Graph Learning for Anomaly Analytics: Algorithms, Applications, and Challenges

JING REN, Federation University Australia, Australia

FENG XIA*, Federation University Australia, Australia

IVAN LEE, University of South Australia, Australia

AZADEH NOORI HOSHYAR, Federation University Australia, Australia

CHARU C. AGGARWAL, IBM T. J. Watson Research Center, USA

Anomaly analytics is a popular and vital task in various research contexts, which has been studied for several decades. At the same time, deep learning has shown its capacity in solving many graph-based tasks like, node classification, link prediction, and graph classification. Recently, many studies are extending graph learning models for solving anomaly analytics problems, resulting in beneficial advances in graph-based anomaly techniques. In this survey, we provide a comprehensive overview of graph learning methods for anomaly analytics tasks. We classify them into four categories based on their model architectures, namely graph convolutional network (GCN), graph attention network (GAT), graph autoencoder (GAE), and other graph learning models. The differences between these methods are also compared in a systematic manner. Furthermore, we outline several graph-based anomaly analytics applications across various domains in the real world. Finally, we discuss five potential future research directions in this rapidly growing field.

1 INTRODUCTION

Anomalies, which are also known as outliers, commonly exist in various real-world networks [12], such as fake reviews in opinion networks [121], fake news in social networks [119], outlier members in collaboration networks [94, 118], flash crowds in traffic networks [50], socially selfish nodes in mobile networks [110], and

---

*Corresponding author

Authors' addresses: Jing Ren, Institute of Innovation, Science and Sustainability, Federation University Australia, Ballarat, VIC, 3353, Australia, ch.yum@outlook.com; Feng Xia, Institute of Innovation, Science and Sustainability, Federation University Australia, Ballarat, VIC, 3353, Australia, f.xia@ieee.org; Ivan Lee, STEM, University of South Australia, Adelaide, SA, Australia, 5001, ivan.lee@unisa.edu.au; Azadeh Noori Hoshyar, Institute of Innovation, Science and Sustainability, Federation University Australia, Brisbane, QLD, 4000, Australia, a.noorihoshyar@federation.edu.au; Charu C. Aggarwal, IBM T. J. Watson Research Center, New York, USA, 10598, charu@us.ibm.com.

network intrusions in computer networks [24]. The exploration of anomaly detection research is dating back to 1960s and it has been a popular research field for several decades [35]. With the increasing demand and broad applications in different domains, anomaly analytic plays an increasingly important role in various communities such as data mining and machine learning.

With the advancement of deep learning, graph learning is proposed subsequently, which is coined for deep learning-based models that are applied into graph-structured data [112, 128]. Due to its convincing performance and explainability, recent years have witnessed, in varied disciplines, an increasing number of studies focusing on anomaly detection and prediction tasks by utilizing deep graph models [99, 137], which is not limited to shallow network embedding such as random walks [40, 111]. As a unique non-Euclidean data structure, graphs are able to represent entities and their relationships in different kinds of scenarios. However, this research direction faces several inevitable problem complexities to all detection methods when applying deep learning and artificial intelligence in real-world networks [57, 105].

- **Irregularity of graph structure.** Unlike other regular structured data, such as text, sequences, and images, nodes in a graph are unordered and can have distinct neighborhoods, which makes the structure of graphs irregular. Therefore, some traditional deep learning architectures cannot be directly applied, such as convolution and pooling operation in convolutional neural networks (CNNs) [72].
- **Heterogeneous anomaly classes.** The types of nodes and links are generally not unitary in a graph, which leads to the emergence of heterogeneous information networks (HINs). HINs usually incorporate more complex information among entities and relationships, especially those containing different modalities [85], which are very important in identifying different types of anomalies in a specific graph.
- **Scalability to real-world networks.** Nowadays, real-world networks such as social networks are composed of millions or even billions of nodes, edges, and attribute information [113]. This kind of large-scale network definitely increases computational complexity. Therefore, it is imperative to devise scalable models having a linear time complexity with respect to the graph size.
- **Label scarcity.** Compared with manually generated graph data, there are mainly two reasons for the sparsity of real-world networks. The first one is the scale-free network structure nature that the degree of nodes in most real-world networks follows long-tailed distribution [123]. The other one is limited by the collection technology and privacy protection in the process of crawling data. Moreover, due to the lack of labeled datasets, devising unsupervised anomaly detection models is becoming important.
- **Diverse types of anomalies.** Several types of anomalies have been explored such as node, edge, subgraph, and path (shown in Fig 2). Node anomalies are entities that show anomalous behaviours in the whole graph compared with other nodes, e.g., users who spread fake news in social networks. Other types of anomalies have similar concepts and their own real-world applications. Here, subgraph anomaly is difficult to detect because the individual nodes could show normal behaviours when extracted from an anomalous subgraph.

There have been a line of deep anomaly detection research demonstrating significantly better performance than conventional models on solving the above-mentioned challenges. Despite the fact that the adopted technologies vary from Graph Convolution Networks (GCNs) to Graph AutoEncoder (GAEs), most methods focus on detecting or predicting an anomaly in a specific situation due to the complexity of existing anomalies. To the best of our knowledge, little attention has been devoted to summarizing these methods in a comprehensive way and clearly analyzing how they are applied to solve real-world application scenarios.

1.1 Related surveys and novelty

There are several surveys related to our work. Zamini et al. [122] summarized the anomaly detection techniques in four real-world application scenarios, namely banking, wireless sensor networks, social networks, and healthcare. Akoglu et al. [3] reviewed the anomaly detection methods using graph metric-based techniques, Ranshous et
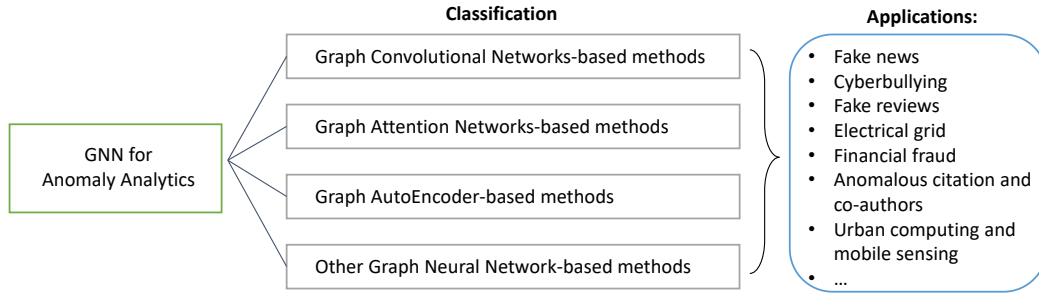
Fig. 1. Classification of deep graph methods in solving anomaly analytics tasks and real-world applications.

al. [78] only focused on anomaly detection methods in dynamic networks, while Bilgin et al. [8] briefly reviewed some non-deep learning methods of detecting anomalies in dynamic networks. Both Chalapathy et al. [11] and Pang et al. [75] concentrated on deep learning enabled anomaly detection in different kinds of data, which is not limited to graph data. [67] reviewed the contemporary deep learning methods for graph anomaly detection and categorized existing work according to the anomalous graph objects. Actually, there are also some surveys focusing on introducing the main concepts and frameworks of Graph Neural Networks (GNNs) [109, 137], and divided the corresponding methods according to the type of GNN models. Inspired by this classification strategy, we also divided the graph learning models in terms of the model type when introducing the specific anomaly detection tasks. However, the main focus of our survey and GNN survey are totally different in spite of the similar classification strategy.

This work is different from previous studies in that we aim to summarize the graph learning methods systematically and comprehensively for detecting anomalies in various graphs, ranging from homogeneous to heterogeneous, non-attributed to attributed, undirected to directed, rather than focusing on only one specific kind of graph. To fill this gap, we divide the existing methods into four categories based on their model architectures and training strategies: Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs), Graph AutoEncoders (GAEs), and other GNN-based methods (shown in Figure 1). The main characteristics of these methods are compared and summarized in Table 1. The characteristics of these basic models are briefly introduced in Section 2. In summary, the contributions of this work are outlined as follows:

- A systematic summarization and comparison of graph learning methods for anomaly analytics is presented. Specifically, we delineate their capabilities in addressing the existing problem complexities among all categories of the methods.
- An overview of major anomaly analysis tasks in various application domains is given.
- Insights into future research directions in this field are provided.

1.2 Organization

The rest of this survey is structured as follows. Section 2 presents the notations and preliminaries of graph learning models, which will be used in the subsequent sections. The anomaly analytics methods are reviewed in Sections 3 to 6. In Section 7, we outline several real-world applications of anomaly analytics that can be solved with deep graph models, and discuss some future research directions and challenges in Section 8. Finally, we briefly conclude this survey in Section 9.

Table 1. Summary of graph learning models in detecting and predicting anomalies

| Type	Method	Graph Type	Anomaly Type	Dataset	Application
GCN	GEM [62]	HIN	Node	-	Malicious account
	GCNSI [26]	Undirected network	Node	Karate, Dolphin, Jazz, US Power grid, Ego-Facebook[1]	Rumor source detection
	GAS [54]	HIN	Node&Edge	-	Spam review detection
	SpecAE [56]	Attributed networks	Node	Cora, Pubmed, PolBlog [76]	-
	DOMINANT [22]	Attributed networks	Node	BlogCatalog [103],Flickr [89], ACM [88]	-
	GCNwithMRF [108]	Directed graph	Node	TwitterSH [53], 1KS-10KN [116]	Social spammer detection
	Bi-GCN [7]	-	Edge	Weibo [65], Twitter [66]	Rumor detection
	GCAN [63]	Weighted graph	Node	Twitter	Fake news detection
	TPC-GCN [134]	HIN	Node	Weibo [65], Reddit [37]	Controversy detection
	AANE [29]	-	Edge	Disney[2], Enron[3]	-
	HMGNN [142]	Vanilla graph	Node	-	Fraud invitation
	GraphRfi [127]	Bipartite Graph	Node	Yelp [79], Amazon [70]	-
	AddGraph [131]	Dynamic graph	Edge	UCI [73], Digg [18]	-
	ST-GCAE [68]	ST Graph	Graph	ShanghaiTech [64]	Anomalous action
	StrGNN [10]	Temporal Graph	Node	UCI, Digg [18]	-
	ANEMONE [45]	-	Node	Cora, Citeseer, Pubmed	-
	CoLA [60]	Attributed network	Node	[103], [89], [88]	-
	GCCAD [13]	Attributed network	Node	Aminer, MAS, Alpha, Yelp	-
GAT	HAGNE [100]	HINs	Graph	-	Unknown malware
	HACUD [41]	Attributed HINs	Node	-	Cash-out user detection
	SemiGNN [95]	Multiview graph	Node	-	Financial fraud
	AA-HGNN [81]	HINs	Node	BuzzFeed[4]	Fake news detection
	mHGNN [32]	Attributed HINs	Subgraph	-	Illicit traded product
	GDN [19]	Directed graph	Node	SWaT [69], WADI [1]	Anomalous sensors
	TGBULLY [34]	Temporal graph	Subgraph	Instagram [39], Vine [77]	Cyberbullying detection
	TADDY [61]	Dynamic graph	Node	Email [5], AS-Topology [6]	-
GAE	AEHE [31]	HINs	Path	ACM [88]	Co-authored event
	AEGIS [21]	Attributed networks	Node	BlogCatalog,Flickr, ACM	-
	DONE [6]	Attributed network	Node	Cora, Citeseer, Pubmed [7]	-
	DeepSphere [90]	Dynamic networks	Node	NYC taxi trip [8], HERMEVENT [20]	-
	UCD [17]	Attributed networks	-	Instagram [39], Vine [77]	Cyberbullying detection
	SL-GAD [10]	Attributed network	Node	Cora, Citeseer, Pubmed	-
Other	GAL [130]	Bipartite graph	Node	Bitcoin-Alpha [52], Tencent-Weibo [44]	-
	CARE [27]	Multi-relation graph	Node	Yelp, Amazon	Camouflaged fraudsters
	Meta-GDN [23]	Cross-network	Node	PubMed [83], Reddit [36]	-
	GAAN [15]	Attributed network	Node	BlogCatalog, Flickr, ACM	-
	MAHINDER [135]	Multi-view HINs	Node	-	Financial defaulter
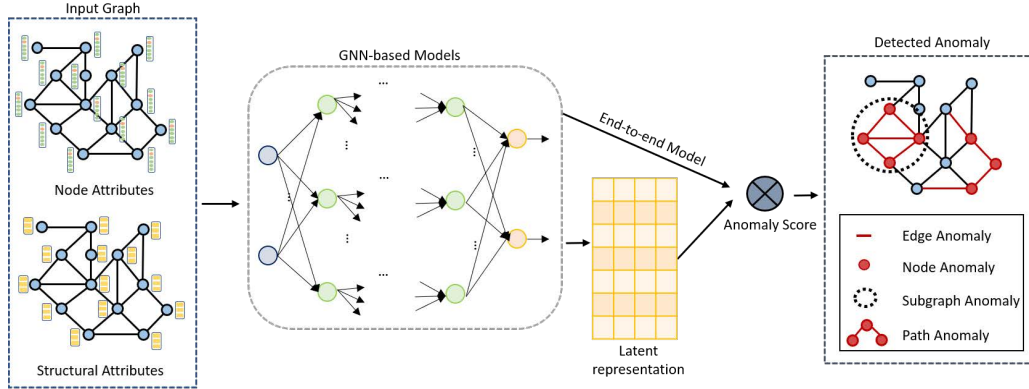
Fig. 2. Illustration of the whole process of detecting anomalies in graph data with deep graph models. The models are mainly divided into two parts according to whether anomaly score is calculated by latent representation or directly generated by end-to-end models. There are mainly four types of graph anomalies, namely node, edge, (sub)graph, and path anomaly.

2 NOTATIONS AND PRELIMINARIES

2.1 Notations

A graph[1] is represented as $G = (V, E)$, where $V = \{v_1, ..., v_n\}$ is a set of n nodes and $E \subseteq V \times V$ is a set of m edges between nodes. A graph may have different types, such as weighted or unweighted, directed or undirected. Here, if a graph is directed, then $e_{ij} = (v_i, v_j) \in E$ denotes an edge pointing from v_i to v_j. The neighborhood set of a node v_i is defined as $N(v_i) = \{v_j \in V | (v_i, v_j) \in E\}$. The adjacency matrix of a graph is a $n \times n$ matrix, which is denoted as A. We use $A(i, :), A(:, j), A(i, j)$ to denote the ith row, j^{th} column, and an element of A, respectively. For an unweighted graph, the element of its adjacency matrix is defined as:

$$\mathbf{A}(i, j) = \begin{cases} 1 & \text{if } e_{ij} \in E \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

For a weighted graph, $A(i, j)$ is defined as the weight of edge e_{ij}. A graph may have node attributes X^V and edge attributes X^E, where X^V is the node attributes matrix and X^E is the edge attributes matrix, respectively. If the feature matrix is used as X for convenience, the default setting X refers to node attributes matrix. Functions are marked with curlicues, e.g., $\mathcal{F}(\cdot)$.

Throughout this paper, we use bold uppercase characters denoting matrices and bold lowercase characters representing vectors, like a matrix A and a vector a. Unless particularly specified, the notations used in this paper are illustrated in Table 2. Then, we provide a formal definition and brief introduction of some predefined matrices to better understand the concepts described in this paper.

---

[1]http://www-personal.umich.edu/~mejn/netdata/&http://snap.stanford.edu/

[2]https://www.ipd.kit.edu/~muellere/consub/

[3]https://www.cs.cmu.edu/~./enron/

[4]https://github.com/KaiDMML/FakeNewsNet/tree/old-version

[5]http://networkrepository.com/email-dnc

[6]http://networkrepository.com/tech-as-topology

[7]https://linqs.org/datasets/

[8]https://portal.311.nyc.gov/

[1]Graph and network are used interchangeably in this paper.

2.2 Preliminaries

Given an undirected graph, the Laplacian matrix is defined as $L = D - A$, where $D \in \mathbb{R}^{n \times n}$ is a diagonal degree matrix with $D_{ii} = \sum_{j=1}^{n} A_{ij}$. $L = Q \Lambda Q^{T}$ denotes eigendecomposition, where $\Lambda \in \mathbb{R}^{n \times n}$ is a diagonal matrix of eigenvalues in ascending order and $Q \in \mathbb{R}^{n \times n}$ is composed of corresponding eigenvectors. The element $P(i, j)$ of transition matrix $P = D^{-1}A$ represents the probability of a random walk from node v_i to node v_j.

As mentioned previously, this survey aims to introduce existing research on graph anomaly detection and prediction. A graph is an abstract data type consisting of a set of nodes (a.k.a. vertices) representing entities, with edges between nodes representing relations or connections. Anomalies in graph data fall into four main categories: node anomaly, edge anomaly, path anomaly, and (sub)graph anomaly. The whole process of detecting anomalies with deep graph models is briefly illustrated in Figure 2.

When learning a deep model on graphs for anomaly analytics tasks, we divid the models into four categories based on their model architectures and training strategies. Here, we briefly introduce the process and potential mechanism of these graph neural network models.

- **Graph Convolutional Networks (GCNs)** Considering that graphs lack a grid structure like image and text, it is impractical to directly apply standard convolution operation on graphs. Graph convolution is generally divided into two categories, *spectral convolution*, which performs Fourier transform on graph signals, and *spatial convolution*, which learns structural information by aggregating node neighbors [107]. The graph signal X in spectral methods is filtered by:

$$Z = f(X, A) = \widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} X \Theta \tag{2}$$

  where Θ is a matrix of learnable parameters and Z is the convolved signal matrix.
  In addition, the equation of learning node representation of node i in a Graph Convolution Network (GCN) can be written as:

$$h_i = \sigma \left(\sum_{j \in N(v_i)} \alpha_{ij} W h_j \right). \tag{3}$$

  where W is weight matrix to be learned, and α_{ij} is set as 1 in GCN. The calculation formula of α_{ij} will be introduced in GAT.

- **Graph Attention Networks (GATs)** It is acknowledged that spatial convolution is to aggregate features from node neighbors to update the hidden state of this node in the next layer. The aggregation operation could be (weighted) summarization, averaging, and maximization. Graph Attention Network (GAT) is a special type of spatial convolution methods. Although some spatial methods also consider the node importance and allocate a predefined weight for every neighbor, a Graph Attention Network is proposed so that the weight of nodes can be learned automatically by applying the attention mechanism to neighbors in the model [92]. The influence α_{ij} of node v_j on node v_i in GAT is calculated as:

$$\alpha_{ij} = \frac{exp(\sigma(a^{T}[W h_i \| W h_j]))}{\sum_{k \in N(i)} exp(\sigma(a^{T}[W h_i \| W h_k]))}. \tag{4}$$

  Here, a denotes a weight vector and the symbol $\|$ is the concatenation operation on two vectors.

- **Graph AutoEncoder (GAE)** Graph autoencoder is a popular model used in unsupervised learning tasks [4, 30]. Similar to the general autoencoder model, GAE is composed of an *encoder* compressing the sparse node vector (input) into a low-dimensional representation through learning node structural features, and a *decoder* reconstructing the dense vector into a high-dimensional vector similar to the input as much as possible. Based on this mechanism, an essential part of loss function in GAE models is to minimize the

difference between the input and output vectors:

$$\min_{\Theta} \mathcal{L}_2 = \parallel A - \hat{A} \parallel_2 + \parallel X - \hat{X} \parallel_2, \tag{5}$$

where A and X are the input node adjacency and attribute matrix, and $\hat{A}$ and $\hat{X}$ are the reconstructed node structure and attribute matrix. It should be noted that the encoder could be any kind of neural network like MLP, Recurent Neural Network (RNN), and GCN. Therefore, there are a line of anomaly detection methods combining GAE and GCN. We refer readers to [128] for more details about deep graph models and their applications.

Table 2. Commonly used notations

Notations	Descriptions
$G = (V, E)$	A graph.
$N(v)$	The neighbors of a node v.
A	The graph adjacency matrix.
D	The diagonal degree matrix.
X	The graph feature matrix.
$D_{ii} = \sum_{j=1}^{n} A_{ij}$	The degree of node i.
$L = D - A$	The Laplacian matrix.
U	The eigenvector matrix of Laplacian matrix.
Λ	The eigenvalue matrix of Laplacian matrix.
A^T	The transpose of the matrix A.
A^n	The nth power of A.
$H^{(l)}$	The hidden representation in the lth layer.
W	The weight parameter matrix.
b	The bias parameter vector.
Z	The generated node embedding matrix.
Θ	Learnable parameters.
$\parallel$	The concatenation of two vectors.
$\sigma(\cdot)$	The sigmoid activation function.
$\lvert \cdot \rvert$	The length of a set.

3 GCN-BASED METHODS

As the most popular structure among the deep graph models, Graph Convolutional Networks (GCNs) can learn and generate node embeddings through the operation of convolution, which refers to the process of aggregating information from the nodes' local neighborhoods. In this section, we introduce the GCN-based anomaly detection and prediction methods, which is also the most popular model type among all anomaly analytics models. The methods are divided into two classifications according to whether the methods are devised for specific anomaly detection tasks or not, namely, general models and task-driven models. A toy model of how anomalous users are detected in social networks with spatial convolution operation is shown in Figure 3. The main characteristics of these methods are summarized in Table 3.
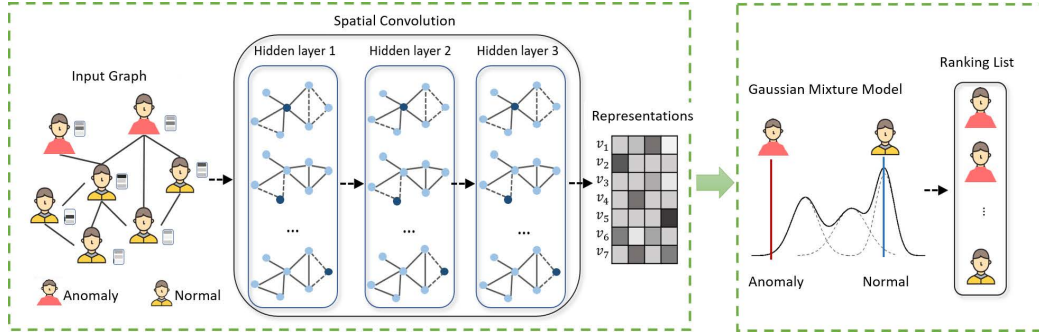
Fig. 3. An illustration of applying spatial convolution operation in anomalous user detection in social networks, where nodes are affected only by their immediate neighbors. Both attribute feature and structure feature could be learned with a GCN model to get the final anomaly ranking list.

3.1 General models

In [56], the authors defined two types of node anomaly according to its global location and topological network structure, named global anomaly and community anomaly, respectively. When learning global anomaly node embeddings, an autoencoder is applied to extract node attributes X. As for community anomaly representation, the authors designed a convolutional encoder and deconvolutional decoder networks based on their neighborhoods. Then, the anomalous node could be detected by measuring the embedding's energy in the Gaussian Mixture Model. Similarly, Ding et al. [22] and Zhu et al. [140] proposed to learn node embeddings by combining AutoEncoder with Graph Convolutional Networks in attributed networks. Specifically, the encoder module extends the operation of convolution in the spectral domain and learns a layer-wise new latent representation. Then, the structure reconstruction decoder $A - \hat{A}$ and attribute reconstruction decoder $X - \hat{X}$ are jointly learned to compute the anomaly score, which can be formulated as:

$$score(v_i) = (1 - \alpha)\|a_i - \hat{a_i}\|_2 + \alpha\|x_i - \hat{x_i}\|_2, \tag{6}$$

where α is a hyper-parameter balancing the importance of reconstructed structure and attribute information.

Instead of detecting anomalous nodes, Duan et al. [29] generated node embeddings by combining GAE and GCN to detect anomalous links. They assumed that a link with a lower value of predicted presence probability is regarded as anomalous, which is calculated as:

$$P_{u,v} < \underset{v' \in N_u}{\mathrm{MEAN}} P_{u,v'} - \mu \cdot \underset{v' \in N_u}{\mathrm{STD}} P_{u,v'}, \tag{7}$$

where MEAN and STD represent the mean and standard operations respectively, and μ is a parameter. P is the predicted presence probability matrix.

In [131], the authors aimed to incorporate all possible features in the proposed framework AddGraph, including structural, content, and temporal features. In AddGraph, they used a GCN incorporating content and structural features, with an attention-based GRU (Gated Recurrent Unit), which can combine short-term and long-term states. After obtaining the hidden states of nodes at timestamp t, the anomalous score of an edge is computed as:

$$\mathcal{F}(i, j, w) = w \cdot \sigma(\beta \cdot (\| a \odot h_i + b \odot h_j \|_2^2 - \mu)), \tag{8}$$

where h_i and h_j are the hidden states of the i-th node and j-th node, respectively. Other characteristics are parameters to be adjusted. a and b are parameters to optimize the output layer, and β and μ are hyper-parameters.

Jin et al. [45] leveraged a multi-scale contrastive learning technique to capture node anomalies in multiple scales. ANEMONE simultaneously performs patch- and context-level contrastive learning via two GCN models. Anomaly is identified by leveraging the statistics of multi-round contrastive scores.

Similarly, Liu et al. [60] also proposed to detect node anomalies in attributed network in a contrastive learning way. The objective of their model CoLA is to discriminate the agreement between the elements within the selected instance pairs, which is finally used to calculate the anomalous scores of nodes. The difference between CoLA and ANEMONE is the process of sampling. CoLA selected the local subgraph including the target node as positive sample, while local graph without target node is negative sample.

Differing from existing graph contrastive learning frameworks for GNN pre-training, Chen et al. [13] performed contrastive learning in a supervised learning way. In other words, the negative samples are constrained to anomalous nodes instead of being constructed according to some rules. Considering that the bottleneck of anomaly detection tasks is the lack of sufficient anomaly labels, they proposed to construct pseudo anomalies via corrupting the original graph.

3.2 Task-driven models

To detect malicious account at a mobile cashless payment platform, Liu et al. [62] jointly learned the topology of a heterogeneous graph and the features of local structures of the nodes. Specifically, they constructed "homogeneous connected subgraph" based on an assumption that an edge (i, i') is added if both account i and i' login to the same device in the original heterogeneous graph. This subgraph is composed of only accounts as nodes. The function to learn the embeddings of nodes is defined as:

$$H^{(l+1)} \leftarrow \sigma(X \cdot W + \sum_{d=1}^{|D|} softmax(\alpha_d) \cdot A^{(d)} \cdot H^{(l)} \cdot V_d), \tag{9}$$

where $|D|$ is the number of subgraphs extracted from the original graph, and V_d is a parameter controlling the shape of the function. Moreover, an attention mechanism is also utilized in the learning process of different types of subgraphs, i.e., $softmax(\alpha_d) = \frac{exp\ \alpha_d}{\sum_i exp\ \alpha_i}$, and $\alpha = [\alpha_1, ..., \alpha_{|D|}]^T$ is a free parameter to be learned.

Based on an assumption that representing nodes with the information of its neighbors will effectively improve the performance of the source node detection task, Dong et al. [26] designed a model GCNSI to locate multiple rumor sources without prior knowledge of the underlying propagation model. This model learns node embeddings by adopting convolution in the spectral domain, which considers its multi-hop neighbors' information. The propagation strategy of GCNSI is modified based on LPSI [106].

Concerning the task of spam review detection, Li et al. [54] aimed to capture the local context and global context of a comment. The proposed model GAS simultaneously integrates a heterogeneous bipartite graph and a homogeneous comment graph. The comment edge embedding in bipartite graph is to aggregate the hidden states of three variables in the previous layer, i.e., the edge itself and its linked two nodes:

$$H_e^l = \sigma(W_E^l \cdot AGG_E^l(H_e^{l-1}, H_{U(e)}^{l-1}, H_{I(e)}^{l-1})), \tag{10}$$

where

$$AGG_E^l(H_e^{l-1}, H_{U(e)}^{l-1}, H_{I(e)}^{l-1}) = H_e^{l-1} \parallel H_{U(e)}^{l-1} \parallel H_{I(e)}^{l-1}. \tag{11}$$

Here, $U(e)$ and $I(e)$ are user node set and item node set linked by edge e, respectively. Similarly, the user and item embedding are calculated in the same way and the comment embedding in the comment graph can be obtained from a general GCN model. Finally, the classification result y can be calculated according to:

$$y = classifier(z_i \parallel z_u \parallel z_c \parallel p_c), \tag{12}$$

where z_i, z_u, and z_c are item, user, and comment embeddings obtained from bipartite graph, and p_c is the comment embedding from comment graph, respectively.

With the aim of detecting a rumor on social media, Bian et al. [7] proposed a top-down GCN (TD-GCN) to model the rumor propagation features, and a bottom-up GCN (BU-GCN) to model the rumor dispersion features, respectively. The node representations are learned over a two-layer GCN:

$$H_1^{TD} \leftarrow \sigma(\hat{A}^{TD} X W_0^{TD}), \tag{13}$$

$$H_2^{TD} \leftarrow \sigma(\hat{A}^{TD} H_1^{TD} W_1^{TD}), \tag{14}$$

where H_1^{TD} and H_2^{TD} refer to the two-layer hidden features of the TD-GCN. The bottom-up features of BU-GCN are calculated in the same manner as Eq. 13 and 14, while the adjacency matrix should be transposed.

Another similar topic in social media is fake news detection. Lu et al. [63] aimed to model the interactions among users by creating a propagation graph as a part of the proposed model. The propagation graph $G_i = (U_i, E_i)$ is constructed by the set of users U_i who share or retweet the topic s_i, and the edge is weighted by the cosine similarity between the feature vectors of users. Then, the user embeddings will be learned by a GCN model based on this weighted propagation graph. Similarly, Zhong et al. [134] created a Topic-Post-Comment graph for target posts in the task of controversy detection, where the nodes represent topic, post, or comment, and the edges refer to the corresponding interactions between two nodes. The node representations are obtained through a two-layer GCN, the same as Eq. 13 and Eq. 14.

As the first application of deep graph model in the task of fraud invitation detection, Zhu et al. [142] proposed HMGNN model by dividing the whole network into $|D|$ mini-graphs, which were represented by hypernodes. The hypergraph is generated by adding edges between mini-graphs. Based on this graph, the convolution for hypernodes is defined as:

$$H^{l+1} \leftarrow \sigma(X_h U^l + \sum_{d \in D} ATT_d(\widetilde{A}_h^d H^l W_d^l + b_d^l)), \tag{15}$$

where ATT_d is the attention mechanism, and U^l are free parameters to be trained. Here, $H^0 = [X; X_h^1; ...; X_h^{|D|}]$ is the initial representation of the whole graph which concatenates the feature matrix of normal- and hyper- nodes.

To detect social spammers in a semi-supervised way, Wu et al. [108] combined Graph Convolutional Networks (GCNs) and Markov Random Field (MRF) on directed social networks. The layer-wise propagation rule of GCN is defined as:

$$H^{(l+1)} = \sigma(D_i^{-1} A_i H^{(l)} W_i^{(l)} + D_o^{-1} A_o H^{(l)} W_o^{(l)}) + \tilde{D}_b^{-\frac{1}{2}} \tilde{A}_b \tilde{D}_b^{-\frac{1}{2}} H^{(l)} W_b^{(l)}, \tag{16}$$

where A_i, A_o, and A_b are three types of adjacency matrices constructed according to three different definitions of neighbors. Considering that different characteristics of pairwise nodes can have different influences on social networks, the authors proposed to use MRF modeling for the joint probability distribution of users' identities. The MRF is formulated as a RNN in this paper to perform multi-step inference when computing the posterior distribution.

Since deliberately inserting fake feedback will cause the recommender system bias to the malicious users' favor, Zhang et al. [127] presented a GCN-based user representation learning framework to perform robust recommendation and fraudster detection in a unified way. Given a weighted bipartite rating graph $G = (U \cup V, E)$, GCN is adopted to capture topological neighborhood information and side information of nodes. The user and item embedding are calculated as:

$$z_u = \sigma(W \cdot AGG(h_k, \forall k \in N(u)) + b), \tag{17}$$

$$z_v = \sigma(W \cdot AGG(h_q, \forall q \in N(v)) + b), \tag{18}$$

where h_k and h_q are the neighbor information for each node. Here, the attention mechanism is incorporated into the aggregation function.

Table 3. A comparison of the GCN-based models

Sec.	Method	Type	Convolution	Characteristic
3.1	SpecAE [56]	Spectral	-	AutoEncoder
	DOMINANT [22]	Spatial	First-order	-
	DeepAD [140]	Spatial	First-order	-
	AANE [29]	Spatial	First-order	-
	AddGraph [131]	Spatial	First-order	GRU + Attention
	ANEMONE [45]	Spatial	First-order	Contrastive learning
	CoLA [60]	Spatial	First-order	Contrastive learning
	GCCAD [13]	Spatial	First-order	Contrastive learning
3.2	GEM [62]	Spatial	First-order	Attention mechanism
	GCNSI [26]	Spectral	First-order	Semi-supervised learning
	GAS [54]	Spatial	First-order	Attention mechanism
	Bi-GCN [7]	Spectral	Polynomial	DropEdge
	GCAN [63]	Spatial	First-order	-
	TPC-GCN [134]	Spatial	First-order	-
	HMGNN [142]	Spatial	First-order	Adversarial learning +Attention
	GCNwithMRF [108]	Spatial	First-order	-
	GraphRfi [127]	Spatial	First-order	Attention mechanism
	TSN [133]	Spatial	First-order	-
	ST-GCAE [68]	Spatial	First-order	Attention mechanism

It is acknowledged that noisy labels will influence the results of anomaly detection algorithms in some degree. Then, instead of directly generating latent representations, Zhong et al. [133] designed a GCN-based model to correct noisy labels before detecting anomalous actions in videos. Here, the feature similarity graph is constructed with nodes denoting snippets and edges referring to the similarity between two snippets. Another temporal consistency graph module is directly built upon the temporal structure of a video.

Pose estimation is the first step of detecting anomalous actions in videos, and the extracted poses can be embedded by deep graph models. In [68], the authors proposed spatio-temporal graph convolution block, which is composed of a spatial-attention graph convolution, a temporal convolution, and a batch normalization. The generated latent vector is fed into a cluster layer to obtain a normality score. Here, the normality score is calculated by a Dirichlet Process Mixture Model (DPMM) for evaluating the distribution of proportional data.

3.3 Discussion

As it can be found from the GCN-based anomaly detection models we have discussed above, the modern GCN model could learn both local and global structure features of a graph with convolution and pooling operations. To improve the training efficiency when imposing GCN on large-scale graphs, neighborhood samplings and layer-wise samplings are two common strategies to deal with the phenomenon that some nodes have high degrees (too many neighbors). In addition to node and edge anomaly, a characteristic of GCN is that it is more suitable to detect (sub)graph or group anomaly when compared with other GNN models.

The aforementioned models mostly focus on learning node features and graph structures, ignoring another important element consisting of a graph, i.e., edge. In some real-world networks, edges generally contain abundant information, such as edge types and corresponding attributes, which could play a key role in anomaly detection tasks. Therefore, incorporating edge features into graph anomaly detection models could be considered as a

future work [14]. Besides, although applying GCN to an inductive setting is verified [36], conducting inductive GCN for graphs without explicit features remains an open problem.

4 GAT-BASED METHODS

In deep graph models, the weights of node neighbors are defined as an equal or default setting. However, the importance of neighbors is mostly different in terms of their attribute and structural features. Motivated by the attention mechanism, Velivckovic et al. [92] proposed a graph attention network (GAT) by applying the attention mechanism to the spatial convolution operation of GCN. A toy example of how attention mechanism is applied into cyberbullying detection is shown in Figure 4. In this section, we summarize and introduce the anomaly analytics algorithms using graph attention networks. The methods are divided into 2 subsections in terms of the anomaly type, i.e., node anomaly detection and (sub)graph anomaly detection. The main characteristics of these methods are summarized in Table 4.
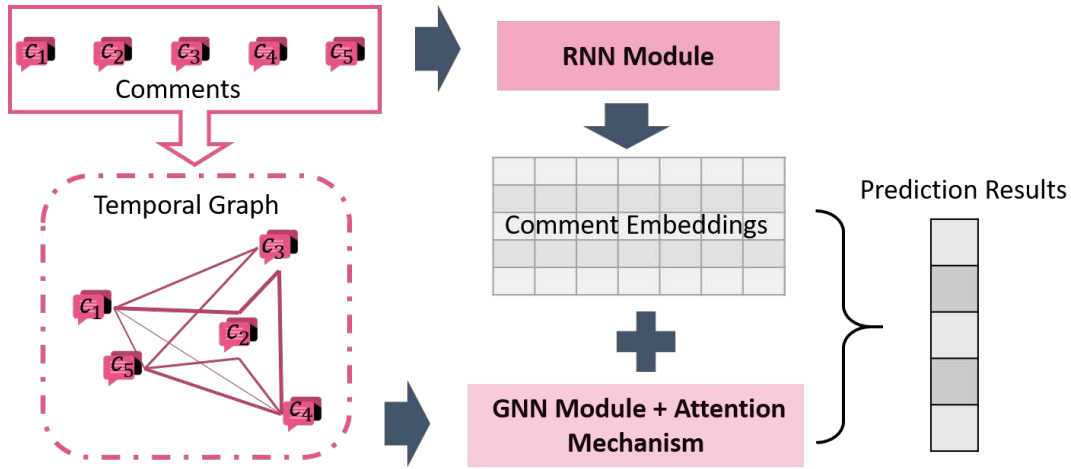

Fig. 4. An illustration of how attention mechanism is applied into cyberbullying detection. Each comment is first encoded by a RNN framework as the initial vector, and the comments are constructed as a temporal graph where nodes represent user comments and edges represent time intervals between two comments. Then, the attention mechanism is applied to learn the temporal information among these comments for final anomaly detection.

4.1 Node anomaly detection

To detect anomalous nodes in an Attributed Heterogeneous Information Network (AHIN), Hu et al. [41] applied feature and path attention mechanism to differentiate the importance of meta-paths as well as attribute information. As a basic analysis tool for heterogeneous graph, a meta-path captures the proximity among multiple nodes from a specific semantic perspective, which could be seen as a high-order structure. For example, the meta path "Author-Paper-Author" (APA) describes that two authors collaborated with each other in a particular paper. The feature attention of neighbor node i on node u in path ρ is calculated as:

$$\hat{\alpha}_{u,i}^{\rho} = \frac{exp(\alpha_{u,i}^{\rho})}{\sum_{j=1}^{K} exp(\alpha_{u,j}^{\rho})}. \tag{19}$$

Table 4. A comparison among different GAT-based models

Method	Attention Level	Objective Function	Other Characteristics
HAGNE [100]	node & path	Error sum of squares	Siamese network for graph matching
HACUD [41]	feature & path	Maximum likelihood estimation	Hierarchical attention mechanism
mHGNN [32]	node	Cross-entropy	Metagraph-guided neighbor search
SemiGNN [95]	node & view	Cross-entropy+Structure similarity	Semi-supervised learning
AA-HGNN [81]	node & schema	Cross-entropy	Adversarial active learning
GDN [19]	node	Mean squared error	Graph deviation scoring
TGBULLY [34]	node	-	Temporal graph interaction learning
MAHINDER [135]	node & path	Cross-entropy	-

The attention weight of path ρ for node u is defined as:

$$\beta_{u,\rho} = \frac{exp(z^{\rho^T} \cdot \tilde{f}_u^C)}{\sum_{\rho' \in \mathbb{P}} exp(z^{\rho'^T} \cdot \tilde{f}_u^C)}. \tag{20}$$

Here, z^ρ is the attention vector of meta-path ρ, and $\tilde{f}_u^C$ is the collection of user representations w.r.t. all meta-paths. The cash-out probability (i.e., anomalous score) is calculated via a regression layer with a sigmoid unit.

To detect user fraud in financial networks, Wang et al. [95] designed a hierarchical attention structure in graph neural network from node-level attention to view-level attention when generating graph embeddings. View-attention mechanism is applied to fuse multiple views of data information into user embeddings. Finally, a softmax function is used on the representations of the embedding layer to get the classification result.

In [81], the authors constructed a news-oriented heterogeneous information network with nodes of creators, subjects, and articles, and two links of write and belong-to. Based on this network, they proposed AA-HGNN to solve the problem of fake news detection. From the perspective of node-level attention, the model first aggregates the importance of the same-type neighbors for each news node and generates an integrated embedding of schema node. By using a transformation matrix, the embeddings of the nodes can be mapped into the same dimension. The logistic regression layer works as the classification layer to generate the detection results.

To give direct explanations like how anomalies deviate from normal behaviors, [19] proposed to use graph attention mechanism to predict the future behavior of a node. The anomaly score of node i is defined as the difference between the expected behavior and observed behavior at time t:

$$\text{Err}_i(t) = |s_i^{(t)} - \hat{s}_i^{(t)}|. \tag{21}$$

For session-level cyberbullying detection, the final embedding is fed into a single-layer dense network and predict its label.

In financial default user detection over online credit payment service, Zhong et al. [135] devised a meta-path-based encoder to capture local structural feature of nodes and links. The path representation is defined as the concatenation of node and link embeddings. Moreover, attention mechanism is applied to capture different importance of nodes/links of a path. After modeling the node and link interactions above, the learned representation is fed into several fully connected neural networks and a regression layer with a sigmoid unit for anomaly classification.

4.2 (Sub)graph anomaly detection

Wang et al. [100] proposed HAGNE to detect unknown malicious programs in computer systems. Instead of setting one-hop nodes as neighbors, the authors construct a contextual neighborhood set by searching for meta-paths. Then, three kinds of aggregators are applied to generate graph embeddings based on the generated meta-path set $\mathbb{M} = \{M_1, M_2, ..., M_{|\mathbb{M}|}\}$, namely, node-wise attentional neural aggregator, which is defined as:

$$h_v^{(i)(k)} = AGG_{node}(h_v^{(i)(k-1)}, \{h_u^{(i)(k-1)}\}_{u \in \mathcal{N}_v^i}), \tag{22}$$

where $i \in \{1, 2, ..., |\mathbb{M}|\}$, $k \in \{1, 2, ..., K\}$ denotes the layer index, and $h_v^{(i)(k)}$ is the feature vector of node v at the k-th layer in meta-path M_i; layer-wise dense-connected neural aggregator, which is inspired by DENSENET [42]:

$$h_v^{(i)(K+1)} = AGG_{layer}(h_v^{(0)}, h_v^{(1)}, ..., h_v^{(K)}); \tag{23}$$

and path-wise attentional neural aggregator, whose attentional weight is defined as:

$$\alpha_{ij} = \frac{exp(\sigma(b[W_b h_v^{(i)(K+1)} \| W_b h_v^{(j)(K+1)}]))}{\sum_{j' \in |\mathbb{M}|} exp(\sigma(b[W_b h_v^{(i)(K+1)} \| W_b h_v^{(j')(K+1)}]))}, \tag{24}$$

Then, the graph embedding can be calculated from the joint representation of all meta-paths:

$$h_G = AGG_{path} = \sum_{i=1}^{|M|} ATT(h_v^{(i)(K+1)}) h_v^{(i)(K+1)}. \tag{25}$$

Graph matching is used to measure the anomalous level of a program [80]. An alert will be triggered if the highest similarity score among all the existing programs is below the threshold. The similarity score is calculated as:

$$Sim(G_{i(1)}, G_{i(2)}) = \frac{h_{G_{i(1)}} \cdot h_{G_{i(2)}}}{\| h_{G_{i(1)}} \| \cdot \| h_{G_{i(2)}} \|}. \tag{26}$$

Subsequently, Fan et al. [32] identified the illicit traded product in underground market with a similar process. After constructing the neighbors set by the meta path-based method, the authors applied an attention mechanism to learn product and buyer embeddings, respectively. Finally, their embeddings are generated by concatenating each embedding based on a specific metagraph.

Social media contains multi-modal information such as comment, user, time, and history. Ge et al. [34] proposed to use temporal graph interaction learning module as a building block to detect cyberbullying in social networks. In this work, the authors incorporated GATs to automatically aggregate information from neighbor nodes to the central node in a temporal graph. Edge in the temporal graph denotes time dynamics, and the weight of the node pair (i, j) is defined as:

$$\alpha(z_i, z_j, t_i, t_j) = tanh((W_o z_i)^T z_j + W_t(t_j - t_i)). \tag{27}$$

4.3 Discussion

As we have explained, GAT is a branch of GCN. To improve GCN, GAT-based methods are separated as a unique section in which the importance of different neighbors on the central node is considered. The difference between these two sections is that the utilized traditional attention mechanism of Section 4 is only applied on nodes, while models in Section 3 are either using attention mechanism on other parts of the framework or using simple GCN function without any attention mechanism.

5 GAE-BASED METHODS

Graph AutoEncoder (GAE) is an unsupervised structure to generate low-dimensional representations, with the aim of minimizing the loss between the input of encoder and the output of decoder [91]. In this section, we present the GAE-based algorithms that are applied to anomaly analytics. The methods are classified into three types according to the training and learning schema, namely, General AutoEncoder, Adversarial Training, and Hypersphere Learning. The main characteristics of these methods are summarized in Table 5. In figure 5, we present a GAE-based model for detecting anomalous citation behaviors in a heterogeneous network.
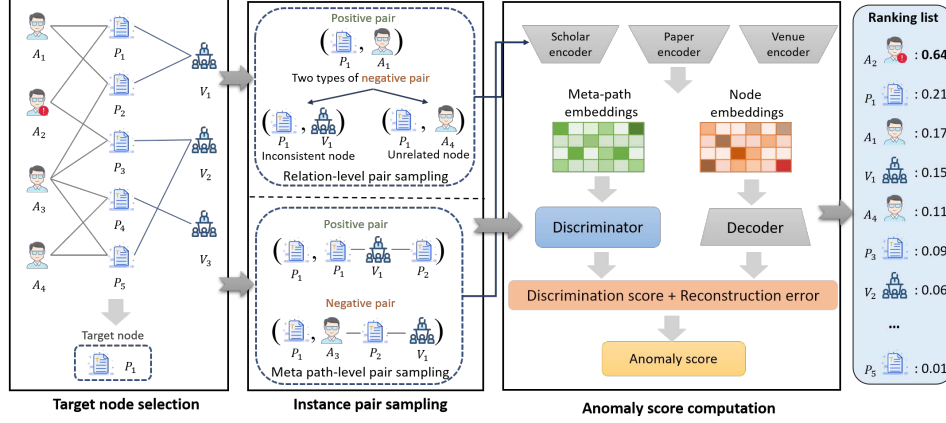

Fig. 5. An illustration of combining graph autoencoder (GAE) with contrastive learning for anomalous academic's detection in heterogeneous networks. After selecting a target node, the second step is to sample positive and negative instances from the network for contrastive learning. Then, different kinds of nodes are encoded with different encoders and a common decoder. The encoder aims to learn structure and attribute feature of nodes and generate low-dimensional vectors, and the decoder aims to reconstruct the input vector as similar as possible. The anomaly score is calculated by combining the discrimination score generated by the discriminator and the reconstruction loss generated by the AutoEncoder.

5.1 General AutoEncoder

Only considering the structure of a heterogeneous network is not sufficient for abnormal event detection due to the sparsity of a network. Fan et al. [31] proposed AEHE to learn both attribute embedding and the second-order structure-preserving node embedding. The heterogeneous attribute embedding of a node is generated by a Multilayer Perceptron (MLP) component, which consists of two hidden layers with ReLU as the activation function. As for the second-order structure embedding, the authors constructed a homogeneous graph by extracting symmetry meta-paths. Autoencoder is used to model the neighborhood structures, which is composed by an encoder:

$$r_i^t = \sigma(W_1^t \cdot s_i^t + b_1^t), \tag{28}$$

and a decoder:

$$\hat{s}_i^t = \sigma(\hat{W}_1^t \cdot r_i^t + \hat{b}_1^t), \tag{29}$$

where r_i^t is the latent representation of entity a_i^t, and $\hat{s}_i^t$ is the reconstructed representation of s_i^t. It should be noted that s_i^t is the ith row of the adjacency matrix, not just a node feature vector. Research have shown that human behaviors reflect self-selection bias and peer influence in online social network, which is closely associated

Table 5. A comparison among different GAE-based models

Method	Type	Objective Function	Other Characteristics
AEHE [31]	GAE	Cross-entropy+L2-reconstruction	Negative sampling
AEGIS [21]	GAE+GAN	Cross-entropy	GDN Encoder
DONE [6]	GAE+Discriminator	L2-reconstruction+Cross entropy	Adversarial learning
DeepSphere [90]	GAE+LSTM	L2-reconstruction	Hypersphere learning
UCD [17]	GAE+GCN	L2-reconstruction	Attention mechanism

with cyberbullying behaviors [33]. In this regard, Cheng et al. [17] used a GCN encoder and an inner product decoder to learn a latent matrix Z by minimizing the following reconstruction error:

$$\mathcal{F}(v_i) = \| A - \hat{A} \|_2^2, \tag{30}$$

where $\hat{A} = \sigma(ZZ^T)$, and $Z = GCN(X, A)$. Then, the anomalous session could be detected by measuring the embedding's energy in the Gaussian Mixture Model, which follows [56].

5.2 Adversarial training

Adversarial methods such as GAN and adversarial attacks are popular in the machine learning community in recent years. In [74], the authors incorporated an adversarial training scheme into GAEs as an additional regularization term. Motivated by this work, Ding et al. [21] proposed AEGIS to learn anomaly-aware node representations through graph differentiation networks (GDNs) for inductive anomaly detection. AEGIS is composed of a GAE to learn node embeddings for training new networks, and a GAN to calculate the anomaly scores of nodes. The autoencoder network is built with the graph differentiative layers. Specifically, a GDN layer has a hierarchical attention structure from node level:

$$h_i^{(l)} = \sigma(W_1 h_i^{(l-1)} + \sum_{j \in N_i} \alpha_{ij} W_2 \Delta_{i,j}^{(l-1)}), \tag{31}$$

where $\Delta_{i,j}$ denotes the embedding difference between nodes i and j; and neighbor level:

$$h_i^l = \sum_{k=1}^{K} \beta_i^k h_i^{(l,k)}. \tag{32}$$

Finally, the anomaly score of node i is computed according to the output of a discriminator:

$$score(z_i) = 1 - D(z_i^{'}). \tag{33}$$

In real-world networks, community outliers deviate significantly from other nodes in the same community in terms of link structures and attributes. To alleviate the influence of these outliers and generate robust node embeddings, Bandyopadhyay et al. [6] mapped every vertex to a low-dimensional vector and detected outliers via a deep autoencoder-based architecture. Moreover, the authors introduced adversarial learning for outlier resistant network embedding. Here, a discriminator is combined with two parallel autoencoders to align the embeddings in terms of link structure and node attributes.

5.3 Hypersphere learning

Inspired by hypersphere learning, Wang et al. [102] proposed One-Class Graph Neural Network (OCGNN) with the aim of minimizing the volume of a hypersphere that encloses normal nodes as much as possible. Then, the nodes out of the hypersphere are regarded as abnormal.

With the aim of identifying anomalous sample cases and nested anomalies within the anomalous tensors, Teng et al. [90] proposed DeepSphere by incorporating hypersphere learning into a LSTM Autoencoder model in a mutual supportive manner. Here, attention mechanism is also applied to differentiate and aggregate different neighbors. The motivation of DeepSphere is that the learned representations at large distance from the outside of hypersphere are regarded as anomalous, while the ones with small distances from the inside of the hypersphere tend to be normal.

5.4 Discussion

GAE is the most popular model in tackling unsupervised graph learning tasks, which can only consider the structural patterns by using graph adjacency matrix. However, GCN-based models are semi-supervised and could capture both node attributes and graph structures. Despite the different architectures between GAE and GCN, existing research have shown that it is possible to combine them together in a unified framework [9]. When applying GCN as the encoder, GAE could be applied to the inductive learning settings where node attributes are incorporated. Considering that the aim of GAE is to reconstruct the input embedding as similar as possible, it should be cautious when selecting the appropriate similarity metrics which have significant influence on subsequent anomaly detection results.
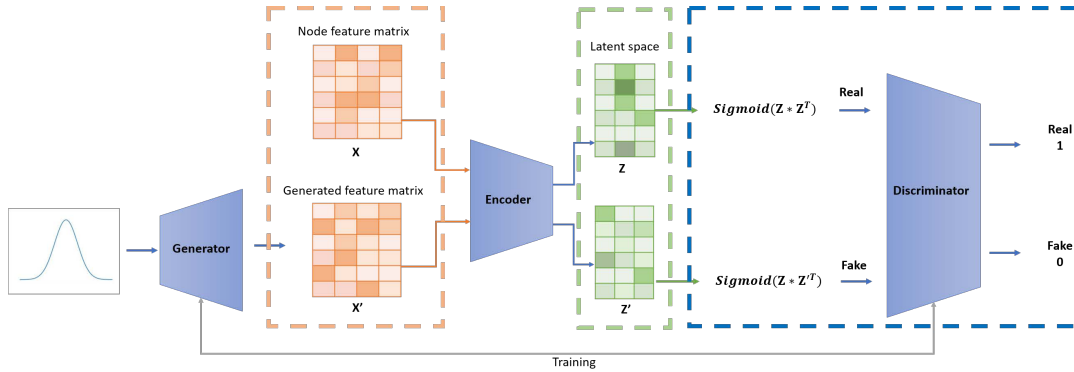

Fig. 6. The Generative Adversarial Network (GAN)-based anomaly detection model is composed of three main parts: a Generator sampling similar node attributes, an Encoder generating low-dimensional node representations, and a Discriminator differentiating real nodes embeddings from generated nodes embeddings.

6 OTHER METHODS

Apart from the deep graph models mentioned above, there are many other popular deep learning models that can be used for anomaly analytics tasks, such as Generative Adversarial Methods [96], Meta-learning [136], and Graph Reinforcement Learning [25]. In this section, we summarize these different deep graph models that are utilized to solve anomaly analytics tasks. The process of detecting anomalies with a Generative Adversarial Network (GAN) is shown in Figure 6. The main characteristics of these methods are summarized in Table 6.

6.1 GAN-based methods

With the rapid growth of research in Generative Adversarial Network (GAN) for high-dimensional data distribution approximation, Chen et al. [15] proposed to detect anomalies with a Generative Adversarial Attributed Network (GAAN), which is composed of three parts: a Generator sampling similar node attributes, an Encoder

Table 6. A comparison among other deep graph models

Method	Method/Problem Innovation
GAAN [15]	Generative adversarial network
GAL [130]	New loss function
CARE-GNN [27]	Reinforcement learning
Meta-GDN [23]	Cross-network meta-learning
OCGNN [102]	One-Class graph neural network
MAHINDER [135]	Financial default user detection over online credit payment service

generating low-dimensional node representations, and a Discriminator differentiating real nodes embeddings from generated nodes embeddings. The anomalous score is defined based on a context reconstruction loss L_G and a structure discriminator loss L_D:

$$\mathcal{F}(v_i) = \alpha \mathcal{L}_G(v_i) + (1 - \alpha)\mathcal{L}_D(v_i), \tag{34}$$

where $\mathcal{L}_G(v_i) = \| x_i - x_i' \|_2$, and $\mathcal{L}_D(v_i)$ is defined as:

$$\mathcal{L}_D(v_i) = \sum_{j=1}^{n} A_{ij} \cdot \sigma(\hat{A}_{ij}, 1) / \sum_{j=1}^{n} A_{ij}. \tag{35}$$

Larger value of $\mathcal{F}(v_i)$ indicates the node v_i is more likely to be anomalous.

6.2 Reinforcement learning-based method

In [130], the authors divided the graph anomaly detection tasks into two classifications, i.e., outlier detection and unexpected dense block detection. When applying graph learning models to generate embeddings of nodes, a new loss function was designed as:

$$\mathcal{L}(u) = \mathbb{E}_{u_+ \sim U_{u_+}, u_- \sim U_{u_-}} max\{0, g(u, u_-) - g(u, u_+) + \triangle_{y_u}\}, \tag{36}$$

where $\triangle_{y_u} = \frac{C}{n_{y_u}^{1/4}}$. $g()$ is a function to denote the similarity of the representations between any two user nodes. Here, U_{u_+} denotes the set of user nodes that has the same label as u, U_{u_-} refers to $U \backslash U_{u_+}$, and $n_{y_u} = |U_{u_+}|$. The construction of sets U_{u_+} and U_{u_-} have different strategies for corresponding tasks. It is impractical to exactly detect camouflaged fraudsters with graph learning detectors. Dou et al. [27] proposed three neural models to enhance the deep graph models against two kinds of camouflages, i.e. feature camouflage and relationship camouflage. Because camouflaged nodes should be filtered when selecting similarity-aware neighbors, a reinforcement learning process is formulated as a Bernoulli Multiarmed Bandit (BMAB) to find the optimal thresholds. The reward mechanism of epoch e is defined as:

$$\mathcal{R}(p_r^{(l)}, a_r^{(l)})^{(e)} = \begin{cases} +1, G(\mathcal{D}_r^{(l)})^{(e-1)} - G(\mathcal{D}_r^{(l)})^{(e)} \geq 0, \\ -1, G(\mathcal{D}_r^{(l)})^{(e-1)} - G(\mathcal{D}_r^{(l)})^{(e)} < 0. \end{cases} \tag{37}$$

Here, $G(\mathcal{D}_r^{(l)})^{(e)}$ refers to the average neighbor distances for relationship r at the l-th layer for epoch e. If the average distance of newly selected neighbors at epoch e is less than that of epoch $e - 1$, then the reward is positive.

6.3 Few-shot learning-based method

To investigate the novel problem of few-shot network anomaly detection under the cross-network setting, Ding et al. [23] designed a new graph learning architecture, namely Graph Deviation Networks (GDNs). GDN in

this paper is composed of three key building blocks: an encoder to generate node embeddings, an abnormality valuator to compute the anomaly score of nodes, and a deviation loss for optimization. Concretely, the GDN model can be formally represented as:

$$\mathcal{F}_\theta(\mathrm{A}, \mathrm{X}) = \mathcal{F}_{\theta_s}(\mathcal{F}_{\theta_e}(\mathrm{A}, \mathrm{X})), \tag{38}$$

which directly maps the input network to anomaly scores (scalar). After detecting anomalies in arbitrary networks, a meta-learner is learned to initialize GDN from multiple auxiliary networks, which possesses the ability to distill comprehensive knowledge of anomalies.

6.4 Discussion

There are also many other neural network structures and learning strategies being applied to detect graph anomalies, such as adversarial learning and reinforcement learning. Considering that the number of these methods is relatively less, we summarize all these methods into one section instead of different sections. Other aspects of common deep graph learning models include but not limited to graph reinforcement learning and graph adversarial learning.

It is well-known that the advantage of reinforcement learning is to actively learn from the feedback. In graph anomaly detection tasks, reinforcement learning could help in optimal selection of neighbors and aggregating them together for more informative node embeddings. Adversarial methods have shown its capacity in generating realistic entities, which improve the detection performance of anomalies that are hardly reconstructed from the latent space. However, this kind of anomaly detection methods faces multiple problems during the training process, such as failure to converge and mode collapse.

7 APPLICATIONS

Thus far, we have reviewed different graph learning methods in anomaly analytics tasks. In this section, we briefly introduce their applications in different kinds of networks.

7.1 Fake news

With the rapid growth of the Internet, social media provides a platform for people to participate and discuss online news more conveniently, like communicating news without the physical distance barrier among individuals and acquiring news at an unprecedented rate. In general, fake news in social media is defined as the verifiably false information that is generated by malicious users or social bots intentionally, with the aim of misleading the public. There have been research showing that fake news spread more quickly and broadly than true news [93].

Detecting fake news, especially at an early stage, is complex and challenging due to the characteristics of fake news. Various types of information are integrated when designing detection strategies, including news-related and social-related features [139]. Among the whole process of detection algorithms, extracting information from network-based features is a procedure to improve the performance of detection results. In social media, users form different kinds of networks in terms of interests, topics, and relations. For example, [71] proposed a heterogeneous graph to incorporate all major social actors and their interactions into node representations, which is constructed by user, news, and sources. Other types of networks also exist, for instance, co-occurrence network indicating user engagements, friendship network showing the following relationships, and diffusion network tracking the source of fake news. We refer readers to [86, 138] for more information about the research on fake news.

7.2 Cyberbullying

Based on the definition of bullying, cyberbullying is, by extension, defined as an aggressive act intentionally carried out by a group or individual using an electronic device, against people who cannot easily defend themselves.

Research have shown that cyberbullying is quite prevalent on social media with 54% of young people reportedly cyberbullied on Facebook [82]. Apart from the traditional research using merely content-based features, recent years have witnessed a proliferation of research focusing on incorporating network-based features (e.g., number of friends, uploads, likes and so on) in detection systems [87]. For example, Cheng et al. [16] refined the cyberbullying detection problem within a multi-modal context. Then, the problem is a process of multiple modalities exploited in a collaborative fashion.

7.3 Fake reviews

Rating platforms require aggregating a large-scale collection of user reviews and ratings about items (e.g., products, movies, or other users), which play a central role in deciding what service to purchase, restaurant to patronize, and movie to watch, to name but a few. However, fraudulent users give fake ratings and even malicious reviews out of personal interest or prejudice. Therefore, it is necessary to detect such users and eliminate the influence of malicious competition among peers on rating systems.

The algorithm of detecting fraudulent users is formulized as a process to calculate the trustworthiness of a user and its ratings. Networks used in this line of research are diverse, ranging from homogeneous to heterogeneous, such as user-product bipartite network with signed edges [2], homogeneous co-review graph with weighted or unweighted edges [47], and a bipartite rating graph with directed and weighted edges [51].

7.4 Electrical grid

A power grid, also known as an electrical grid, can be constructed as a graph, with nodes denoting generators and edges indicating power lines. Several research questions about anomaly detection or prediction need to be solved in an electrical system. For example, when an electrical component has failed or is going to fail, how could we detect or predict it accurately? Another more challenging problem is to determine the locations of a limited budget of sensors, then it is easier to detect and predict grid anomalies in advance.

Existing anomaly detection algorithms mainly focus on graph theory-based measures instead of graph learning methods. For example, Hooi et al. [38] detected sensor-level anomalies by designing detectors for three common types of anomalies, and constructed an optimization strategy for sensor placement, with the aim of maximizing the probability of detecting an anomaly. Li et al. [55] proposed an index to measure the distance between each past graph and the current graph, thereby generating anomaly scores of a graph in a specific timestamp.

7.5 Financial defaulter

Despite the huge benefits created by online financial services to the society, we have been witnessing a huge growth in financial frauds. The types of frauds in financial scenarios include cash-out behaviors [41], insurance fraud [95], and default users [135]. These frauds severely damage the security of users and service providers, which is a serious problem that needs to be solved.

In financial systems, users engage in interactions and have multiple sources of information. These data form a large multi-view network that conventional methods cannot fully exploit. By integrating the features of various kinds of objects and their interactions, [41] aims to identify whether a user is a cash-out user or not. Default user is defined as a user who is likely to fail to make required payments on time in the future [135]. Hence, these kinds of research questions are generally formulated as binary classification problems.

7.6 Anomalous citations and co-authors

In the context of big scholarly data, the concept of Academic Social Networks (ASNs) is created. ASNs are complex heterogeneous networks formed by academic entities and their relationships [49], such as co-authorship network, co-citation network, and co-word network. Among these complex relationships and interactions,

abnormal academic behaviors (e.g., citations and collaborations) commonly and implicitly exist in different kinds of ASNs [5].

In [46], the authors proposed five heuristic rules to define five types of anomalous citation from the perspective of journal-level citation count. Another kind of anomalous citation is defined in terms of citation context, which is identified by analyzing the context similarity between two publications. As for academic collaboration, Fan et al. [31] analyzed the author-paper-author meta-path (co-authorship) to discover rare pattern events in a heterogeneous information network, where each event is denoted by a specific meta-path. To detect anomalous citations, Liu et al. [58] first applied transfer learning to automatically identify unmarked citation purposes and then, applied a deep graph learning framework for anomalous citation detection.

7.7 Urban computing and mobile sensing

In the process of constructing a smart city, urban anomalies like traffic congestion widely occur and sometimes it may bring serious environmental, economic, and social threats to the public [126]. To prevent tragedies, the use of smart devices and sensors to detect urban anomalies is of great value. Since the urban data are collected in real time through mobile devices or distributed sensors, they are generally modeled as spatial-temporal graphs that have timesteps and location tags.

The emergence of urban big data inspired many novel research on anomaly detection and prediction, such as air quality prediction [132], traffic speed prediction [115], and crime detection [104]. By modeling the urban data as a global cross-region hypergraph, [114] proposed to encode crime dependent representations and spatial temporal dynamics for crime prediction. As for intelligent transportation system, [98] proposed a model based on integration of a modified GCN and LSTM to predict anomaly distribution and duration.

8 FUTURE DIRECTIONS

There are several ongoing or future research directions that are worthy of discussion. In this section, we summarize five potential research directions of anomaly analytics on graph data.

8.1 Anomaly detection on graphs with complex types

Most of the existing research focus on detecting anomalies on simple graphs, while real-world networks are more complicated and have different types, such as heterogeneous graph with multiple node types [124, 129], spatio-temporal graph evolving with time [43], and hypergraph with relations not limited to pairwise relations [97]. Detecting and predicting anomalies on these complex graphs involve technical challenges. For example, as nodes and links which are representing entities and relationships in real-world networks are constantly evolving over time, anomalous entities/relationships might sometimes present normal behaviors as other entities in static networks. This will decrease the accuracy of anomaly detection methods [8]. So, how to model the temporal characteristic of dynamic networks and update real-time graph embeddings remain as important challenges. As for heterogeneous graph anomaly detection, how to incorporate both attribute and structure information of various types of nodes and edges into the graph learning model is a key problem [101]. Therefore, anomaly detection and prediction on complex graphs is still a potential research direction need to be further explored.

8.2 Interpretable and robust anomaly detection algorithms

Despite the fact that representation learning methods relieve much of the cost of handling features manually, a major limitation of current graph embedding approaches is the lack of interpretability. Unlike the general tasks, an interpretable model for anomaly analytics can help people to understand the results, thereby avoiding the potential model risks and human bias. Apart from result visualization and benchmark evaluation, efforts must be devoted to improving the interpretability of graph learning methods from the perspective of neural network

structures. Interpretable models for anomaly analytics can be presented clearly and are likely to be accepted by the public. For example, [71] could identify which neighbor of an anomalous node influenced most by differentiating the edge weights generated by the attention mechanism. Moreover, it is acknowledged that adversarial attacks will influence the model's accuracy and performance. Therefore, how to enhance the robustness of a model is another challenge. Several studies regarding interpretability and robustness can be found in [28, 117, 141].

8.3 Anomalous subgraph detection

Recent years have substantially witnessed superior performance on detecting point anomalies, while users in real-world tend to carry out abnormal behaviors in groups, such as spreading rumors and telecommunications fraud. Apart from this, graph data have diverse structures and forms, while existing methods are not available for all situations. Methods regarding group or subgraph anomaly detection have been less explored [120], especially for complex network structures like hypergraph and multi-modal graph.

8.4 Novel applications of anomaly prediction

While most of the works we reviewed aim to detect existing anomalies, there are still significant works to be done in predicting anomalies in advance. For example, predicting traffic jams ahead of time in transportation networks can help people map out another travel route and avoid congestion situations [48]. Therefore, developing representation learning frameworks that are truly appropriate to anomaly prediction settings in a timely manner is necessary to prevent accidents, huge financial loss, or even deaths.

As a special data structure, graphs are often employed as an auxiliary tool to combine with many research fields, such as biology, chemistry, and social science. Considering that anomalies are defined quite different in various scenarios, domain knowledge is thereby necessary when applying anomaly prediction models into novel applications.

8.5 Fairness in anomaly analysis

Recent years have witnessed a surge of attention in fair machine learning models [59]. Consequently, several fairness metrics have been proposed as the constraints of objective function in various machine learning models to guarantee the equality of the prediction results. As for anomaly detection tasks, whether users can trust the detection results of the models is still a significant problem [125]. It is due to the fact that incorrect anomaly detection results may sometimes lead to serious consequences, such as wrong object detection when dealing with criminals and fraudsters. In [84], the authors formally defined fairness-aware outlier detection problem and proposed a model to satisfy the fairness criteria. However, fairness on graph anomaly detection is still of concern and deserve further attention.

9 CONCLUSION

In this survey, we comprehensively reviewed anomaly analytics methods using graph learning models. The algorithms are divided into four classifications: graph convolutional network-based methods, graph attention network-based methods, graph autoencoder-based methods, and other graph learning models. A thorough comparison and summarization of these methods are provided in this paper. Then, we enumerated and briefly introduced several real-world applications of graph anomaly analytics. Finally, we discussed five future research directions when applying deep learning methods into graph anomaly analytics.

REFERENCES

[1] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P Mathur. 2017. WADI: a water distribution testbed for research in the design of secure cyber physical systems. In *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks*. 25–28.

[2] Leman Akoglu, Rishi Chandy, and Christos Faloutsos. 2013. Opinion fraud detection in online reviews by network effects. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 7. 2–11.

[3] Leman Akoglu, Hanghang Tong, and Danai Koutra. 2015. Graph based anomaly detection and description: A survey. *Data mining and knowledge discovery* 29, 3 (2015), 626–688.

[4] Jinwon An and Sungzoon Cho. 2015. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE* 2, 1 (2015), 1–18.

[5] Xiaomei Bai, Feng Xia, Ivan Lee, Jun Zhang, and Zhaolong Ning. 2016. Identifying anomalous citations for objective evaluation of scholarly article impact. *PloS one* 11, 9 (2016), e0162364.

[6] Sambaran Bandyopadhyay, Saley Vishal Vivek, and MN Murty. 2020. Outlier resistant unsupervised deep architectures for attributed network embedding. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 25–33.

[7] Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. 2020. Rumor detection on social media with bi-directional graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 549–556.

[8] Cemal Cagatay Bilgin and Bülent Yener. 2006. Dynamic network evolution: Models, clustering, anomaly detection. *IEEE Networks* 1 (2006).

[9] Aleksandar Bojchevski and Stephan Günnemann. 2018. Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking. In *International Conference on Learning Representations*. 1–13.

[10] Lei Cai, Zhengzhang Chen, Chen Luo, Jiaping Gui, Jingchao Ni, Ding Li, and Haifeng Chen. 2021. Structural temporal graph neural networks for anomaly detection in dynamic graphs. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3747–3756.

[11] Raghavendra Chalapathy and Sanjay Chawla. 2019. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407* (2019).

[12] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)* 41, 3 (2009), 1–58.

[13] Bo Chen, Jing Zhang, Xiaokang Zhang, Yuxiao Dong, Jian Song, Peng Zhang, Kaibo Xu, Evgeny Kharlamov, and Jie Tang. 2022. GCCAD: Graph Contrastive Learning for Anomaly Detection. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[14] Zhengdao Chen, Lisha Li, and Joan Bruna. 2019. Supervised Community Detection with Line Graph Neural Networks. In *International Conference on Learning Representations*.

[15] Zhenxing Chen, Bo Liu, Meiqing Wang, Peng Dai, Jun Lv, and Liefeng Bo. 2020. Generative Adversarial Attributed Network Anomaly Detection. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1989–1992.

[16] Lu Cheng, Jundong Li, Yasin N Silva, Deborah L Hall, and Huan Liu. 2019. XBully: Cyberbullying detection within a multi-modal context. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 339–347.

[17] Lu Cheng, Kai Shu, Siqi Wu, Yasin N Silva, Deborah L Hall, and Huan Liu. 2020. Unsupervised Cyberbullying Detection via Time-Informed Gaussian Mixture Model. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 185–194.

[18] Munmun De Choudhury, Hari Sundaram, Ajita John, and Dorée Duncan Seligmann. 2009. Social synchrony: Predicting mimicry of user actions in online social media. In *2009 International conference on computational science and engineering*, Vol. 4. IEEE, 151–158.

[19] Ailin Deng and Bryan Hooi. 2021. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4027–4035.

[20] Cristiano Di Crescenzo, Giulia Gavazzi, Giacomo Legnaro, Elena Troccoli, Ilaria Bordino, and Francesco Gullo. 2017. HERMEVENT: a news collection for emerging-event detection. In *Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics*. 1–10.

[21] Kaize Ding, Jundong Li, Nitin Agarwal, and Huan Liu. 2020. Inductive anomaly detection on attributed networks. In *29th International Joint Conference on Artificial Intelligence, IJCAI 2020*. International Joint Conferences on Artificial Intelligence, 1288–1294.

[22] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. 2019. Deep anomaly detection on attributed networks. In *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 594–602.

[23] Kaize Ding, Qinghai Zhou, Hanghang Tong, and Huan Liu. 2021. Few-shot network anomaly detection via cross-network meta-learning. In *Proceedings of the Web Conference 2021*. 2448–2456.

[24] Qi Ding, Natallia Katenka, Paul Barford, Eric Kolaczyk, and Mark Crovella. 2012. Intrusion as (anti) social communication: Characterization and detection. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 886–894.

[25] Kien Do, Truyen Tran, and Svetha Venkatesh. 2019. Graph transformation policy network for chemical reaction prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 750–760.

[26] Ming Dong, Bolong Zheng, Nguyen Quoc Viet Hung, Han Su, and Guohui Li. 2019. Multiple rumor source detection with graph convolutional networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 569–578.

[27] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S Yu. 2020. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 315–324.

[28] Mengnan Du, Ninghao Liu, and Xia Hu. 2019. Techniques for interpretable machine learning. *Commun. ACM* 63, 1 (2019), 68–77.

[29] Dongsheng Duan, Lingling Tong, Yangxi Li, Jie Lu, Lei Shi, and Cheng Zhang. 2020. AANE: Anomaly Aware Network Embedding For Anomalous Link Detection. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1002–1007.

[30] Heba El-Fiqi, Min Wang, Kathryn Kasmarik, Anastasios Bezerianos, Kay Chen Tan, and Hussein A Abbass. 2021. Weighted Gate Layer Autoencoders. *IEEE Transactions on Cybernetics* 52, 8 (2021), 7242–7253.

[31] Shaohua Fan, Chuan Shi, and Xiao Wang. 2018. Abnormal event detection via heterogeneous information network embedding. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1483–1486.

[32] Yujie Fan, Yanfang Ye, Qian Peng, Jianfei Zhang, Yiming Zhang, Xusheng Xiao, Chuan Shi, Qi Xiong, Fudong Shao, and Liang Zhao. 2020. Metagraph Aggregated Heterogeneous Graph Neural Network for Illicit Traded Product Identification in Underground Market. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 132–141.

[33] Ruth Festl and Thorsten Quandt. 2013. Social relations and cyberbullying: The influence of individual and structural attributes on victimization and perpetration via the internet. *Human communication research* 39, 1 (2013), 101–126.

[34] Suyu Ge, Lu Cheng, and Huan Liu. 2021. Improving cyberbullying detection with user interaction. In *Proceedings of the Web Conference 2021*. 496–506.

[35] Frank E Grubbs. 1969. Procedures for detecting outlying observations in samples. *Technometrics* 11, 1 (1969), 1–21.

[36] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 1025–1035.

[37] Jack Hessel and Lillian Lee. 2019. Something¡¯s Brewing! Early Prediction of Controversy-causing Posts from Discussion Features. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1648–1659.

[38] Bryan Hooi, Dhivya Eswaran, Hyun Ah Song, Amritanshu Pandey, Marko Jereminov, Larry Pileggi, and Christos Faloutsos. 2018. GRIDWATCH: Sensor placement and anomaly detection in the electrical grid. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 71–86.

[39] Homa Hosseinmardi, Sabrina Arredondo Mattson, Rahat Ibn Rafiq, Richard Han, Qin Lv, and Shivakant Mishra. 2015. Analyzing labeled cyberbullying incidents on the instagram social network. In *International conference on social informatics*. Springer, 49–66.

[40] Mingliang Hou, Jing Ren, Da Zhang, Xiangjie Kong, Dongyu Zhang, and Feng Xia. 2020. Network embedding: Taxonomies, frameworks and applications. *Computer Science Review* 38 (2020), 100296.

[41] Binbin Hu, Zhiqiang Zhang, Chuan Shi, Jun Zhou, Xiaolong Li, and Yuan Qi. 2019. Cash-out user detection based on attributed heterogeneous information network with a hierarchical attention mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 946–953.

[42] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.

[43] Hong Huang, Ruize Shi, Wei Zhou, Xiao Wang, Hai Jin, and Xiaoming Fu. 2021. Temporal Heterogeneous Information Network Embedding.. In *IJCAI*. 1470–1476.

[44] Meng Jiang. 2016. Catching social media advertisers with strategy analysis. In *Proceedings of the First International Workshop on Computational Methods for CyberSafety*. 5–10.

[45] Ming Jin, Yixin Liu, Yu Zheng, Lianhua Chi, Yuan-Fang Li, and Shirui Pan. 2021. ANEMONE: Graph Anomaly Detection with Multi-Scale Contrastive Learning. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3122–3126.

[46] Baani Leen Kaur Jolly, Lavina Jain, Debajyoti Bera, and Tanmoy Chakraborty. 2020. Unsupervised anomaly detection in journal-level citation networks. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*. 27–36.

[47] Parisa Kaghazgaran, James Caverlee, and Anna Squicciarini. 2018. Combating crowdsourced review manipulators: A neighborhood-based approach. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 306–314.

[48] Fanhui Kong, Jian Li, Bin Jiang, and Houbing Song. 2019. Short-term traffic flow prediction in smart multimedia system for Internet of Vehicles based on deep belief network. *Future Generation Computer Systems* 93 (2019), 460–472.

[49] Xiangjie Kong, Yajie Shi, Shuo Yu, Jiaying Liu, and Feng Xia. 2019. Academic social networks: Modeling, analysis, mining and applications. *Journal of Network and Computer Applications* 132 (2019), 86–103.

[50] Xiangjie Kong, Ximeng Song, Feng Xia, Haochen Guo, Jinzhong Wang, and Amr Tolba. 2018. LoTAD: Long-term traffic anomaly detection based on crowdsourced bus trajectory data. *World Wide Web* 21, 3 (2018), 825–847.

[51] Srijan Kumar, Bryan Hooi, Disha Makhija, Mohit Kumar, Christos Faloutsos, and VS Subrahmanian. 2018. Rev2: Fraudulent user prediction in rating platforms. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 333–341.

[52] Srijan Kumar, Francesca Spezzano, VS Subrahmanian, and Christos Faloutsos. 2016. Edge weight prediction in weighted signed networks. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 221–230.

[53] Kyumin Lee, Brian Eoff, and James Caverlee. 2011. Seven months with the devils: A long-term study of content polluters on twitter. In *Proceedings of the international AAAI conference on web and social media*, Vol. 5. 185–192.

[54] Ao Li, Zhou Qin, Runshi Liu, Yiqun Yang, and Dong Li. 2019. Spam review detection with graph convolutional networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2703–2711.

[55] Shimiao Li, Amritanshu Pandey, Bryan Hooi, Christos Faloutsos, and Larry Pileggi. 2022. Dynamic Graph-Based Anomaly Detection in the Electrical Grid. *IEEE Transactions on Power Systems* 37, 5 (2022), 3408–3422.

[56] Yuening Li, Xiao Huang, Jundong Li, Mengnan Du, and Na Zou. 2019. SpecAE: Spectral autoencoder for anomaly detection in attributed networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2233–2236.

[57] Jiaying Liu, Xiangjie Kong, Feng Xia, Xiaomei Bai, Lei Wang, Qing Qing, and Ivan Lee. 2018. Artificial intelligence in the 21st century. *IEEE Access* 6 (2018), 34403–34421.

[58] Jiaying Liu, Feng Xia, Xu Feng, Jing Ren, and Huan Liu. 2022. Deep Graph Learning for Anomalous Citation Detection. *IEEE Transactions on Neural Networks and Learning Systems* 33, 6 (2022), 2543–2557.

[59] Lydia T Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. 2018. Delayed impact of fair machine learning. In *International Conference on Machine Learning*. PMLR, 3150–3158.

[60] Yixin Liu, Zhao Li, Shirui Pan, Chen Gong, Chuan Zhou, and George Karypis. 2021. Anomaly detection on attributed networks via contrastive self-supervised learning. *IEEE transactions on neural networks and learning systems* 33, 6 (2021), 2378–2392.

[61] Yixin Liu, Shirui Pan, Yu Guang Wang, Fei Xiong, Liang Wang, Qingfeng Chen, and Vincent CS Lee. 2021. Anomaly detection in dynamic graphs via transformer. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[62] Ziqi Liu, Chaochao Chen, Xinxing Yang, Jun Zhou, Xiaolong Li, and Le Song. 2018. Heterogeneous graph neural networks for malicious account detection. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2077–2085.

[63] Yi-Ju Lu and Cheng-Te Li. 2020. 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*. 505–514.

[64] Weixin Luo, Wen Liu, and Shenghua Gao. 2017. A revisit of sparse coding based anomaly detection in stacked RNN framework. In *Proceedings of the IEEE International Conference on Computer Vision*. 341–349.

[65] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks. In *25th International Joint Conference on Artificial Intelligence, IJCAI 2016*. International Joint Conferences on Artificial Intelligence, 3818–3824.

[66] Jing Ma, Wei Gao, and Kam-Fai Wong. 2017. Detect Rumors in Microblog Posts Using Propagation Structure via Kernel Learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 708–717.

[67] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z Sheng, Hui Xiong, and Leman Akoglu. 2021. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering* (2021). https://doi.org/10.1109/TKDE.2021.3118815

[68] Amir Markovitz, Gilad Sharir, Itamar Friedman, Lihi Zelnik-Manor, and Shai Avidan. 2020. Graph embedded pose clustering for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10539–10547.

[69] Aditya P Mathur and Nils Ole Tippenhauer. 2016. SWaT: A water treatment testbed for research and training on ICS security. In *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*. IEEE, 31–36.

[70] Julian John McAuley and Jure Leskovec. 2013. From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd international conference on World Wide Web*. 897–908.

[71] Van-Hoang Nguyen, Kazunari Sugiyama, Preslav Nakov, and Min-Yen Kan. 2020. Fang: Leveraging social context for fake news detection using graph representation. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 1165–1174.

[72] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. 2016. Learning convolutional neural networks for graphs. In *International conference on machine learning*. PMLR, 2014–2023.

[73] Tore Opsahl and Pietro Panzarasa. 2009. Clustering in weighted networks. *Social networks* 31, 2 (2009), 155–163.

[74] Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. 2018. Adversarially regularized graph autoencoder for graph embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 2609–2615.

[75] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. 2021. Deep learning for anomaly detection: A review. *ACM Computing Surveys (CSUR)* 54, 2 (2021), 1–38.

[76] Bryan Perozzi, Leman Akoglu, Patricia Iglesias Sánchez, and Emmanuel Müller. 2014. Focused clustering and outlier detection in large attributed graphs. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1346–1355.

[77] Rahat Ibn Rafiq, Homa Hosseinmardi, Richard Han, Qin Lv, Shivakant Mishra, and Sabrina Arredondo Mattson. 2015. Careful what you share in six seconds: Detecting cyberbullying instances in Vine. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 617–622.

[78] Stephen Ranshous, Shitian Shen, Danai Koutra, Steve Harenberg, Christos Faloutsos, and Nagiza F Samatova. 2015. Anomaly detection in dynamic networks: A survey. *Wiley Interdisciplinary Reviews: Computational Statistics* 7, 3 (2015), 223–247.

[79] Shebuti Rayana and Leman Akoglu. 2015. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*. 985–994.

[80] Jing Ren, Feng Xia, Xiangtai Chen, Jiaying Liu, Mingliang Hou, Ahsan Shehzad, Nargiz Sultanova, and Xiangjie Kong. 2021. Matching Algorithms: Fundamentals, Applications and Challenges. *IEEE Transactions on Emerging Topics in Computational Intelligence* 5, 3 (2021), 332–350.

[81] Yuxiang Ren, Bo Wang, Jiawei Zhang, and Yi Chang. 2020. Adversarial Active Learning based Heterogeneous Graph Neural Network for Fake News Detection. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 452–461.

[82] Semiu Salawu, Yulan He, and Joanna Lumsden. 2017. Approaches to automated detection of cyberbullying: A survey. *IEEE Transactions on Affective Computing* 11, 1 (2017), 3–24.

[83] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI magazine* 29, 3 (2008), 93–93.

[84] Shubhranshu Shekhar, Neil Shah, and Leman Akoglu. 2021. Fairod: Fairness-aware outlier detection. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. 210–220.

[85] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S Yu Philip. 2016. A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering* 29, 1 (2016), 17–37.

[86] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter* 19, 1 (2017), 22–36.

[87] Anna Squicciarini, Sarah Rajtmajer, Y Liu, and Christopher Griffin. 2015. Identification and characterization of cyberbullying dynamics in an online social network. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*. 280–285.

[88] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. ArnetMiner: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 990–998.

[89] Lei Tang and Huan Liu. 2009. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 817–826.

[90] Xian Teng, Muheng Yan, Ali Mert Ertugrul, and Yu-Ru Lin. 2018. Deep into hypersphere: robust and unsupervised anomaly discovery in dynamic networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 2724–2730.

[91] Fei Tian, Bin Gao, Qing Cui, Enhong Chen, and Tie-Yan Liu. 2014. Learning deep representations for graph clustering. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, Vol. 28. 1293–1299.

[92] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of the 7th International Conference on Learning Representations*.

[93] Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science* 359, 6380 (2018), 1146–1151.

[94] Liangtian Wan, Yuyuan Yuan, Feng Xia, and Huan Liu. 2019. To your surprise: Identifying serendipitous collaborators. *IEEE Transactions on Big Data* 7, 3 (2019), 574–589.

[95] Daixin Wang, Jianbin Lin, Peng Cui, Quanhui Jia, Zhen Wang, Yanming Fang, Quan Yu, Jun Zhou, Shuang Yang, and Yuan Qi. 2019. A semi-supervised graph attentive network for financial fraud detection. In *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 598–607.

[96] Hongwei Wang, Jialin Wang, Jia Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Learning graph representation with generative adversarial nets. *IEEE Transactions on Knowledge and Data Engineering* 33, 8 (2019), 3090–3103.

[97] Jingcheng Wang, Yong Zhang, Yun Wei, Yongli Hu, Xinglin Piao, and Baocai Yin. 2021. Metro passenger flow prediction via dynamic hypergraph convolution networks. *IEEE Transactions on Intelligent Transportation Systems* 22, 12 (2021), 7891–7903.

[98] Kun Wang, Zhengyang Zhou, Xu Wang, Pengkun Wang, Qi Fang, and Yang Wang. 2022. A2DJP: A Two Graph-based Component Fused Learning Framework for Urban Anomaly Distribution and Duration Joint-Prediction. *IEEE Transactions on Knowledge and Data Engineering* (2022). https://doi.org/10.1109/TKDE.2022.3176650

[99] Min Wang, Heba El-Fiqi, Jiankun Hu, and Hussein A Abbass. 2019. Convolutional neural networks using dynamic functional connectivity for EEG-based person identification in diverse human states. *IEEE Transactions on Information Forensics and Security* 14, 12 (2019), 3259–3272.

[100] Shen Wang, Zhengzhang Chen, Xiao Yu, Ding Li, Jingchao Ni, Lu-An Tang, Jiaping Gui, Zhichun Li, Haifeng Chen, and Philip S Yu. 2019. Heterogeneous Graph Matching Networks for Unknown Malware Detection. In *28th International Joint Conference on Artificial Intelligence, IJCAI 2019*. International Joint Conferences on Artificial Intelligence, 3762–3770.

[101] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The world wide web conference*. 2022–2032.

[102] Xuhong Wang, Baihong Jin, Ying Du, Ping Cui, Yingshui Tan, and Yupu Yang. 2021. One-class graph neural networks for anomaly detection in attributed networks. *Neural computing and applications* 33, 18 (2021), 12073–12085.

[103] Xufei Wang, Lei Tang, Huiji Gao, and Huan Liu. 2010. Discovering overlapping groups in social media. In *2010 IEEE international conference on data mining*. IEEE, 569–578.

[104] Yaqian Wang, Liang Ge, Siyu Li, and Feng Chang. 2020. Deep Temporal Multi-Graph Convolutional Network for Crime Prediction. In *International Conference on Conceptual Modeling*. Springer, 525–538.

[105] Yanling Wang, Jing Zhang, Shasha Guo, Hongzhi Yin, Cuiping Li, and Hong Chen. 2021. Decoupling representation learning and classification for gnn-based anomaly detection. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1239–1248.

[106] Zheng Wang, Chaokun Wang, Jisheng Pei, and Xiaojun Ye. 2017. Multiple source detection without knowing the underlying propagation model. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. 217–223.

[107] Max Welling and Thomas N Kipf. 2017. Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*.

[108] Yongji Wu, Defu Lian, Yiheng Xu, Le Wu, and Enhong Chen. 2020. Graph convolutional networks with markov random field reasoning for social spammer detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 1054–1061.

[109] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.

[110] Feng Xia, Behrouz Jedari, Laurence Tianruo Yang, Jianhua Ma, and Runhe Huang. 2016. A signaling game for uncertain data delivery in selfish mobile social networks. *IEEE Transactions on Computational Social Systems* 3, 2 (2016), 100–112.

[111] Feng Xia, Jiaying Liu, Hansong Nie, Yonghao Fu, Liangtian Wan, and Xiangjie Kong. 2019. Random walks: A review of algorithms and applications. *IEEE Transactions on Emerging Topics in Computational Intelligence* 4, 2 (2019), 95–107.

[112] Feng Xia, Ke Sun, Shuo Yu, Abdul Aziz, Liangtian Wan, Shirui Pan, and Huan Liu. 2021. Graph Learning: A Survey. *IEEE Transactions on Artificial Intelligence* 2, 2 (2021), 109–127.

[113] Feng Xia, Wei Wang, Teshome Megersa Bekele, and Huan Liu. 2017. Big scholarly data: A survey. *IEEE Transactions on Big Data* 3, 1 (2017), 18–35.

[114] Lianghao Xia, Chao Huang, Yong Xu, Peng Dai, Liefeng Bo, Xiyue Zhang, and Tianyi Chen. 2021. Spatial-Temporal Sequential Hypergraph Network for Crime Prediction with Dynamic Multiplex Relation Learning. In *IJCAI*. 1631–1637.

[115] Qinge Xie, Tiancheng Guo, Yang Chen, Yu Xiao, Xin Wang, and Ben Y Zhao. 2020. Deep graph convolutional networks for incident-driven traffic speed prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1665–1674.

[116] Chao Yang, Robert Harkreader, Jialong Zhang, Seungwon Shin, and Guofei Gu. 2012. Analyzing spammers' social networks for fun and profit: a case study of cybercriminal ecosystem on Twitter. In *Proceedings of the 21st international conference on World Wide Web*. 71–80.

[117] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. GNNExplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems* 32 (2019), 9240.

[118] Shuo Yu, Jiaying Liu, Haoran Wei, Feng Xia, and Hanghang Tong. 2020. How to optimize an academic team when the outlier member is leaving? *IEEE Intelligent Systems* 36, 3 (2020), 23–30.

[119] Shuo Yu, Jing Ren, Shihao Li, Mehdi Naseriparsa, and Feng Xia. 2022. Graph Learning for Fake Review Detection. *Frontiers in Artificial Intelligence* 5 (2022), 229–270.

[120] Shuo Yu, Feng Xia, Yuchen Sun, Tao Tang, Xiaoran Yan, and Ivan Lee. 2020. Detecting Outlier Patterns With Query-Based Artificially Generated Searching Conditions. *IEEE Transactions on Computational Social Systems* 8, 1 (2020), 134–147.

[121] Reza Zafarani, Xinyi Zhou, Kai Shu, and Huan Liu. 2019. Fake news research: Theories, detection strategies, and open problems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3207–3208.

[122] Mohamad Zamini and Seyed Mohammad Hossein Hasheminejad. 2019. A comprehensive survey of anomaly detection in banking, wireless sensor networks, social networks, and healthcare. *Intelligent Decision Technologies* 13, 2 (2019), 229–270.

[123] An Zeng, Zhesi Shen, Jianlin Zhou, Jinshan Wu, Ying Fan, Yougui Wang, and H Eugene Stanley. 2017. The science of science: From the perspective of complex systems. *Physics Reports* 714 (2017), 1–73.

[124] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 793–803.

[125] Hongjing Zhang and Ian Davidson. 2021. Towards fair deep anomaly detection. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 138–148.

[126] Mingyang Zhang, Tong Li, Yue Yu, Yong Li, Pan Hui, and Yu Zheng. 2020. Urban Anomaly Analytics: Description, Detection, and Prediction. *IEEE Transactions on Big Data* 8, 3 (2020), 809–826.

[127] Shijie Zhang, Hongzhi Yin, Tong Chen, Quoc Viet Nguyen Hung, Zi Huang, and Lizhen Cui. 2020. GCN-Based User Representation Learning for Unifying Robust Recommendation and Fraudster Detection. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 689–698.

[128] Ziwei Zhang, Peng Cui, and Wenwu Zhu. 2020. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering* 34, 1 (2020), 249–270.

[129] Jianan Zhao, Xiao Wang, Chuan Shi, Binbin Hu, Guojie Song, and Yanfang Ye. 2021. Heterogeneous graph structure learning for graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4697–4705.

[130] Tong Zhao, Chuchen Deng, Kaifeng Yu, Tianwen Jiang, Daheng Wang, and Meng Jiang. 2020. Error-Bounded Graph Anomaly Loss for GNNs. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1873–1882.

[131] Li Zheng, Zhenpeng Li, Jian Li, Zhao Li, and Jun Gao. 2019. AddGraph: Anomaly Detection in Dynamic Graph Using Attention-based Temporal GCN.. In *IJCAI*. 4419–4425.

[132] Yu Zheng, Furui Liu, and Hsun-Ping Hsieh. 2013. U-air: When urban air quality inference meets big data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1436–1444.

[133] Jia-Xing Zhong, Nannan Li, Weijie Kong, Shan Liu, Thomas H Li, and Ge Li. 2019. Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1237–1246.

[134] Lei Zhong, Juan Cao, Qiang Sheng, Junbo Guo, and Ziang Wang. 2020. Integrating Semantic and Structural Information with Graph Convolutional Network for Controversy Detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 515–526.

[135] Qiwei Zhong, Yang Liu, Xiang Ao, Binbin Hu, Jinghua Feng, Jiayu Tang, and Qing He. 2020. Financial defaulter detection on online credit payment via multi-view attributed heterogeneous information network. In *Proceedings of The Web Conference 2020*. 785–795.

[136] Fan Zhou, Chengtai Cao, Kunpeng Zhang, Goce Trajcevski, Ting Zhong, and Ji Geng. 2019. Meta-gnn: On few-shot node classification in graph meta-learning. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2357–2360.

[137] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open* 1 (2020), 57–81.

[138] Xinyi Zhou and Reza Zafarani. 2020. A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys (CSUR)* 53, 5 (2020), 1–40.

[139] Xinyi Zhou, Reza Zafarani, Kai Shu, and Huan Liu. 2019. Fake news: Fundamental theories, detection strategies and challenges. In *Proceedings of the twelfth ACM international conference on web search and data mining*. 836–837.

[140] Dali Zhu, Yuchen Ma, and Yinlong Liu. 2020. DeepAD: A Joint Embedding Approach for Anomaly Detection on Attributed Networks. In *International Conference on Computational Science*. Springer, 294–307.

[141] Dingyuan Zhu, Ziwei Zhang, Peng Cui, and Wenwu Zhu. 2019. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1399–1407.

[142] Yong-Nan Zhu, Xiaotian Luo, Yu-Feng Li, Bin Bu, Kaibo Zhou, Wenbin Zhang, and Mingfan Lu. 2020. Heterogeneous Mini-Graph Neural Network and Its Application to Fraud Invitation Detection. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 891–899.