartel" refers to the behavior of journal groups exchanging citations with excessive frequency [19]. Increasing competition in the academic publishing market has led to such practice becoming more common in recent years. Many instances of cartels have been reported. For instance, Clarivate Analytics, the publisher of the annual Journal Citation Report (JCR), has featured and suspended journals which distort JIF caused by different anomalous citation patterns[1]. The Scholarly Kitchen also reports citation cartel cases[2]. Recipient journals can enhance their JIF by up to 94% by receiving citations from donor journals in the JIF years. As such indicators can reflect the quality and prestige of journals, it is essential to detect such behaviors for academic fairness.

Unlike self-citations which are easy to detect, the issue of journal citation cartels has not been widely documented. Few algorithms are designed for detecting citation cartels. Determining the boundaries of "excessive" has become the biggest challenge. It seems feasible to detect such anomalous pairs based on network community approaches. However, such approaches may lead to false positives because publications in one journal tend to cite publications in other similar journals in the same research field, thus forming a closely-connected community.

Although the aforementioned literature explains the existence of anomalous citations through citation network analysis and publishing ethics elaboration, to the best of our

---

[1]http://help.prod-incites.com/incitesLiveJCR/JCRGroup/titleSuppressions
[2]https://scholarlykitchen.sspnet.org/2012/04/10/emergence-of-a-citation-cartel/

knowledge, automatic anomalies detection methods for citation networks have not been well studied. On the one hand, existing graph anomaly detection algorithms cannot capture abnormal subtle signal strengths to support pairwise anomalies detection in citation networks. On the other hand, the lack of a ground-truth dataset with considerable size challenges the application of supervised machine learning methods in anomalies detection.

B. Graph Neural Networks

The concept of graph neural network (GNN) was initially outlined by Gori et al. [20], which is a kind of neural network specially designed for processing highly irregular graph data. GNN models can be categorized into Recurrent Graph Neural Networks (RecGNNs), Convolutional Graph Neural Networks (ConvGNNs), Graph Autoencoders (GAEs), and Spatial-Temporal Graph Neural Networks (STGNNs) [21].

RecGNNs rely on recurrent neural architectures and apply the same parameter set recurrently over nodes to learn node representations. They assume a node constantly exchanges information/message with its neighbors in a graph until a stable equilibrium is reached. Early research on RecGNNs includes GNN [22], GraphESN [23], Gated GNN [24]. ConvGNNs are closely related to RecGNNs. Different from RecGNNs, ConvGNNs handle the cyclic mutual dependencies architecturally by adopting a fixed number of layers with different weights. ConvGNNs can be categorized into spectral-based ConvGNNs and spatial-based ConvGNNs. Spectral-based approaches define graph convolution based on graph signal processing and spatial-based ConvGNNs including DCNN [25], GraphSAGE [26], FastGCN [27], GIN [28], and SemanticGCN [29] inherit ideas from RecGNNs by information propagation. Recently, scholars have paid attention to compute mutual information between high dimensional input/output pairs of deep neural networks in diverse domains such as images and speech. Specifically, in deep graph learning, Velickovic et al. [30] propose Deep Graph Infomax (DGI) for learning representations for graph-structured inputs by maximizing mutual information between global representations and the local patches in an unsupervised manner. Following on DGI, Ren et al. [31] propose Heterogeneous Deep Graph Infomax (HDGI) to learn node representations by maximizing local-global mutual information.

GAEs learn network embeddings and graph generative distributions by encoding nodes/graphs into the latent space and reconstructing the graph from encoded information. GAEs are available for network embedding (learn latent representations by graph structural information reconstruction) [32], [33] and graph generation (generate step by step or output the graph all at once) [34], [35]. STGNNs aim to capture the graph dynamics while assuming the interdependence between connected nodes. The dominant merit of STGNNs is that they can capture spatial and temporal dependencies of a graph simultaneously. STGNNs are divided into two directions, i.e., RNN-based methods [36] and CNN-based methods [37]. RNN-based methods filter the input and use state passed to the loop unit by graph convolution to learn the spatial-temporal

dependence. CNN-based methods interleave 1-D-CNN layers with graph convolutional layers in a non-recursive manner. Compared with RNN-based methods, CNN-based approaches can realize parallel computing, stable gradients, and low-memory requirements.

In addition to typical GNNs mentioned above, many variations of GNN have been developed recently, i.e., GRCN [38], GAUGM [39], GNN-Guard [40], and AMGCN [41]. It has been proven that GNN brings breakthrough improvements to fulfilling various tasks in different areas such as e-commerce, chemistry, and biomedicine. Specific applications of GNNs include computer vision, natural language processing, recommender systems, and so on [42], [43].

Graph anomaly detection with graph learning has received growing attention recently. Existing work can be categorized into anomalous nodes detection methods, anomalous edge detection methods, and anomalous subgraphs detection methods according to the anomalous graph objects that they can detect. The detection of anomalous nodes is the focus of scholars. Scholars focus on using GAT, GCN, and GAE to identify malicious users, fake news, and financial fraud. Only few works have put their efforts into the anomalous edge and subgraph detection based on graph learning. Ouyang et al. [44] model the distribution of edges based on a neural network and identify existing edges that are least likely to appear as anomalies. AddGraph [45] is a GCN-based framework for anomalous edge detection in dynamic networks. There is a huge gap between the existing anomalous edge/subgraph/graph detection techniques and the emerging demands for more advanced solutions in various applications.

III. PRELIMINARIES AND DEFINITIONS

Before we give details about the proposed framework, we first delve into the concept of anomalous citations and mathematical preliminaries used throughout the paper.

A. Definition of Anomalous Citation

The aforementioned literature explains the existence of different anomalous citation patterns through citation network analysis and publishing ethics elaboration. Unfortunately, there is no clear definition of anomalous citation, as well as automatic anomalous citation detection methods. This research aims to solve the problem of anomalies detection in the citation network. To this end, as a first step, we refer to the above-mentioned aberration citations to give the definition of anomalous citations.

*Definition 1: **Anomalous Citations***.

In this paper, the citation relationship between paper pairs that satisfies the conditions described below is regarded as anomalous:

1) The cited paper and the citing paper are disparate from the perspective of contents, and there is no clear purpose in the scientific citation context; and
2) The citing paper contains excessive relational citations.

B. Mathematical Preliminaries

In this part, we introduce the notations we will use in the remainder of the paper. An academic citation network G consists of a set of nodes $V = \{v_1, v_2, ..., v_{|V|}\}$ (represent papers) and a set of edges $E = \{e_1, e_2, ..., e_{|E|}\}$, which describe the interaction in terms of citing-cited relationship between nodes. The connection between nodes can be described by the adjacency matrix $\boldsymbol{A}$. $\boldsymbol{A}_{ij}$ denotes the element in i-th row and j-th column of $\boldsymbol{A}$. If there is a link between node v_i and v_j, $\boldsymbol{A}_{ij} = 1$, otherwise $\boldsymbol{A}_{ij} = 0$.

In this research, the anomalous citation detection problem is formulated as a binary classification problem. Specifically, the citation between each paper pair is either a non-anomalous citation or an anomalous one. Matrix $\boldsymbol{Y} \in \mathbb{R}^{|V| \times |V|}$ contains labels for each paper pair, where the value of the element located in its i-th row, j-th column $y_{ij} = 1$ means that the citation from paper i to paper j is anomalous, otherwise $y_{ij} = 0$. TABLE I provides detailed description of major notations we use in the remaining part.

TABLE I
DESCRIPTION OF NOTATIONS

Notation	Description
G	the academic citation network
V	the set of nodes (represent papers)
E	the set of edges (represent citing-cited relationship)
$n_1 = |V|$	the number of nodes in the citation network G
$n_2 = |E|$	the number of edges in the collaboration network G
Y	the label set for edges
y_L	the labeled citations
m_1	the dimension of the node attribute vectors
m_2	the dimension of the edge attribute vectors
d_1	the dimension of node representation
d_2	the dimension of edge representation
r	the learning rate for the autoencoder
$\boldsymbol{X} \in \mathbb{R}^{n_1 \times m_1}$	the node local feature matrix
$\boldsymbol{Z} \in \mathbb{R}^{n_2 \times m_2}$	relation features for the paper pairs (edge)
$\boldsymbol{A} \in \mathbb{R}^{n_1 \times n_1}$	node connection information (adjacency matrix)
$\boldsymbol{H}^n \in \mathbb{R}^{n_1 \times d_1}$	the final node representations
$\boldsymbol{H}^e \in \mathbb{R}^{n_2 \times d_2}$	the final edge representations

C. Problem Description

According to the above symbols and definitions, the input of the anomalous citation detection task includes a paper pair (i, j) associated with their local features $\boldsymbol{X}$, relation (edge) features $\boldsymbol{Z}$ for the paper pair, citation context between i and j, and partially labeled citations y_L. We aim to obtain labels for remaining unmarked citations by the learning and training process. That is, given a paper pair (i, j) and related feature information, GLAD can output the citation relationship between i and j.

IV. THE GLAD FRAMEWORK

In this section, we describe the proposed framework for identifying anomalous citations in detail. Specifically, the overall framework contains three main components, i.e., node feature learning, edge feature learning, and anomalous citation identification. Fig. 1 gives a graphic example of GLAD. Graph Neural Networks can effectively learn the complex structures of graph-structured data. In the academic citation network, both node and edge attributes can reflect the network properties. For example, from the perspective of node attributes such as the textual information, the research topic and content of a paper are closely related to the abstract of a paper, which helps to judge the similarity of two papers in topic and content. At the same time, edge attributes such as citation purpose and self-citation can reflect the conflict of interest between the two papers. Hence we incorporate attributes for both edges and nodes into GNNs models and formalize the task as modeling the relations between vertices. In order to preserve the global and local network structures in the joint space composed of network structures, node attributes, and edge attributes, we exploit the nodes' and edges' proximity jointly by using the GLAD framework. We hope that the process of feature learning can retain the feature information of both nodes and edges, and at the same time integrate the learned feature representations during the classification process. The individual node and edge feature learning processes are unsupervised. For node feature learning, we select DGI as the workhorse method for our node representation module because it can integrate node attributes and network structure by maximizing mutual information between patch representations and the corresponding graphical high-level summary based on GCN in a completely unsupervised manner. The edge feature learning process is established based on the autoencoder because it can learn the edge feature representations in an unsupervised manner and the and the learning results can be explained by reconstruction. In the following, we elaborate on the specific process of each part and the integrations of these parts.

A. Node Feature Learning

Features of nodes can reflect their inherent attributes. In our framework, node features can be regarded as the local properties of papers, such as research topics, and importance in the citation network. These features reflect the importance of papers and topic similarity between paper pairs. For example, cited papers and citing papers should be similar/related in research topics or contents. Node feature learning is carried out from two aspects, textual feature learning and local-global information fusion.

1) Textual Feature Learning: Textual information including titles, keywords, abstracts, and main texts is a major component of scientific publications. In order to preserve the local information as well as proximity of each paper from the perspective of textual feature, motivated by Doc2vec [46], the process of vectorizing text into a numeric space is illustrated as follows.

We first build up a corpus of text information (paper abstracts) as a set $T = \{t_1, t_2, ..., t_{|V|}\}$, where t_i is the list of paper i's abstract. In the training process, each abstract is represented by a unique vector in terms of a column of matrix $\boldsymbol{D} \in \mathbb{R}^{|V| \times p}$, where $|V|$ represents the number of the paragraphs (abstracts) and each abstract is mapped to p dimensions. Each word of the abstract is also represented by a unique vector in terms of a column in matrix $\boldsymbol{W} \in \mathbb{R}^{N_w \times q}$, where N_w is the number of words in the vocabulary and each
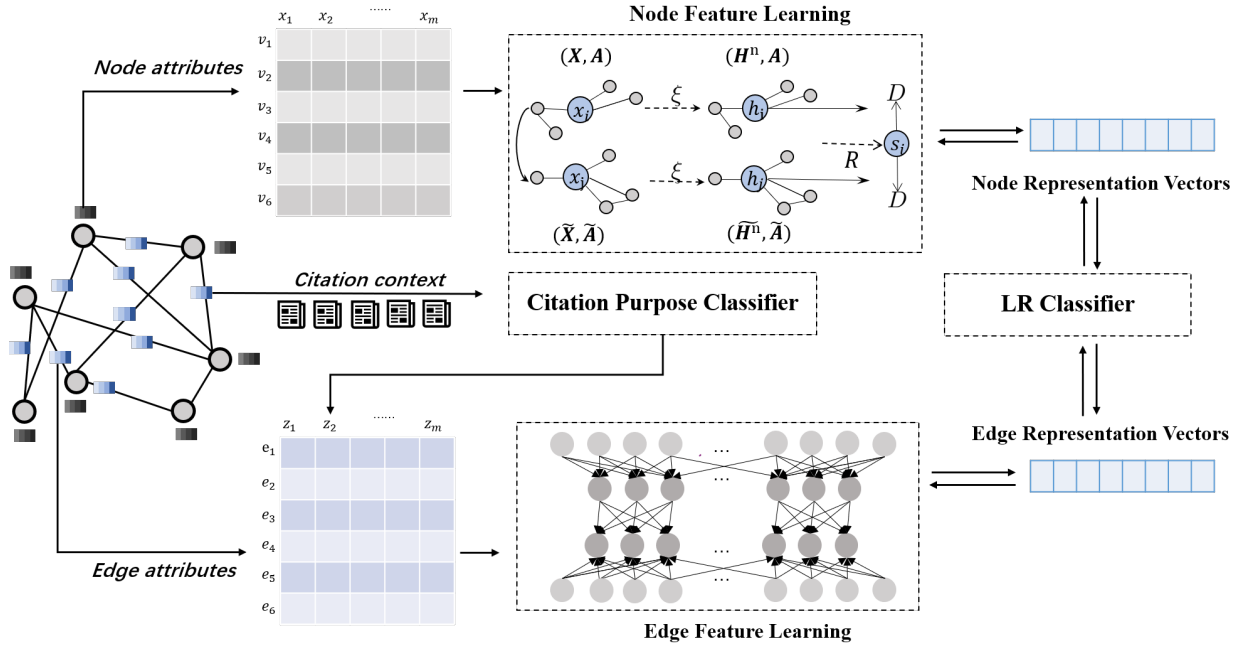
Fig. 1. The overall framework of GLAD.

word is mapped to q dimensions. The goal is to predict a word with the highest probability in the current context by giving a specific paragraph vector and several word vectors. Specifically, the given sequence of training words will be fixed through a preset window win. win represents the maximum distance between the given words and the predicted word in the sentence. The window will traverse the entire corpus. When sliding to position t, the abstract vector d_i of the abstract and the word vectors of $|win|$ words in the front and $|win|$ words in the back of the word w_t are used to predict the word w_t that will appear at this position with the maximum probability. The task is to maximize the average log-likelihood:

$$\frac{1}{M - 2|win|} \sum_{t=|win|}^{M-|win|} \log p(w_{t-|win|}, ..., \qquad (1)$$

$$w_{t-1}, w_{t+1}, ..., w_{t+|win|}, d_i)$$

where M is the number of all training words (the length of the sequence) and d_i is the document representation vector of the abstract list containing the context word in the current window. The prediction task is performed by hierarchical softmax:

$$p(w_t | w_{t-|win|}, ..., w_{t-1}, w_{t+1}, ..., w_{t+|win|}, d_i)$$
$$= \frac{\exp(Pr_{w_t})}{\sum_{j=1}^{N_w} \exp(Pr_{w_j})} \qquad (2)$$

where Pr is the log probability of the output, which is computed as:

$$Pr = U\boldsymbol{a}(w_{t-|win|}, ..., w_{t-1}, w_{t+1}, ..., w_{t+|win|}, d_i; \boldsymbol{W}, \boldsymbol{D}) + b \qquad (3)$$

where U and b are softmax parameters. $\boldsymbol{a}$ is constructed by averaging word vector extracted from $\boldsymbol{W}$ and document vector extracted from $\boldsymbol{D}$. The paper abstracts are vectorized in the latent space by using the PV-DM (Distributed Memory version

of Paragraph Vector) model [46], where similar papers will be close to each other.

*2) **Local-Global Information Fusion**:* Through the textual information learning process illustrated in Section IV-A1, we have obtained the textual representations for each paper. However, we hope that node representations can reflect not only textual information but also node importance in the citation network. To tackle this problem, inspired by Deep Graph Infomax (DGI) [30], the procedure for the node representations generation is fully summarized as follows.

For nodes in G, we first connect their textual representations and artificially defined features illustrated in TABLE II. Hence, a set of node features $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{n_1}\}$ is generated, where $n_1 = |V|$. $\boldsymbol{A} \in \mathbb{R}^{n_1 \times n_1}$ is also provided, where $\boldsymbol{A}$ represents node connection information. The final high-level node feature representations are learning through the encoder, $\xi : \mathbb{R}^{n_1 \times m_1} \times \mathbb{R}^{n_1 \times n_1} \to \mathbb{R}^{n_1 \times d_1}$.

TABLE II
DESCRIPTION OF ARTIFICIALLY DEFINED NODE FEATURES

Notation	Description
$\boldsymbol{d}$	textual representation of the paper
A_{cic}	citation in-degree centrality for the first author of the paper
A_{coc}	citation out-degree centrality for the first author of the paper
P_{odc}	citation out-degree centrality of the paper

The encoder learning process relies on maximizing local mutual information rather than minimizing the construction error. Negative examples are generated by using the corruption function $\chi : (\widetilde{\boldsymbol{X}}, \widetilde{\boldsymbol{A}}) = \chi(\boldsymbol{X}, \boldsymbol{A})$. χ modifies the network to obtain negative samples. Here, we perturb the node feature matrix by adding row-wise Gaussian noise, while keeping the graph structure unchanged. The patch representations for each node can be learned through graph convolutional networks ε,

which can integrate the information of neighbors for the target node. Information integration process can be formulated as:

$$\varepsilon(\boldsymbol{X}, \boldsymbol{A}) = \sigma(\widehat{\boldsymbol{D}}^{-\frac{1}{2}} \widehat{\boldsymbol{A}} \widehat{\boldsymbol{D}}^{-\frac{1}{2}} \boldsymbol{H}_l^n \boldsymbol{W}_l^n) \qquad (4)$$

where $\widehat{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$. $\widehat{\boldsymbol{D}}$ is the degree matrix of $\widehat{\boldsymbol{A}}$ and $\boldsymbol{H}^n$ is the learning feature of each layer. $\boldsymbol{W}_l^n$ is a learnable parameter matrix in the l-th layer of node representation learning. For the input layer, $\boldsymbol{H}_0^n = \boldsymbol{X}$. σ is a non-linear activation function. We can get the patch representation $\boldsymbol{h}_i^n$ for each node in positive samples set through Eq.(4). Similarly, we can also generate patch representations $\tilde{\boldsymbol{h}}_i^n$ through Eq.(4) for negative samples. Global summaries $\boldsymbol{s}$ are obtained by passing patch representations through the readout function $\mathcal{R}$:

$$\boldsymbol{s} = \mathcal{R}(\boldsymbol{H}^n) = \frac{1}{N_p} \sum_{i=1}^{N_p} \boldsymbol{h}_i^n. \qquad (5)$$

Note that N_p is the number of positive samples. A discriminator $\mathcal{D}$ is employed to discriminate positive and negative patch-summary pairs:

$$\mathcal{D}(\boldsymbol{h}_i^n, \boldsymbol{s}) = \sigma(\boldsymbol{h}_i^{nT} \boldsymbol{W}^n \boldsymbol{s}). \qquad (6)$$

where $\boldsymbol{W}^n$ is a learnable scoring matrix for the discriminator. The final objective function is:

$$L_n = \frac{1}{N_p + N_n} \Big(\sum_{i=1}^{N_p} \mathbb{E}_{(\boldsymbol{X}, \boldsymbol{A})}[\log \mathcal{D}(\boldsymbol{h}_i^n, \boldsymbol{s})]$$
$$+ \sum_{j=1}^{N_n} \mathbb{E}_{(\widetilde{\boldsymbol{X}}, \widetilde{\boldsymbol{A}})}[1 - \log \mathcal{D}(\widetilde{\boldsymbol{h}}_j^n, \boldsymbol{s})] \Big) \qquad (7)$$

where N_n is the number of negative samples. Thus we can generate the representation for each node by integrating local information and global structural information into the final feature representations.

B. Citation Purpose Classification

Citation context can provide important semantic information about the relation between the paper and its references, i.e., the authors' intention. The definition of anomalous citations brings the problem of identifying the purpose of the citation. That is, given a target paper i and its citation in a given paper j, we need to identify the intention behind selecting j and citing it by the author of i.

Previous studies have been devoted to studying different purposes of citing other publications, including comparison, use, criticizing, and so on. Researchers also have come up with different schemes composed of citation purpose categories. In this paper, we adopt a scheme that contains six purpose categories (as shown in TABLE III) after studying the previously used citation taxonomies [47].

We propose an algorithm to identify the citation purpose based on the citation context, which we refer to as CPU (Citation PUrpose) algorithm. Different from previous work, we incorporate the semantic recognition process into the citation purpose classification task, which is to train the classification model based on feature representations obtained from a set of labeled citation contexts. As shown in Fig. 2, CPU mainly

TABLE III
ANNOTATION SCHEME FOR CITATION PURPOSE

Category	Description
Criticizing	The citing paper describes the weakness or strengths of the reference
Comparison	The citing paper compares the present approach or results with the reference
Use	The citing paper uses the method, idea or tool of the reference
Substantiating	Current results, claims of the citing work substantiate or verify the reference or they support each other
Basis	The reference motivates the work of citing paper
Other	The citation doesn't belong to any of the above categories

consists of two parts: feature representation learning process and finetuning process.

To capture semantic information and bidirectional relations in sentences of the citation context, the most important step is to learn a good feature representation for each word. As there are few labeled samples for citation contexts, CPU adopts SCIBERT [48] for pre-training, which is a pre-trained language model established on BERT for scientific text. The core idea of pre-training is to learn good feature representations for words by running a self-supervised learning method on the basis of a massive scientific corpus. Thus we can directly use the feature representations as the word embedding feature of the classification task.

The overall framework for feature representation learning is established on a multi-layer transformer architecture, which is composed of several encoders and decoders. The encoder consists of multi-head attention and a full connection, which is used to convert input corpora into feature vectors. The input of the decoder is the output of the encoder. It is composed of masked multi-head attention, multi-head attention, and a full connection to output the conditional probability of final results. The input of pre-training $\mathbf{E}(i)$ is the unit sum of three embedding features, including wordpiece embedding, position embedding, and segment embedding. The architecture for the pre-training process includes two self-supervised tasks:

1) Masked Language Model. During training, some words are masked from the input corpus. Then the words are predicted by $\mathbf{T}(i)$, the output of the last Transformer at position i. In the experiment, 15% of WordPiece Token [49] will be randomly masked. When training the model, a sentence will be input into the model multiple times for parameter learning.

2) Next Sentence Prediction. This task is to determine whether two sentences follow each other by using the classification token $\mathbf{C}$ of the sentence. The training data is generated by randomly extracting two consecutive sentences from the parallel corpus. 50% of the consecutive sentences are retained as positive samples (IsNext). The second sentence of the remaining 50% is randomly extracted from the corpus. The relationship between these sentences is "NotNext".

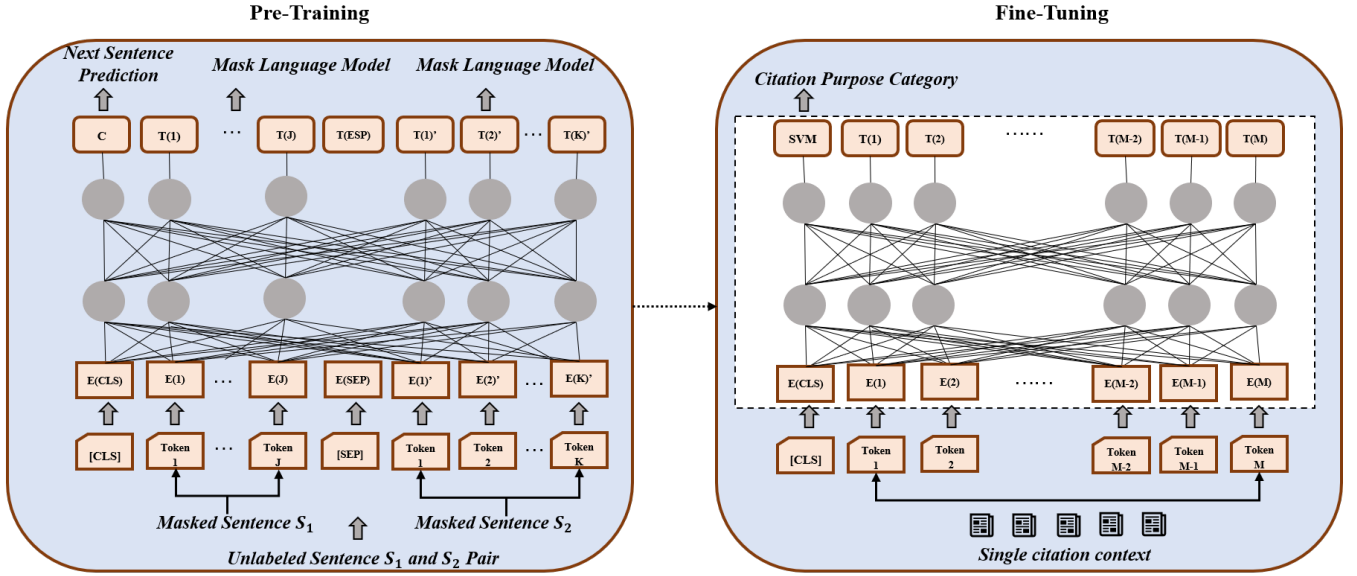In finetuning process, we feed final vectors for the token

Fig. 2. The architecture of CPU.

into a linear classification layer. For this purpose, we perform Support Vector Machines (SVM) on citation context representations. Since the problem is a multi-class classification problem, we need to train N_c classifiers, where there are a total of N_c classes of training data. The i-th training sample is (c_i, l_i), where c_i is the textual representation of the citation context learned by CPU. The category label $l_i \in \{1, 2, ..., N_c\}$. There are N samples. For each category presented in TABLE III, we need to train a binary classifier. Specifically, for each category, we take it as the positive class. All the samples of the remaining $N_c - 1$ categories are considered as the negative class. Hence we construct a binary SVM to separate the j-th category from the rest of the $N_c - 1$ category. It can be regarded as solving a quadratic programming problem:

$$\min_{\boldsymbol{w}^j, b^j, \varepsilon^j} \frac{1}{2}(\boldsymbol{w}^j)^T \boldsymbol{w}^j + R \sum_{i=1}^{N} \varepsilon_i^j$$
$$\text{s.t.} \quad (\boldsymbol{w}^j)^T \phi(\boldsymbol{c}_i) + b^j \geq 1 - \varepsilon_i^j, \text{if } l_i = j \quad (8)$$
$$(\boldsymbol{w}^j)^T \phi(\boldsymbol{c}_i) + b^j \leq -1 + \varepsilon_i^j, \text{if } l_i \neq j$$
$$\varepsilon_i^j \geq 0$$

where the subscript i represents the index of the sample. The superscript $j \in \{1, 2, ..., N_c\}$. $\boldsymbol{w}^j$ is a normal vector that determines the direction of the hyperplane. ϕ is used to map the input samples to the high-dimensional space. Specifically, we map the sample c_i to the feature space and get $\phi(\boldsymbol{c}_i)$. ε_i^j and R represent the slack variable and the regularization coefficient, respectively. The Lagrangian duality of Eq.(8) is:

$$\max \sum_{i=1}^{N} \alpha_i^j - \frac{1}{2} \sum_{i,n=1}^{N} \alpha_i^j \alpha_n^j l_i l_n \langle \phi(\boldsymbol{c}_i), \phi(\boldsymbol{c}_n) \rangle,$$
$$\text{s.t.} \quad 0 \leq \alpha_i^j \leq R, i = 1, 2, ..., N; \sum_{i=1}^{N} \alpha_i^j l_i = 0 \quad (9)$$

where α denotes the Lagrangian multiplier of inequality constraints. For $\boldsymbol{c}_{new}$, the decision function of type j adopts the following function:

$$d_{new}^j = (\boldsymbol{w}^j)^T \phi(\boldsymbol{c}_{new}) + b^j = \sum l_i \alpha_i^j k(\boldsymbol{c}_i, \boldsymbol{c}_{new}) + b^j \quad (10)$$

where k is the kernel function, $k(\boldsymbol{c}_i, \boldsymbol{c}_{new}) = \phi(\boldsymbol{c}_i)^T \phi(\boldsymbol{c}_{new})$. For $\boldsymbol{c}_{new}$, there are N_c decision functions and N_c outputs. The classification function f is:

$$f(\boldsymbol{c}_{new}) = \arg \max_{j \in \{1, 2, ..., N_c\}} [\sum l_i \alpha_i^j k(\boldsymbol{c}_i, \boldsymbol{c}_{new}) + b^j] \quad (11)$$

By solving the dual problem, the optimal Lagrangian multiplier vector can be obtained. Then we can get the decision function, which is used to classify new data.

C. Edge Feature Learning

Edge attributes can depict relationship intensity. Based on the definition of anomalous citations, edge features are used to help us determine whether there is a clear purpose and excessive relations exist between the cited paper and the citing paper. To preserve the edge proximity, motivated by Tu et al. [50], a deep autoencoder is used to obtain feature representations for edges in the network. An autoencoder consists of two parts, i.e., an encoder and a decoder. Thus hidden layers of the autoencoder can continuously encode and decode information obtained from the input data, thereby automatically capturing the characteristics of the input data and keeping them unchanged.

For each edge connecting the citing paper and the cited paper in the citation network, we consider the attributes presented in TABLE IV. Thus we can get the edge feature set represented as $\boldsymbol{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_{n_2}\}$, where $n_2 = |E|$.

CP is decided by classification results of CPU. If the classification results belong to one of the first five categories (Criticizing, Comparison, Use, Substantiating, and Basis), we treat them as citations with a clear purpose. *JF* represents the

TABLE IV
DESCRIPTION OF EDGE FEATURES

Notation	Description
CP	Whether the purpose of the citation is clear or not
SF	Whether the citation is a self-citation or not
SJ	Whether citing and cited papers are published in the same journal
JF	The proportion of the journal to which the cited paper published in the reference list of the cited paper
SI	Whether first authors of citing and cited papers have been working in the same institution
SR	Whether citing and cited papers have same references
AC	Whether authors of citing and cited papers have collaborated
CB	Whether the first author of the cited paper used to cite citing paper's first author before

proportion of the journal to which the cited paper published in the reference list of the cited paper. It can be calculated as Eq.(12):

$$JF = \frac{|P_{cj}|}{|R_e|} \quad (12)$$

where $|P_{cj}|$ is the number of papers in the reference list that publish in the same journal as the cited paper and $|R_e|$ is the number of reference of the citing paper.

The autoencoder adopts Z as input. In each hidden layer of the autoencoder, the transformation mechanism can be formulated as:

$$\begin{aligned} \boldsymbol{h}_1^e &= f(\boldsymbol{W}_1^e \boldsymbol{z} + \boldsymbol{b}_1^e) \\ \boldsymbol{h}_i^e &= f(\boldsymbol{W}_i^e \boldsymbol{h}_{i-1}^e + \boldsymbol{b}_i^e), i = 2, ..., k \end{aligned} \quad (13)$$

where f is the activation function, i.e., the Sigmoid function. $\boldsymbol{W}_i^e$ and $\boldsymbol{b}_i^e$ represent the transformation matrix and the bias vector of the i-th layer, respectively. k represents the number of hidden layers. The output of $(i-1)$-th layer is fed into the i-th hidden layer. For any arbitrary real value vector, we use the Sigmoid function to map it to range $[0, 1]$.

The ultimate goal of edge feature learning is minimizing the reconstruction error between the final feature representation and input. The reconstruction loss is computed as:

$$L_e = ||\boldsymbol{Z} - \boldsymbol{H}^e||_F^2. \quad (14)$$

By minimizing the distances between the reconstructed representations and the original input, we can get the final edge feature representations.

D. Anomalous Citation Identification

Now we have obtained node representations and edge representations through the process of feature learning. As mentioned previously, we formulate the task as a binary classification task. Accordingly, we need to train a supervised classifier to predict the label for each paper pair. Specifically, for each unlabeled paper pair $e = (p_i, p_j)$, the final output should be its label (0 or 1). We adopt Logistic Regression (LR) as the classifier. The loss function L_{lr} of LR is:

$$L_{lr} = |L - \sigma(\boldsymbol{H} \cdot \boldsymbol{W}^l + \boldsymbol{B}^l)| \quad (15)$$

where $\boldsymbol{H}$ connects nodes representations and edge representations. $\boldsymbol{W}^l$ and $\boldsymbol{B}^l$ refer to the weight matrix and bias matrix, respectively. σ is the Sigmoid activation function. For the first-order gradient-based optimization, we use the Adam method.

E. Overall Reconstruction

To achieve better performance, we jointly train node feature learning, edge feature learning, and classification processes. The joint loss function L_{sum} is computed as:

$$\begin{aligned} L_{sum} =& L_n + L_e + \beta L_{lr} + \alpha(\sum_i (||\boldsymbol{W}_i^e||_F^2 + ||\boldsymbol{B}_i^e||_F^2) \\ &+ \sum_l ||\boldsymbol{W}_l^n||_F^2) \end{aligned} \quad (16)$$

where α and β are two hyperparameters to adjust the weights of each part. Our goal is to minimize L_{sum}. For this purpose, we use back propagation algorithm (BP) with stochastic gradient descent to train the overall framework. The overall architecture of GLAD is summarized in Algorithm 1.

Algorithm 1 Anomalous Citation Detection with GLAD

Input: $\boldsymbol{A}$, $\boldsymbol{X}$, $\boldsymbol{Z}$, α, β, r, L, and the convergence condition ϵ.

Output: l for each $e = (i, j)$

1: **Initiate**: randomly initiate $\theta_1 \leftarrow \{\boldsymbol{W}_i^e, \boldsymbol{B}_i^e\}_{i=1}^m$, $\theta_2 \leftarrow \{\boldsymbol{W}_l^n\}$, and $\theta_3 \leftarrow \{\boldsymbol{W}^l, \boldsymbol{B}^l\}$.
2: **Pre-train**: set r, conduct edge feature learning process and node feature learning process independently.
3: **Train**:
4: **while** ϵ is true **do**
5:     randomly generate a batch of data from edge features and local node features
6:     calculate L_n based on Eq.(7), get reconstructed $\boldsymbol{H}^n$
7:     calculate L_e based on Eq.(14), get reconstructed $\boldsymbol{H}^e$
8:     concatenate $\boldsymbol{H} \leftarrow [\boldsymbol{H}^n, \boldsymbol{H}^e]$
9:     calculate L_{sum} based on Eq.(16)
10:     update $\theta_1, \theta_2, \theta_3$
11: **end while**

F. Model Complexity Analysis

The overall framework contains three critical parts, i.e., node feature learning, edge feature learning, and classification. The node representation $\boldsymbol{H}^n$ is generating by maximizing mutual information in an unsupervised manner, whose training complexity is $O(c_n n_n I_n d_n (N_p + N_n))$, where N_p, N_n represent the number of positive samples and negative samples, respectively. c_n is the sample size (the number of nodes $|V|$). I_n represents the iteration times. n_n is the dimension of node features and d_n is the layer dimension. The edge embedding $\boldsymbol{H}^e$ is generated through a deep autoencoder. The complexity of generating the above embedding matrix is $O(c_e n_e d_e I_e)$, where c_e is the size of samples (the number of edges $|E|$), n_e is the dimension of edge features, d_e is the maximum dimension of the hidden layer, and I_e represents the iteration times. The classification process adopts Logistic Regression and the training complexity is $O(n * k)$, where n represents the sample size and k is the feature dimension, which is related to the embedding dimensions. Thus, the overall training complexity of GLAD is $O(c_n n_n I_n d_n (N_p + N_n) + c_e n_e d_e I_e + I_l n * k)$ which is approximate to $O(I_n |V| d_n + I_e |E| d_e + I_l k |E|)$.
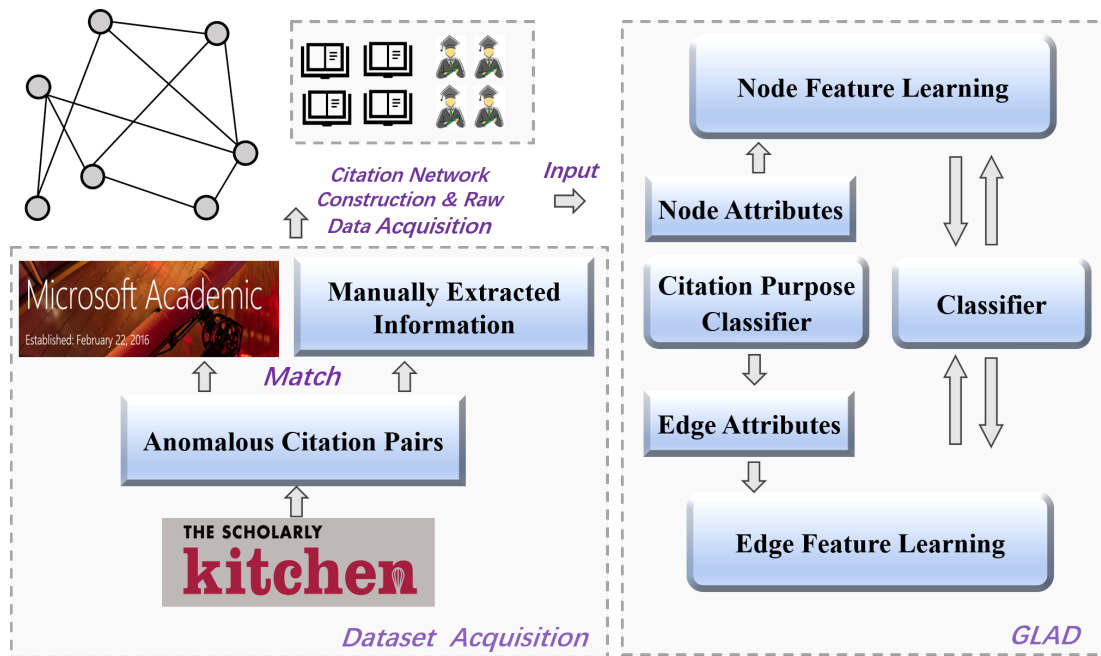
Fig. 3. Experiment procedure.

V. EXPERIMENTS

In this section, we will introduce the datasets including citation purpose annotation, ground-truth anomalous citations, and the constructed citation network. We validate the effectiveness of GLAD by conducting a series of experiments with discussions of the experimental results. Fig. 3 presents the overall experiment procedure.

A. Datasets

1) Citation Purpose Annotation: The annotation process for citation context is conducted by postgraduate students with a good background in natural language processing. We ask them to determine the purpose of citing the target reference by choosing from the six purpose categories described in TABLE III. To estimate inter-annotator agreement, each citation context is assigned to three different annotators. Kappa Coefficient [51] is used to measure the agreement:

$$K = \frac{P(A) - P(E)}{1 - P(E)} \quad (17)$$

where $P(A)$ represents the relative observed agreement among three annotators and $P(E)$ represents the hypothetical probability of agreement.

In our task, the agreement among three annotators are $K_{12} = 0.606$, $K_{13} = 0.477$, and $K_{23} = 0.677$, respectively, which indicates moderate and substantial agreements. Besides, uncertain labeled results are discarded. The distribution of the purpose categories are shown in TABLE V.

2) Anomalous Citation Dataset: For anomalous citation detection in academic networks, ground-truth anomalous citations are not clearly defined. Hence, we artificially create a dataset motivated by previous research [52]. The dataset generation process is described as follows in detail.

TABLE V
THE PROPORTION OF CITATION PURPOSE IN ANNOTATION

Categories	Number	Proportion
Criticizing	150	14%
Comparison	98	9%
Use	183	17%
Substantiating	43	4%
Basis	323	30%
Neutral/ Other	280	26%

We first collect the paper titles which are reported as papers causing "journal cartels" in [53]. Most of these papers are published in donor journals including the *Medical Science Monitor*, and *The Scientific World Journal*. These papers cite papers published in *Cell Transplantation* at an excessively high rate. For example, the paper titled "Regenerative medicine for neurological disorders" published in *The Scientific World Journal* has cited 124 papers in total, 96 of which are published in *Cell Transplantation*. Two of the four authors of that paper are editors of *Cell Transplantation*. The impact factor of *Cell Transplantation* has dropped by removing the reference of that paper. We also crawl the citation cartels detection results from previous research [52]. Taking the paper "Crystal structure refinement with SHELXL?" published in *Acta Crystallographica C* as an example. The paper has received 594 citations from the donor journals *IUCrData* and *Acta Crystallographica Section E*, accounting for 94% of the total citations in 2017. Removing the citations from donor journals decreases the JIF of *Acta Crystallographica C* by 22%. Details of the anomalous citations dataset are described in TABLE VI.

3) Microsoft Academic Search Dataset: The citation relationships of anomalous papers are obtained from a widely used

TABLE VI
DETAILS OF ANOMALOUS DATASETS

Paper Title	Publishing Journal	Anomalous Citation Relationship	Year	State
Stem cells have the potential to rejuvenate regenerative medicine research	*Medical Science Monitor*	References to papers published in *Cell Transplantation* and *Medical Science*	2008-2009	Retracted
Regenerative medicine for neurological disorders	*The Scientific World Journal*	References to papers published in *Cell Transplantation* and *The Scientific World Journal*	2008-2009	Retracted
The continued promise of stem cell therapy in regenerative medicine	*Medical Science Monitor*	References to papers published in *Cell Transplantation* and *Medical Science Monitor*	2009-2010	Retracted
A showcase of bench-to-bedside regenerative medicine at the 2010 ASNTR	*The Scientific World Journal*	References to papers published in *Cell Transplantation* and *The Scientific World Journal*	2010-2011	Retracted
Technology and innovation: 2010 a year in review	*Cell Transplantation*	References to papers published in *Technology and Innovation* and *Cell Transplantation*	2009-2010	Retracted
Crystal structure refinement with SHELXL?	*Acta Crystallographica Section C*	Citations from *IUCrData* and *Acta Crystallographica Section E*	2017	Normal
The SCARE 2018 statement: Updating consensus Surgical CAse REport (SCARE) guidelines	*International Journal of Surgery*	Citations from *Annals of Medicine and Surgery* and *International Journal of Surgery Case Reports*	2019	Normal
Regulatory Intermediaries in the age of Governance	*Annals of American Academy of Political and Social Science*	Citations from Annals of *Medicine and Surgery* and *International Journal of Regulation and Governance*	2019	Normal

bibliographic dataset, Microsoft Academic Graph (MAG)[3]. It contains abundant information for each paper such as journal name, publication year, authors, abstract, citation relationship, and so on. In particular, it provides citation context, that is, the raw text where the citation is mentioned in the paper. We use citation contexts to conduct citation purpose classification process as mentioned in Section IV-B. We match the anomalous papers mentioned in TABLE VI and construct the citation network centered on them. Edges for anomalous citations are labeled as anomalous, and the rest are normal. The features of papers (such as abstract) and citation relationships (such as citation context) can also be obtained and calculated in this process. It's important to note that the dataset doesn't contain some retracted papers. We searched for the corresponding papers on the Internet and then added them back manually. Finally, the constructed network contains 4718 citing-cited relationships, of which 300 citations are anomalous.

B. Evaluation Metrics

In this paper, the task of anomalous citation identification is formulated as a binary classification problem, i.e., for each paper pair (p_i, p_j), whether (p_i, p_j) is anomalous (yes or no). Therefore, we adopt four widely used metrics for classification tasks, including Accuracy, Precision, Recall, and F1-score to validate the performance of the model. Specifically, before the task of anomalous citation identification, we divide the training set into two categories:

- Positive sample: The positive sample set P contains all real anomalous paper pairs (labeled as 1);
- Negative sample: The negative sample set N contains all sample pairs who are not anomalous paper pairs (labeled as 0).

After the classification process, we assume that set TP includes all pairs in P that are correctly classified as anomalous citations, and FN are misclassified pairs in P. In the same way, the set FP involves the misjudged pairs in N and TN is the correctly judged set of N. Apparently, $N = FP + TN$. Hence, the indicators can be calculated as follows:

- $Accuracy = \dfrac{|TP| + |TN|}{|P| + |N|}$
- $Precision = \dfrac{|TP|}{|TP| + |FP|}$
- $Recall = \dfrac{|TP|}{|TP| + |FN|}$
- $F1 - score = \dfrac{2 * Precision * Recall}{Precision + Recall}$.

Furthermore, we also utilize ROC-AUC (AUC) to evaluate the overall performance of all methods. AUC is calculated based on the relative ranking of the predicted probabilities of all instances.

C. Baselines

Due to the lack of any existing baseline for the given task, we compare GLAD with the following state-of-the-art models of graph neural networks because our feature learning process is built on graph neural networks.

- **DeepWalk [54]**. Deepwalk uses the co-occurrence relationship between nodes in the network to learn node representations. The latent representations are generated by employing skip-gram.
- **Node2vec [55]**. Node2vec obtains node sequences through random walk in terms of both DFS and BFS. Hence the low-dimensional feature representations can comprehensively preserve the similarity of DFS and BFS neighborhoods.
- **LINE [56]**. LINE is also a graph embedding method based on the assumption of neighborhood similarity.

TABLE VII
PERFORMANCE COMPARISON OF DIFFERENT ANOMALY DETECTION METHODS BASED ON GRAPH LEARNING ON THE TASK OF ANOMALOUS CITATION
DETECTION

Methods \ Metrics	AUC-ROC	Accuracy	Precision	Recall	F1-score
GDN	0.7431±0.015	0.6254±0.0054	0.5731±0.0038	0.7315±0.0043	0.6313±0.0037
Meta-GDN	0.7653±0.013	0.6233±0.0022	0.6152±0.0025	0.7506±0.0112	0.6579±0.0046
OC-GCN	0.6531±0.011	0.5437±0.0036	0.5229±0.0078	0.8237±0.0057	0.6312±0.0044
OC-GAT	0.8115±0.012	0.8431±0.0012	0.8345±0.0023	0.8441±0.0011	0.8564±0.0034
OC-SAGE	0.8652±0.012	0.8646±0.0065	0.8843±0.0037	0.8542±0.0036	0.8465±0.0045
CARE-Att	0.8935±0.022	0.8754±0.0012	0.9012±0.0122	0.8534±0.0014	0.8653±0.0043
CARE-Weight	0.8963±0.014	0.8644±-0.0025	**0.8754**±0.0017	0.8743±0.0034	0.8853±0.0055
CARE-Mean	0.8812±0.012	0.8623±0.0067	0.8671±0.0056	0.8586±0.0049	0.8785±0.0027
CARE-GNN	**0.8983**±0.012	0.8825±0.0034	0.8945±0.0015	0.8432±0.0026	0.8654±0.0033
GLAD	0.8953±0.0014	**0.913**±0.0087	**0.9683**±0.0054	0.854±0.0109	**0.9075**±0.0090

LINE uses BFS to construct neighborhoods. In addition, LINE can also be applied in weighted graphs.

- **GraphSAGE [26].** GraphSAGE is an inductive framework, which can use vertex feature information (such as text attributes) to efficiently generate embedding for nodes.
- **DGI [30].** DGI learns node representations by maximizing mutual information between the graph-level summary representation and the local patches. DGI is an unsupervised network representation learning framework.
- **MLP**. MLP (Multi-layer perceptron) neural network is a common ANN (artificial neural network) algorithm, which consists of an input layer, an output layer, and one or more hidden layers. Different layers of MLP neural network are fully connected.

The comparison results are presented in Fig. 4. We also compare GLAD with state-of-the art methods for anomaly detection based on graph learning:

- **GDN [57]**. GDN is a new GNN architecture designed for network anomaly detection with limited labeled data. Specifically, the authors also propose Meta-GDN, which integrates cross-network meta-learning to detect anomalies with few labeled instances.
- **OCGNN [58]**. OCGNN is a one-class classification framework that contains a series of algorithms for graph anomaly detection. OCGNN combines the powerful representation ability of GNNs along with the classical one-class objective. In this work, we compare GLAD with OC-GCN, OC-GAT, and OC-SAGE.
- **CARE [59]**. CARE enhances the GNN aggregation process with three unique modules against camouflages including a label-aware similarity measure, a similarity-aware neighbor selector, and a relation-aware neighbor aggregator. In this work, we compare GLAD with various forms of CARE, including CARE-Att, CARE-Weight, CARE-Mean, and CARE-GNN.

Table VII lists the results of comparing GLAD with baselines. For the sake of fair comparisons with GLAD which considers both node attributes and edge attributes, we concatenate the raw edge attribute matrix Z to learned node embeddings of the methods that ignore the edge attributes such as Deepwalk, LINE, Node2vec, and DGI, i.e., $H \leftarrow [H^n; Z]$. Furthermore, we also compare GLAD with different variants to examine the learning efficacy of each module and eliminate the effect of each module. TABLE VIII presents the details of each variant. First, we only utilize node feature learning process mentioned in Section IV-A (GLAD-N) and edge feature learning mentioned in Section IV-C (GLAD-E) to detect anomalous citations. To examine whether node and edge features are useful, we use random vectors to replace the node features in the process of node feature learning (GLAD-RN) and the edge features in the process of edge feature learning (GLAD-RE). We then remove the loss function of training nodes features L_n (GLAD-EL) and the loss function of training edge features L_e in the final loss function (GLAD-NL) to observe the model performance without auxiliary training objective. Finally, we also use the edge feature learning module (based on the autoendoder) to learn the node features and connect the embeddings with edge representations as the final input (GLAD-EE).

D. Results and Analysis

The experimental results are presented together with a number of case studies from the following three perspectives, including (1) comparison against baselines, (2) comparison of GLAD variants, and (3) parameter sensitivity.

1) Comparison against Baselines: The results of comparing GLAD with all baselines mentioned in Section V-C are shown in Fig. 4. From the results, we can observe that Our method GLAD outperforms all the baselines in terms of Accuracy, Precision, and F1-score, which shows the effectiveness of GLAD. Although the results of the baselines on Recall are higher than the proposed method, their Precisions are lower. Higher Precisions indicate that when we use the model to determine that a citation is an anomalous citation, then it is highly likely to be anomalous. We don't expect to find all anomalous citations but hope that the identified anomalous citations are as accurate as possible. In addition,

TABLE VIII
SUMMARY OF VARIANTS FOR COMPARISON

Variants of GLAD	Node Feature Learning	Edge Feature Learning	Random Vectors for Node Features	Random Vectors for Edge Features	Loss Function for Node Learning	Loss Function for Edge Learning	Edge Learning Module for Node Representation
GLAD-N	√				√	√	
GLAD-E		√			√	√	
GLAD-RN	√	√	√		√	√	
GLAD-RE	√	√		√	√	√	
GLAD-EL	√	√				√	
GLAD-NL	√	√			√		
GLAD-EE	√	√			√	√	√
GLAD	√	√			√	√	

TABLE IX
CITATION PURPOSE CLASSIFICATION RESULTS OF CPU USING DIFFERENT CLASSIFIERS

Methods	Metrics / Categories	Criticizing	Comparison	Use	Substantiating	Basis	Neutral/Other
SVM	Precision	64.30%	58.10%	57.90%	59.90%	45.70%	54.60%
	Recall	79.20%	51.70%	67.30%	64.10%	44.60%	77.30%
	F1-score	70.70%	56.30%	66.40%	60.90%	44.90%	67.60%
Logistic Regression	Precision	61.10%	49.90%	66.80%	52.10%	46.70%	52.30%
	Recall	68.40%	55.60%	52.50%	43.50%	44.10%	65.40%
	F1-score	63.30%	51.90%	55.80%	47.50%	44.30%	55.10%
Naive Bayes	Precision	67.70%	59.90%	33.20%	50.60%	55.30%	49.90%
	Recall	69.00%	56.40%	45.90%	66.70%	42.10%	54.20%
	F1-score	68.10%	57.00%	39.60%	59.40%	48.30%	52.60%

TABLE X
THE PERFORMANCE OF GLAD WITH DIFFERENT BATCH SIZE

Metrics / Batch Size	1	4	8	16	32	64
Accuracy	0.8958 ± 0.009	$\mathbf{0.8965} \pm 0.0098$	0.8943 ± 0.0099	0.8953 ± 0.0098	0.8958 ± 0.0098	0.8954 ± 0.0098
Precision	0.9547 ± 0.006	$\mathbf{0.9552} \pm 0.0066$	0.9522 ± 0.0068	0.9542 ± 0.0067	$\mathbf{0.9552} \pm 0.0066$	0.9545 ± 0.0067
Recall	0.8309 ± 0.012	$\mathbf{0.8884} \pm 0.010$	0.8871 ± 0.0102	0.8305 ± 0.0121	0.8305 ± 0.0121	0.8305 ± 0.0121
F1-score	0.8885 ± 0.0101	$\mathbf{0.8957} \pm 0.0098$	0.8943 ± 0.0099	0.8881 ± 0.010	0.8885 ± 0.0101	0.8882 ± 0.0101

GLAD has achieved the best results on the comprehensive indicator F1-score. In all the comparison methods, DGI and GraphSAGE have better performances, which illustrates that learning attribute features is more effective in the task of anomalous citation detection. The comparison between graph neural networks and MLP shows that feature learning based on neural networks can improve the performance of the model. Compared with directly inputting features, learning features first can effectively improve the performance of anomalous citation detection.

The results of comparing GLAD with all baselines for anomaly detection mentioned in Section V-C are presented in TABLE VII. Among all GNN baselines in TABLE VII, in terms of Accuracy, Precision, and F1-score, our approach outperforms all the other compared methods by a significant margin. It demonstrates that GLAD can better detect abnormal edges than other methods. GLAD is capable of extracting comprehensive knowledge across the citation network, which further enhances the detection performance on the target network. Besides, experimental results have shown that the
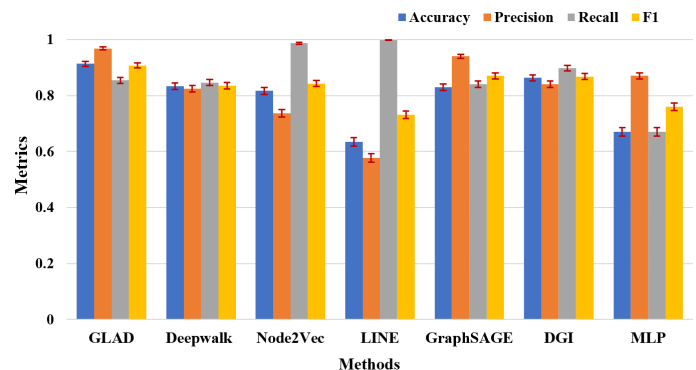

Fig. 4. Performance comparison of different methods on the task of anomalous citation detection.

proposed model performs better than OC-based model. One-class classification based methods attempt to model normal examples and classify new examples as either normal or abnormal. In this paper, we aim to detect anomalous citations through analyzing the contexts, and give an explicit definition

of anomalous citation shown in Section III. In real-world scenarios, there are different kinds of citation purposes between two papers. In other words, it is impossible to learn all features of normal citations due to the complexity and diversity of normal samples. Therefore, the one-class classifier is not suitable for detecting anomalous citations.

2) Comparison of GLAD Variants: A key ingredient of GLAD is the feature learning of both node attributes and edge attributes. To validate the effectiveness of this mechanism, here we examine the performance of GLAD with only node feature learning process or with only edge feature learning process. From Fig. 5, From the figure, we can clearly observe that GLAD is 36.2% and 44.4% higher than GLAD-N and GLAD-E respectively on Precision. GLAD is 21.7% and 26.3% higher than GLAD-N and GLAD-E in terms of F1-score, respectively. In general, the performance of GLAD is better than GLAD-N and GLAD-E, which suggests that utilizing both node and edge attributes in feature learning processes is necessary, and GLAD can facilitate them well to help detect anomalous citations. Besides, if we use random vectors for feature learning (GLAD-RN and GLAD-RE), the performances will be better than that of GLAD-N and GLAD-E on F1-score, but there is still a certain gap compared with GLAD. The proposed features are more effective than random vectors. If we do not minimize the loss function of feature learning (but still use these features), the performance of the models will be further improved, but will not exceed GLAD. Without the auxiliary training objectives, the model performs worse. Finally, using the autoencoder to learn the node features and edge features at the same time (GLAD-EE), the best results are obtained in all variants. This shows the necessity of feature learning for both edges and nodes. However, in the task of anomalous citation detection, the performance of node learning with autoencoder is not as good as that of DGI. In conclusion, although some indicators of some variants exceed GLAD, the overall performance of GLAD is better than that of all variants, which indicates that the proposed features are effective and it is necessary to conduct feature learning processes.
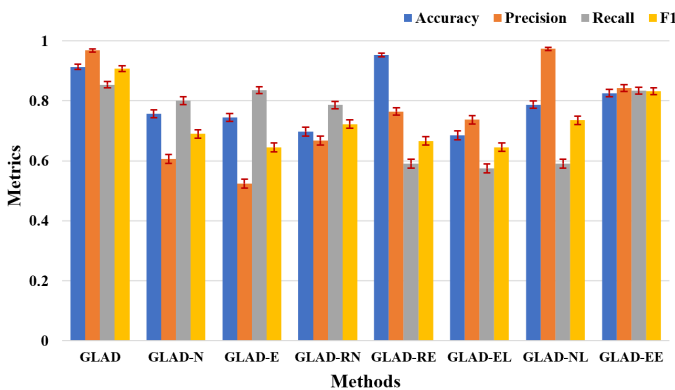

Fig. 5. Performance comparison of different variants on the task of anomalous citation detection.

3) Parameter Sensitivity: We conduct a series of experiments to examine how different parameters influence the

performance of GLAD. We report the performance of GLAD with respect to the following parameter settings: 1) batch size; 2) embedding dimension; 3) learning rate; 4) number of hidden layers and units in the autoencoder. For each parameter setting, we fix other parameters and try 10 different splits of the training and testing. The results are reported as the average performance of the 10 experiments. The results also include confidence intervals of each index.

First of all, TABLE IX presents the citation purpose classification results of CPU using different classifiers including SVM, Logistic Regression, and Naive Bayes. By comparing the results of experiments, we can find that the SVM performs better in the task of citation purposes classification. Besides, one of the edge features *CP* refers to that whether the purpose of the citation is clear or not. In this paper, we also try binary classification. The classification precision can be improved to 78.4%.

Batch Size. TABLE X lists the experimental results of GLAD's performance when hyperparameter batch size is set doubled increasing from 1 to 64. We can observe that the performance of GLAD is relatively stable within a large range of batch sizes. In addition, the performance of GLAD achieves the best performance in Accuracy, Precision, F1-score, and Recall when the batch size is set to 4. This observation demonstrates that larger batch sizes dont necessarily lead to better performance.

Embedding Dimension. We empirically evaluate the impact of edge embedding dimension. TABLE XI illustrates the performance of GLAD when altering the edge embedding dimension from 32 to 512. We can conclude that the performance of GLAD slightly fluctuates within a large range of edge embedding dimensions, and the performance drops when the edge embedding dimension is too small. The model can reach a relatively better state when the embedding dimension is set to 512.

Learning Rate. Learning rate controls the update speed of GLAD. TABLE XII shows how different values of the learning rate influence the performance of GLAD. The results show that the performance drop when the learning rate is either too large or too small. We can see that the proposed model performs best in Accuracy, Precision, and F1-score when the learning rate is set to 0.01. Besides, the value of Recall is relatively stable despite the learning rate varies from 0.001 to 0.1.

Number of Hidden Layers. To investigate the impact of different sizes of neural networks on the performance of GLAD, we set 8 different experimental groups by adjusting the numbers of hidden layers and hidden units in the process of edge feature learning. The detailed settings of the 8 groups can be seen in TABLE XIII. According to TABLE XIV, we can conclude that GLAD performs better when the number of hidden layers is set to 2 (Group 5). Furthermore, increasing the number of hidden layers does not always improve the metrics.

VI. CONCLUSION

In this work, we aim to solve the problem of anomaly detection in academic citation networks. For this purpose, we develop a novel deep graph learning framework, namely

TABLE XI
THE PERFORMANCE OF GLAD WITH DIFFERENT EMBEDDING DIMENSION

Metrics Embedding Dimension	32	64	128	256	512
Accuracy	$0.8913 \pm$ 0.0150	$0.8947 \pm$ 0.0148	$0.8967 \pm$ 0.0146	$0.8965 \pm$ 0.0146	$\mathbf{0.897} \pm$ 0.0146
Precision	0.96 ± 0.0095	$\mathbf{0.9641} \pm$ 0.0090	0.96 ± 0.0094	$0.9627 \pm$ 0.0091	$0.9622 \pm$ 0.0091
Recall	$0.8165 \pm$ 0.0187	0.82 ± 0.0185	$\mathbf{0.828} \pm$ 0.0181	$0.825 \pm$ 0.0183	$0.8265 \pm$ 0.0182
F1-score	$0.8825 \pm$ 0.0156	$0.8862 \pm$ 0.0153	$0.8891 \pm$ 0.0151	$0.8885 \pm$ 0.0151	$\mathbf{0.8892} \pm$ 0.0151

TABLE XII
THE PERFORMANCE OF GLAD WITH DIFFERENT LEARNING RATE

Metrics Learning Rate	0.001	0.005	0.01	0.1
Accuracy	0.872 ± 0.0162	0.8727 ± 0.0162	$\mathbf{0.874} \pm 0.0161$	0.8727 ± 0.0162
Precision	0.9232 ± 0.0129	0.9258 ± 0.0127	$\mathbf{0.9279} \pm 0.0125$	0.9248 ± 0.0128
Recall	$\mathbf{0.8115} \pm 0.0190$	0.8105 ± 0.0190	0.811 ± 0.0190	$\mathbf{0.8115} \pm 0.0190$
F1-score	0.8638 ± 0.0166	0.8643 ± 0.0166	$\mathbf{0.8655} \pm 0.0166$	0.8644 ± 0.0166

TABLE XIII
NUMBER OF HIDDEN UNITS IN EACH LAYER FOR AUTOENCODER

Group	No. layers	No. units
1	1	8
2	2	8,6
3	3	8,6,4
4	4	8,6,4,2
5	2	8,10
6	3	8,10,20
7	4	8,10,20,30
8	5	8,10,20,30,40

GLAD, to model and identify anomalous citations. We propose to incorporate the node feature learning process and edge feature learning process into the framework. Thus GLAD can make full use of information from node attributes and network structures. Particularly, we propose an effective algorithm to capture semantic information of citation contexts. We conduct comprehensive experiments to validate the performance of GLAD. Experimental results demonstrate the effectiveness of the proposed anomalous citation detection framework as well as the proposed citation purpose classification algorithm.

However, we only consider node attributes and connection information (adjacency matrix) in the node feature learning process. More network structural information such as preferential attachment and Admaic-Adar index could be incorporated. Besides, the construction of anomalous datasets and the labeling of citation purpose are time-consuming. It would be an interesting topic to generate such ground-truth data for further research.

ACKNOWLEDGMENTS

REFERENCES

[1] S. Fortunato, C. T. Bergstrom, K. Börner, J. A. Evans, D. Helbing, S. Milojević, A. M. Petersen, F. Radicchi, R. Sinatra, B. Uzzi *et al.*, "Science of science," *Science*, vol. 359, no. 6379, 2018.

[2] X. Kong, Y. Shi, S. Yu, J. Liu, and F. Xia, "Academic social networks: Modeling, analysis, mining and applications," *Journal of Network and Computer Applications*, vol. 132, pp. 86–103, 2019.

[3] W. Wang, J. Liu, T. Tang, S. Tuarob, F. Xia, Z. Gong, and I. King, "Attributed collaboration network embedding for academic relationship mining," *ACM Transactions on the Web (TWEB)*, vol. 15, no. 1, pp. 1–20, 2020.

[4] X. Bai, H. Pan, J. Hou, T. Guo, I. Lee, and F. Xia, "Quantifying success in science: An overview," *IEEE Access*, vol. 8, pp. 123 200–123 214, 2020.

[5] C. G. Prabha, "Some aspects of citation behavior: A pilot study in business administration," *Journal of the American Society for Information Science*, vol. 34, no. 3, pp. 202–206, 1983.

[6] M. Dong, B. Zheng, N. Quoc Viet Hung, H. Su, and G. Li, "Multiple rumor source detection with graph convolutional networks," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 569–578.

[7] Y.-J. Lu and C.-T. Li, "Gcan: Graph-aware co-attention networks for explainable fake news detection on social media," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 505–514.

[8] T. Bian, X. Xiao, T. Xu, P. Zhao, W. Huang, Y. Rong, and J. Huang, "Rumor detection on social media with bi-directional graph convolutional networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 549–556.

[9] A. Li, Z. Qin, R. Liu, Y. Yang, and D. Li, "Spam review detection with graph convolutional networks," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 2703–2711.

[10] S. Yu, F. Xia, Y. Sun, T. Tang, X. Yan, and I. Lee, "Detecting outlier patterns with query-based artificially generated searching conditions," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 1, pp. 134–147, 2020.

TABLE XIV
THE PERFORMANCE OF GLAD WITH DIFFERENT HIDDEN LAYERS AND HIDDEN UNITS IN EDGE FEATURE LEARNING

Group / Metrics	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	Group 8
Accuracy	0.7212 ± 0.021	0.896 ± 0.0147	0.8717 ± 0.0162	0.8777 ± 0.0156	**0.8992** ± 0.0144	0.8945 ± 0.0146	0.8702 ±0.0164	0.844 ± 0.0183
Precision	0.6759 ± 0.0222	0.961 ± 0.0093	0.9236 ± 0.0129	0.9104 ± 0.0136	**0.9645** ± 0.0089	0.9433 ± 0.0110	0.9308 +- 0.0124	0.9221± 0.0135
Recall	**0.85** ± 0.0169	0.8255 ± 0.0183	0.8105 ± 0.0190	0.838 ± 0.0176	0.829 ± 0.0181	0.8395 ± 0.0175	0.8 +- 0.0196	0.7515 ± 0.02187
F1-score	0.753 ± 0.0205	0.8881 ± 0.0152	0.8634 ± 0.0167	0.8727 ± 0.0159	**0.8916** ± 0.0149	0.8884 ± 0.0150	0.8604 ± 0.0169	0.8281 ± 0.0190

[11] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection," *Information Sciences*, vol. 479, pp. 448–455, 2019.

[12] F. Xia, W. Wang, T. M. Bekele, and H. Liu, "Big scholarly data: A survey," *IEEE Transactions on Big Data*, vol. 3, no. 1, pp. 18–35, 2017.

[13] X. Kong, J. Zhang, D. Zhang, Y. Bu, Y. Ding, and F. Xia, "The gene of scientific success," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 14, no. 4, pp. 1–19, 2020.

[14] K. Moustafa, "Aberration of the citation," *Accountability in Research*, vol. 23, no. 4, pp. 230–244, 2016.

[15] C. Bartneck and S. Kokkelmans, "Detecting h-index manipulation through self-citation analysis," *Scientometrics*, vol. 87, no. 1, pp. 85–98, 2011.

[16] Z. Corbyn, "An easy way to boost a paper's citations," *Nature*, 2010.

[17] M. Mimouni, M. Ratmansky, Y. Sacher, S. Aharoni, and A. Mimouni-Bloch, "Self-citation rate and impact factor in pediatrics," *Scientometrics*, vol. 108, no. 3, pp. 1455–1460, 2016.

[18] J. P. Ioannidis, "A generalized view of self-citation: Direct, co-author, collaborative, and coercive induced self-citation," *Journal of Psychosomatic Research*, vol. 78, no. 1, pp. 7–11, 2015.

[19] G. Franck, "Scientific communication–a vanity fair?" *Science*, vol. 286, no. 5437, pp. 53–55, 1999.

[20] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2. IEEE, 2005, pp. 729–734.

[21] F. Xia, K. Sun, S. Yu, A. Aziz, L. Wan, S. Pan, and H. Liu, "Graph learning: A survey," *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 2, pp. 109–127, 2021.

[22] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2008.

[23] C. Gallicchio and A. Micheli, "Graph echo state networks," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2010, pp. 1–8.

[24] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," in *4th International Conference on Learning Representations, ICLR*, 2016.

[25] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 1993–2001.

[26] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, 2017, pp. 1024–1034.

[27] J. Chen, T. Ma, and C. Xiao, "Fastgcn: Fast learning with graph convolutional networks via importance sampling," in *International Conference on Learning Representations*, 2018.

[28] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *International Conference on Learning Representations*, 2019.

[29] L. Wu, Z. Li, H. Zhao, Q. Liu, J. Wang, M. Zhang, and E. Chen, "Learning the implicit semantic representation on graph-structured data," in *Database Systems for Advanced Applications - 26th International Conference, DASFAA*, vol. 12681, 2021, pp. 3–19.

[30] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax." in *Seventh International Conference on Learning Representations*, 2019.

[31] Y. Ren and B. Liu, "Heterogeneous deep graph infomax," in *Workshop of Deep Learning on Graphs: Methodologies and Applications co-located with the Thirty-Fourth AAAI Confer-ence on Artificial Intelligence*, 2020.

[32] K. Tu, P. Cui, X. Wang, P. S. Yu, and W. Zhu, "Deep recursive network embedding with regular equivalence," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2357–2366.

[33] W. Yu, C. Zheng, W. Cheng, C. C. Aggarwal, D. Song, B. Zong, H. Chen, and W. Wang, "Learning deep network representations with adversarially regularized autoencoders," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2663–2671.

[34] T. Ma, J. Chen, and C. Xiao, "Constrained generation of semantically valid graphs via regularizing variational autoencoders," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 7113–7124.

[35] A. Bojchevski, O. Shchur, D. Zügner, and S. Günnemann, "Netgan: Generating graphs via random walks," in *International Conference on Machine Learning*. PMLR, 2018, pp. 610–619.

[36] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D.-Y. Yeung, "Gaan: Gated attention networks for learning on large and spatiotemporal graphs," *Conference on Uncertainty in Artificial Intelligence*, 2018.

[37] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 922–929.

[38] D. Yu, R. Zhang, Z. Jiang, Y. Wu, and Y. Yang, "Graph-revised convolutional network," vol. 12459, 2020, pp. 378–393.

[39] T. Zhao, Y. Liu, L. Neves, O. Woodford, M. Jiang, and N. Shah, "Data augmentation for graph neural networks," *The Thirty-Fifth AAAI Conference on Artificial Intelligence*, 2021.

[40] X. Zhang and M. Zitnik, "Gnnguard: Defending graph neural networks against adversarial attacks," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[41] X. Wang, M. Zhu, D. Bo, P. Cui, C. Shi, and J. Pei, "Am-gcn: Adaptive multi-channel graph convolutional networks," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1243–1253.

[42] J. Liu, F. Xia, L. Wang, B. Xu, X. Kong, H. Tong, and I. King, "Shifu2: A network representation learning based model for advisor-advisee relationship mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, pp. 1763–1777, 2021.

[43] L. Wang, J. Ren, B. Xu, J. Li, W. Luo, and F. Xia, "Model: Motif-based deep feature learning for link prediction," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 2, pp. 503–516, 2020.

[44] L. Ouyang, Y. Zhang, and Y. Wang, "Unified graph embedding-based anomalous edge detection," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[45] L. Zheng, Z. Li, J. Li, Z. Li, and J. Gao, "Addgraph: Anomaly detection in dynamic graph using attention-based temporal gcn." in *IJCAI*, 2019, pp. 4419–4425.

[46] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International Conference on Machine Learning*, 2014, pp. 1188–1196.

[47] A. Abu-Jbara, J. Ezra, and D. Radev, "Purpose and polarity of citation: Towards nlp-based bibliometrics," in *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies*, 2013, pp. 596–606.

[48] I. Beltagy, K. Lo, and A. Cohan, "Scibert: A pretrained language model for scientific text," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3606–3611.

[49] M. Schuster and K. Nakajima, "Japanese and korean voice search," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 5149–5152.

[50] C. Tu, Z. Zhang, Z. Liu, and M. Sun, "Transnet: Translation-based network representation learning for social relation extraction." in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017, pp. 2864–2870.

[51] J. Cohen, "Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit." *Psychological Bulletin*, vol. 70, no. 4, p. 213, 1968.

[52] S. Kojaku, G. Livan, and N. Masuda, "Detecting citation cartels in journal networks," *arXiv preprint arXiv:2009.09097*, 2020.

[53] P. Davis, "The emergence of a citation cartel," *The Scholarly Kitchen*, vol. 10, pp. 15–17, 2012.

[54] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 701–710.

[55] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 855–864.

[56] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web*, 2015, pp. 1067–1077.

[57] K. Ding, Q. Zhou, H. Tong, and H. Liu, "Few-shot network anomaly detection via cross-network meta-learning," in *Proceedings of the Web Conference 2021*, 2021, pp. 2448–2456.

[58] X. Wang, B. Jin, Y. Du, P. Cui, Y. Tan, and Y. Yang, "One-class graph neural networks for anomaly detection in attributed networks," *Neural Computing and Applications*, vol. 33, pp. 12 073–12 085, 2021.

[59] 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*, 2020, pp. 315–324.
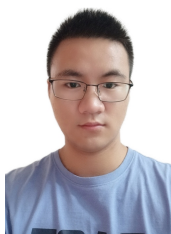
Xu Feng received the BSc degree in software engineering from Heilongjiang University, China, in 2019. He is currently pursuing the Master degree in School of Software, Dalian University of Technology, China. His research interests include big scholarly data and data science.

Jing Ren received the Bachelor degree from Huaqiao University, China, in 2018, and the Master degree from Dalian University of Technology, China, in 2020. She is currently pursuing the Ph.D. degree in School of Engineering, IT and Physical Sciences, Federation University Australia. Her research interests include data science, graph learning, anomaly detection, and social computing.

Jiaying Liu received the BSc and Ph.D. degrees in software engineering from Dalian University of Technology, China. She is currently an Assistant Professor in School of Economics and Management, Dalian University of Technology, China. Her research interests include data science, big scholarly data, and social network analysis.

Feng Xia (M'07-SM'12) received the BSc and PhD degrees from Zhejiang University, Hangzhou, China. He was Full Professor and Associate Dean (Research) in School of Software, Dalian University of Technology, China. He is Associate Professor and former Discipline Leader (IT) in School of Engineering, IT and Physical Sciences, Federation University Australia. Dr. Xia has published 2 books and over 300 scientific papers in international journals and conferences. His research interests include data science, artificial intelligence, graph learning, and systems engineering. He is a Senior Member of IEEE and ACM.

Huan Liu (F'12) received the B.Eng. degree in computer science and electrical engineering from Shanghai Jiaotong University and the Ph.D. degree in computer science from the University of Southern California. He is currently a Professor of computer science and engineering at Arizona State University. He was recognized for excellence in teaching and research in computer science and engineering at Arizona State University. His research interests include data mining, machine learning, social computing, and artificial intelligence, investigating problems that arise in many real-world applications with high-dimensional data of disparate forms, such as social media, group interaction and modeling, data preprocessing (feature selection), and textweb mining. His well-cited publications include books, book chapters, and encyclopedia entries and conference, and journal papers. He serves on journal editorial boards and numerous conference program committees, and is a Founding Organizer of the International Conference Series on Social Computing, Behavioral Cultural Modeling, and Prediction. He is a Fellow of IEEE, ACM, AAAI, and AAAS.