Anomaly Detection in Multiplex Dynamic Networks: from Blockchain Security to Brain Disease Prediction

Ali Behrouz
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada
alibez@cs.ubc.ca

Margo Seltzer
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada
mseltzer@cs.ubc.ca

Abstract

The problem of identifying anomalies in dynamic networks is a fundamental task with a wide range of applications. However, it raises critical challenges due to the complex nature of anomalies, lack of ground truth knowledge, and complex and dynamic interactions in the network. Most existing approaches usually study networks with a single type of connection between vertices, while in many applications interactions between objects vary, yielding multiplex networks. We propose ANOMULY, a general, unsupervised edge anomaly detection framework for multiplex, dynamic networks. In each relation type, ANOMULY sees node embeddings at different GNN layers as hierarchical node states and employs a GRU cell to capture temporal properties of the network and update node embeddings over time. We then add an attention mechanism that incorporates information across different types of relations. Our case study on brain networks shows how this approach could be employed as a new tool to understand abnormal brain activity that might reveal a brain disease or disorder. Extensive experiments on nine real-world datasets demonstrate that ANOMULY achieves state-of-the-art performance.

1 Introduction

Identifying anomalous activities in networks is a long-standing and vital problem with a wide variety of applications in different domains, e.g., finance, social networks, security, and public health [1, 2, 3, 4, 5]. While several anomaly detection approaches focus on the topological properties of networks [6, 7, 8, 9, 10, 11], detecting anomalies in real-world networks also requires attention to their dynamic nature [5]. Anomalies might appear as malware in computer systems [12], social bots and social spammers in social networks [13], or financial fraud in financial systems [1, 14]. Accordingly, anomaly detection in dynamic (evolving) complex systems has recently attracted much attention.

Most prior work focuses on detecting anomalies in dynamic networks whose edges are all of the same type [5, 15, 16, 17, 18, 19, 20, 21]; these networks are called single-layer, dynamic networks. However, in many complex dynamic systems, there are many different kinds of interactions between objects. For example, interactions between people can be social or professional, and professional interactions can differ according to topics. We model graphs with different kinds of edges as Multilayer or Multiplex networks [22]. In these networks, the different types of connections are complementary to each other, providing more complex and richer information than simple graphs. Surprisingly, anomaly detection in multiplex networks is relatively less explored and has only recently attracted attention.

Existing approaches to anomaly detection in multiplex networks suffer from three main limitations: **(1)** Structure and feature inflexibility: existing methods assume pre-defined anomaly patterns or man-made features. Such approaches are application dependent and do not easily generalize to

NeurIPS 2022 Temporal Graph Learning Workshop held in New Orleans, United States

different domains. Moreover, in real-world networks, anomalies might be more complex in nature, and it is nearly impossible to detect anomalies with high accuracy using pre-defined patterns/roles. **(2)** Same importance for all type of connections: these methods treat each relation type (i.e., layer) identically, assigning the same importance to each layer. However, real-world multiplex networks can contain noisy/insignificant layers [23, 24]. Moreover, all vertices might not participate equally in all layers, so which layers are noisy/insignificant can be different for each vertex [23, 24]. **(3)** Lack of edge anomaly detection: previous methods for anomaly detection in multiplex networks focus on identifying anomalous nodes, subgraphs, or events. However, in many real-world applications, a connection between two vertices might be an anomaly [15, 17, 19, 20]. This anomalous connection might be a suspicious transaction in a financial network, a fake follower in a social network, or an abnormal functional correlation between two regions of the brain.

Existing methods for anomaly detection in single-layer dynamic graphs also exhibit limitations. **(1)** Structure inflexibility: even in single layer networks, most existing anomaly detection methods for dynamic networks rely on pre-defined patterns or heuristic rules (see [25, 26]). These heuristic rules are usually content features or long-term temporal factors. However, due the complex nature of real-world anomalies, these factors are not flexible and are restricted to specific patterns. **(2)** Memory usage: deep learning based methods [19, 20], which are commonly proposed, require storing entire snapshots of the network at each time window, consuming large and increasing amounts of memory.

To mitigate the above limitations, we introduce ANOMULY (**Ano**maly Detection in **Mul**tiplex D**y**namic Networks). To take advantage of both temporal properties and complementary information present in multiple relation (edge) types, ANOMULY extends the idea of *hierarchical node states* [27] to multiplex dynamic networks by using an attention mechanism that incorporates information about different relation types. Next, it uses selective negative sampling to learn anomalous edges in an unsupervised manner. To the best of our knowledge, ANOMULY is the first edge-anomaly detection method for multiplex networks. Further, when it is possible to model a simple network as a multiplex network, ANOMULY outperform existing simple network approaches, because the multiplex network provides richer and more complex information than does a simple network [3, 4, 5].

Consider the following two applications for anomaly detection in dynamic multiplex networks:

Applications: Brain Networks. Monitoring functional systems in the human brain is a fundamental task in neuroscience [28, 29]. Each node in a brain network represents a region of interest (ROI), which is responsible for a specific function, and edges represent high functional correlation between two ROIs. A temporal brain network is usually derived from functional magnetic resonance imaging (fMRI), which lets us measure the statistical association between the functionality of ROIs over time. Since a (dynamic) brain network generated from an individual can be noisy and incomplete [30, 31], prior work used the average of brain networks from many individuals [32, 33]. However, these methods ignore the complex relationships in each individual's brain. We can capture these missing relationships by modeling the network as a multiplex (dynamic) network [34, 31], where each layer represents an individual's brain network. We show that ANOMULY detects abnormal connections in the brains of people with attention deficit hyperactivity disorder (ADHD). (see section 4).

Applications: Fraud Detection in Multiple Blockchain Networks. Anomaly detection in (dynamic) blockchain transaction networks has recently attracted enormous attention [35, 36, 37, 38, 39, 40], due to the emergence of a huge assortment of financial systems' applications [41, 42, 43]. While most existing work focuses on detecting illicit activity in a single blockchain network, recent research shows that cryptocurrency criminals increasingly employ cross-cryptocurrency trades to hide their identity [44, 45]. Accordingly, Yousaf et al. [46] have recently shown that analyzing links across several blockchain networks is critical for identifying emerging criminal activity on the blockchain. An edge anomaly detection approach in multiplex networks can be employed to detect suspicious transactions and identify criminal activities across several blockchain transaction networks more accurately.

The contributions of this work are: **(1)** We present a novel layer-aware node embedding approach in multiplex dynamic networks, *Snapshot Encoder*, which uses an attention mechanism to incorporate both temporal and structural information on different relation types. **(2)** We present ANOMULY, a general end-to-end unsupervised learning method for anomalous edge detection in multiplex dynamic networks, using a GRU cell to incorporate the outputs of *Snapshot Encoder* for different snapshots. **(3)** We demonstrate a new application of edge-anomaly detection in dynamic multiplex networks and present a case study on brain networks of people living with attention deficit hyperactivity disorder

(ADHD). Our results show the effectiveness and usefulness of ANOMULY in identifying abnormal connections of different ROIs of the human brain. This approach could be employed as a new tool to understand abnormal brain activity that might reveal a disease or disorder. **(4)** We conduct extensive experiments on nine real-world multiplex and simple networks. Results show the superior performance of ANOMULY in both single-layer and multiplex networks.

2 Related Work

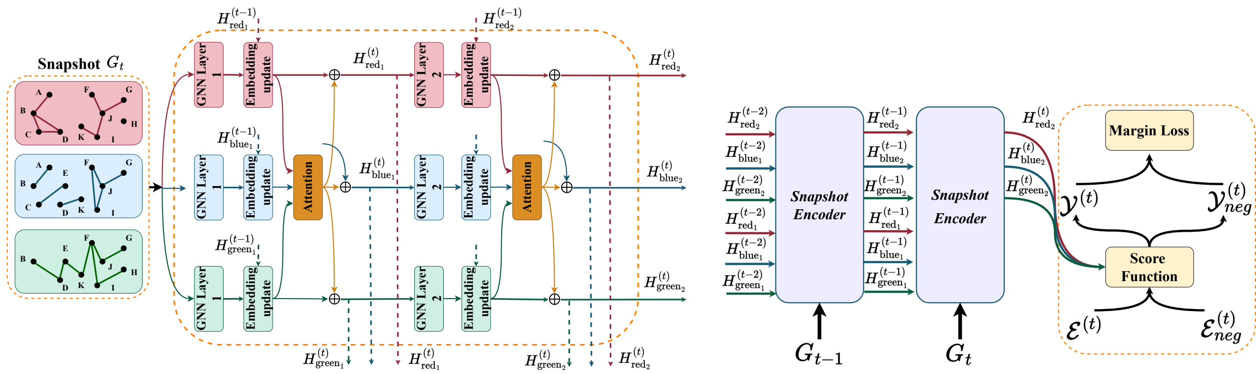
We begin with a brief review of anomaly detection algorithms in dynamic simple networks, then methods for dynamic and multiplex graph learning, and finally anomaly detection in multiplex networks. To situate our research in a broader context, we discuss work on anomaly detection in brain and blockchain networks (Appendix A). For additional related work, we refer the reader to the extensive survey by Ma et al. [4].

Anomaly Detection in Dynamic Networks. There has been much work in anomaly detection for single-layer dynamic networks. That work falls in five categories: **(1)** Probabilistic methods [15, 47, 48, 49, 50, 51, 52] that identify anomalies based on deviations from regular communication patterns. **(2)** Distance-based methods [16, 18, 25, 26, 53] that use certain time-evolving measures of dynamic network structures and use their changes to detect anomalies. **(3)** Density-based methods [54, 55, 56] that view anomalies as subgraphs with high density or as subgraphs with sudden changes in their density. **(4)** Matrix factorization methods [57, 58, 59, 60, 61] that use the low-rank property of network structures and define anomalies as breakers of this property. **(5)** Learning-based methods [19, 20, 62, 63, 64, 65], that combine the graph embedding method into the anomaly detection approach. These learning-based models must store the entire snapshot, which requires large memory, limiting their scalability. To show the effectiveness of ANOMULY in even single-layer networks, we compare it with NETWALK[20], and ADDGRAPH [19] in section 4. All these methods apply to single-layer networks only and do not naturally extend to multiplex networks. We further discuss the novelty of the architecture of our approach in Appendix A.

Dynamic Graph Neural Networks. The problem of learning from dynamic networks has been extensively studied in the literature [66, 67, 68, 69, 70, 71, 72, 73]. The first group of existing methods use Recurrent Neural Networks (RNN) and then replace the linear layer with a graph convolution layer [72, 74, 75]. The second group uses a GNN as a feature encoder and then deploys a sequence model on top of the GNN to encode temporal properties [67, 76, 77]. However, all these models have limitations in both model design and training strategy [27]. To address these limitations, You et al. [27] proposed ROLAND, a graph learning framework for dynamic graphs that can re-purpose any static GNN to dynamic graphs. However, this framework cannot be used for graphs with different types of edges (multiplex networks). Our work extends ROLAND to multiplex networks and introduces an attention mechanism that incorporates the relation-specific hierarchical node states in each snapshot, taking advantage of additional information present in multiplex networks.

Multiplex Graph Learning. In a multiplex network, also known as multilayer, multi-view, or multi-dimensional networks, all nodes have the same type, but edges (relations) have multiple types [22]. Several methods have been proposed to learn network embeddings on multiplex networks by integrating information from individual relation type [78, 79, 80, 81, 82, 83]. Other work proposed Graph Convolutional Networks (GCN) methods for multiplex networks [84, 85, 86]. Inspired by Deep Graph Infomax [87], Park et al. [88] and Jing et al. [89] proposed unsupervised approaches to learn node embeddings by maximizing the mutual information between local patches and the global representation of the entire graph. Zhang et al. [90] proposed a method that uses a latent space to integrate the information across multiple views. Recently, Wang et al. [91] proposed DPMNE to learn from incomplete multiplex networks.

Anomaly Detection in Multiplex Networks. The problem of anomaly detection in multiplex networks has recently attracted attention. Mittal and Bhatia [92] use eigenvector centrality, page rank centrality, and degree centrality as handcrafted features for nodes to detect anomalies in static multiplex networks. Bindu et al. [93] proposed a node anomaly detection algorithm in static multiplex networks that uses handcrafted features based on clique/near-clique and star/near-star structures. Bansal and Sharma [94] defined a quality measure, Multi-Normality, which employs the structure and attributes together of each layer to detect attribute coherence in neighborhoods between layers.



(a) Architecture of the *Snapshot Encoder* (b) ANOMULY Framework

Figure 1: Framework and design of ANOMULY model.

Maulana and Atzmueller [95] use centrality of all nodes in each layer and apply many-objective optimization with full enumeration based on minimization to obtain Pareto Front. Then, they use Pareto Front as a basis for finding suspected anomaly nodes. Chen et al. [96] proposed ANOMMAN that uses an auto-encoder module and a GCN-based decoder to detect node anomalies in static multiplex networks. Although this model can learn from the data, it is limited to static networks, and it treats each layer equally in the *Structure Reconstruction* step. Finally, Ofori-Boateng et al. [45] developed a new persistence summary and used it to detect events in dynamic multiplex blockchain networks.

All of these approaches are designed to detect topological anomalous subgraphs, nodes, or events, and cannot identify anomalous edges. Moreover, as we discussed in section 1, these methods, except ANOMAN [96], are based on pre-defined patterns/roles or handcrafted features, while real-world network anomalies have complex nature. Therefore, these models cannot be generalized to different domains, limiting their application.

3 Proposed Method: ANOMULY

3.1 Preliminaries

We first precisely define multiplex dynamic networks, and then we formalize the problem of edge anomaly detection in multiplex dynamic networks.

DEFINITION 1 (MULTIPLEX DYNAMIC NETWORKS). *Let $\mathcal{G} = \{G_r\}_{r=1}^{\mathcal{L}} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$ denotes a multiplex dynamic network, where $G_r = (\mathcal{V}, \mathcal{E}_r, \mathcal{X})$ is a graph of the relation type $r \in \mathcal{L}$ (aka layer), $\mathcal{V}$ is the set of nodes, $\mathcal{E} = \bigcup_{r=1}^{\mathcal{L}} \mathcal{E}_r$ is the set of edges, and $\mathcal{X} \in \mathbb{R}^{|\mathcal{V}| \times f}$ is a matrix that encodes node attribute information for nodes in $\mathcal{V}$. Given a relation type r, we denote the set of vertices in the neighborhood of $u \in \mathcal{V}$ in relation r as $\mathcal{N}_r(u)$. Each edge $e = (u, v, r_e) \in \mathcal{E}$ is associated with an edge type (layer) $r_e \in \mathcal{L}$ and a timestamp τ_e, and each node $v \in \mathcal{V}$ is associated with a timestamp τ_v.*

We take a snapshot-based anomaly detection approach for multiplex dynamic networks: a multiplex dynamic graph $\mathcal{G} = \{\mathcal{G}^{(t)}\}_{t=1}^{T}$ can be represented as a sequence of multiplex network snapshots, where each snapshot is a static multiplex graph $\mathcal{G}^{(t)} = \{G_r^{(t)}\}_{r=1}^{\mathcal{L}} = (\mathcal{V}^{(t)}, \mathcal{E}^{(t)}, \mathcal{X}^{(t)})$ with $\mathcal{V}^{(t)} = \{v \in \mathcal{V} | \tau_v = t\}$ and $\mathcal{E}^{(t)} = \{e \in \mathcal{E} | \tau_e = t\}$. Our goal is to detect anomalous edges in $\mathcal{E}^{(t)}$. Specifically, for each edge $e = (u, v, r) \in \mathcal{E}^{(t)}$, we produce a layer-dependent anomalous probability $\varphi_r(e)$ in layer $r \in \mathcal{L}$.

3.2 ANOMULY Framework

We now introduce our framework for edge anomaly detection in dynamic networks with multiple types of interactions. Figure 1 provides an overview of the framework. To learn the pattern of normal edges, ANOMULY uses the *Snapshot Encoder* architecture to encode each snapshot of the network. *Snapshot*

Encoder uses GNNs and incorporates structural and temporal properties of the graph as well as node features in each layer. Next, it uses an attention mechanism to take advantage of complementary information from different layers. Since the graph is dynamically changing, the node embeddings need to change. To this end, *Snapshot Encoder* uses a GRU cell [97] to update the hierarchical node states over time. Finally, we use the hierarchical node states at each timestamp to calculate the layer-dependent anomalous probabilities of an existing edge and a negative sampled edge and use them as inputs to a margin loss computation. ANOMULY's algorithm appears in detail in Appendix B.

GNN Architecture. A GNN iteratively aggregates messages from the local neighborhood of nodes to learn node embeddings. For a given type of relation r, we use the embedding matrix $\tilde{H}_r^{(\ell)} = \{\tilde{\mathbf{h}}_{r_u}^{(\ell)}\}_{u \in \mathcal{V}}$ to denote the embedding of all vertices in relation type r after applying ℓ-th GNN layer. Given a relation type r, the ℓ-th layer of the GNN, $\tilde{H}_r^{(\ell)} = \textsc{Gnn}_r(\tilde{H}_r^{(\ell-1)})$, is defined as:

$$\begin{aligned} \mathbf{m}^{\ell}_{r_{(v \to u)}} &= W_r^{(\ell)} \textsc{Concat}\left(\tilde{\mathbf{h}}_{r_v}^{(\ell-1)}, \tilde{\mathbf{h}}_{r_u}^{(\ell-1)}, \tau_{(v,u,r)} \right), \\ \tilde{\mathbf{h}}_{r_u}^{(l)} &= \textsc{Agg}^{(\ell)} \left(\left\{ \mathbf{m}^{\ell}_{r_{(v \to u)}} | v \in \mathcal{N}_r(u) \right\} \right) + \tilde{\mathbf{h}}_{r_u}^{(\ell-1)}. \end{aligned} \tag{1}$$

In our experiments, we follow You et al. [27], and use summation as the aggregation function, i.e., $\textsc{Agg}(.) = \textsc{Sum}(.)$. We also use skip-connections [98] after aggregation.

Update Modules. Given the snapshot $\mathcal{G}^{(t)} = (\mathcal{V}^{(t)}, \mathcal{E}^{(t)}, \mathcal{X}^{(t)})$ at time t and a relation type r, we denote the embedding matrix in relation r after the ℓ-th GNN layer at time t by $\tilde{H}_r^{(t)^{(\ell)}} = \{\tilde{\mathbf{h}}_{r_u}^{(t)^{(\ell)}}\}_{u \in \mathcal{V}}$. To take advantage of historical data and update the node embeddings at each timestamp, in the *Embedding update* block (see Figure 1), we use a GRU cell [97]. Given a relation type r, the output of the ℓ-th *Embedding update*, $\hat{H}_r^{(t)^{(\ell)}} = \{\hat{\mathbf{h}}_{r_u}^{(t)^{(\ell)}}\}_{u \in \mathcal{V}}$, is:

$$\hat{H}_r^{(t)^{(\ell)}} = \textsc{Gru}_r \left(\tilde{H}_r^{(t)^{(\ell)}}, H_r^{(t-1)^{(\ell)}} \right), \tag{2}$$

where $\tilde{H}_r^{(t)^{(\ell)}}$ is the output of the ℓ-th GNN layer, and $H_r^{(t-1)^{(\ell)}}$ is the layer-aware embedding matrix at time $t-1$. The layer-aware embedding matrix will be defined later in this section (see Equation 5).

Attention Mechanism. The role of our attention mechanism is to incorporate information from different relation types in a weighted manner. As discussed in section 1, the importance of a layer can differ for different nodes, so we cannot calculate a single weight for each layer. Accordingly, we suggest an attention mechanism that learns the importance of layer r for an arbitrary node $u \in \mathcal{V}$. Let $\zeta_u^{(t)^{(\ell)}}$ be the aggregated hidden feature of node $u \in \mathcal{V}$ after the ℓ-th attention layer at time t, we call it a network-level embedding, and $\alpha_{r_u}^{(\ell)}$ indicates the importance of relation type r for vertex u, then:

$$\zeta_u^{(t)^{(\ell)}} = \sum_{r=1}^{L} \alpha_{r_u}^{(\ell)} \hat{\mathbf{h}}_{r_u}^{(t)^{(\ell)}}, \tag{3}$$

where $\hat{\mathbf{h}}_{r_u}^{(t)^{(\ell)}}$ is the output of ℓ-th *Embedding update* for node u at time t. Following the recent attention-based models [88, 99], we use the softmax function to define the importance weight of relation type r for node u:

$$\alpha_{r_u}^{(\ell)} = \frac{\exp\left(\sigma \left(\mathbf{s}_r^{(t)^{(\ell)}T} \mathbf{W}_r^{\text{att}} \; \hat{\mathbf{h}}_{r_u}^{(t)^{(\ell)}} \right) \right)}{\sum_{k=1}^{L} \exp\left(\sigma \left(\mathbf{s}_k^{(t)^{(\ell)}T} \mathbf{W}_k^{\text{att}} \; \hat{\mathbf{h}}_k^{(t)^{(\ell)}} \right) \right)}, \tag{4}$$

where $\mathbf{s}_r^{(t)^{(\ell)}}$ is a summary of the network in relation type r at time t, i.e., $\mathbf{s}_r^{(t)^{(\ell)}} = \sum_{u \in \mathcal{V}} \hat{\mathbf{h}}_{r_u}^{(t)^{(\ell)}}$, and $\mathbf{W}_r^{\text{att}}$ is a trainable weight matrix. In our experiments, we use $\tanh(.)$ as the activation function $\sigma(.)$.

Layer-aware Embedding. The output of the attention mechanism is a network-level node embedding matrix, which summarizes the properties of nodes over all relation types. Given a relation type r, to obtain the layer-aware node embedding of a vertex $u \in \mathcal{V}$, we aggregate the output of the *Embedding*

Table 1: Network Statistics.

Dataset	Multiplex Networks						Single-layer Networks		
	RM	DKPol	Amazon	Ethereum	Ripple	DBLP	Bitcoin	Amazon-S	DBLP-S
$\|\mathcal{V}\|$	91	490	17.5K	221K	54K	513K	3.7K	8.6K	23K
$\|\mathcal{E}\|$	14K	20K	282K	473K	837K	1M	24.1K	90K	95.2K
$\|\mathcal{L}\|$	10	3	2	6	5	10	1	1	1

update block, i.e. $\hat{\mathbf{h}}_{r_u}^{(t)^{(\ell)}}$, and this network-level node embedding, i.e. $\zeta_u^{(t)^{(\ell)}}$. That is:

$$\mathbf{h}_{r_u}^{(t)^{(\ell)}} = \text{AGG}^{(\ell)}\left(\hat{\mathbf{h}}_{r_u}^{(t)^{(\ell)}}, \zeta_u^{(t)^{(\ell)}}\right). \tag{5}$$

Based on Equation 5, we obtain the layer-aware node embedding matrix, $H_r^{(t)^{(\ell)}} = \{\mathbf{h}_{r_u}^{(t)^{(\ell)}}\}_{u\in\mathcal{V}}$, for any relation type r. Note that we use the layer-aware node embedding matrix at time $t-1$, $H_r^{(t-1)^{(\ell)}}$, in Equation 2 to update node embeddings after the ℓ-layer GNN layer.

Anomalous Score Computation. Now, we get the layer-aware node embedding matrix $H_r^{(t)} = H_r^{(t)^{(L)}}$ at time t, for each relation type r. Here L is the number of GNN layers. Inspired by Zheng et al. [19], for an edge $(u,v)\in\mathcal{E}_r$, we define its anomalous score as follows:

$$\varphi_r^{(t)}(u,v) = \sigma\left(\eta.\left(||\mathbf{a}\odot\mathbf{h}_{r_u}^{(t)} + \mathbf{b}\odot\mathbf{h}_{r_v}^{(t)}||_2^2 - \mu\right)\right), \tag{6}$$

where $\sigma(.)$ is an activation function, $\mathbf{a}$ and $\mathbf{b}$ are trainable vectors, and η and μ are hyperparameters.

Training and Loss Function. In the training phase, we employ a negative sampling approach in multiplex networks to corrupt edges and generate anomalous connections. Inspired by the negative sampling methods proposed by Wang et al. [100] and Zheng et al. [19], given a relation type r, and a normal edge $(u,v)\in\mathcal{E}_r$, we employ a Bernoulli distribution such that we replace u (resp. v) with probability $\frac{deg_r(u)}{deg_r(u)+deg_r(v)}$ (resp. $\frac{deg_r(v)}{deg_r(u)+deg_r(v)}$) in relation type r to generate random negative samples. Since the corrupted edges might be normal, a strict loss function (e.g., cross-entropy) can affect the performance. Accordingly, we employ the margin-based pairwise loss [101] in each relation type r. Given a relation type r, we also employ a $L2$-regularization loss, $\mathscr{L}_r^{reg}$, which is the summation of the $L2$ norm of all trainable parameters, to avoid overfitting. Finally, to aggregate the loss function over all relation types in the multiplex networks, we use the average of loss functions, i.e.:

$$\mathscr{L} = \frac{1}{|\mathcal{L}|}\left(\sum_{r=1}^{\mathcal{L}}\sum_{(u,v)\in\mathcal{E}_r}\sum_{(u',v')\notin\mathcal{E}_r}\max\{0,\gamma+\varphi_r(u,v)-\varphi_r(u',v')\} + \lambda\mathscr{L}_r^{reg}\right). \tag{7}$$

Here, $0\le\gamma\le 1$ is the margin between normal and corrupted edges.

4 Experiments

Experimental Setup and Metrics. The *Snapshot Encoder* employs a GNN with 200 hidden dimensions for node states, layers with skip-connection, sum aggregation, and batch-normalization. We tune hyper-parameters by cross-validation on a rolling basis and search the hyper-parameters over **(i)** the numbers of layers (1 to 5); **(ii)** learning rate (0.001 to 0.01); and **(iii)** the margin γ (in increments of 0.05 in the range 0.3 to 0.7). The values of the hyper-parameters are reported in Table 5 in Appendix D. We implement ANOMULY with the GraphGym library [102] and use an NVIDIA V100 GPU in the experiments. We use the AUC (the area under the ROC curve) as the metric of comparison. The higher the AUC value, the higher the quality of the method.

Datasets. We use nine real-world public datasets [31, 45, 103, 104, 105, 106] whose domains cover social, co-authorship, blockchain, and co-purchasing networks. We summarize their statistics in Table 1 and provide detailed descriptions in Appendix C. Since the ground truth for anomaly detection is difficult to obtain [3], we extend the methodology in existing studies [3, 19, 20] to multiplex networks and propose a new approach to inject anomalous edges into our datasets (more

Table 2: Performance comparison in multiplex networks (AUC).

Methods	RM		DKPol		Amazon		DBLP		Ethereum		Ripple	
Anomaly %	1%	5 %	1%	5 %	1%	5 %	1%	5 %	1%	5 %	1%	5 %
Single-layer Methods												
GOUTLIER	0.7138	0.6982	0.6844	0.6597	0.6973	0.6672	0.7059	0.6901	0.7017	0.6799	0.7036	0.6851
CM-SKETCH	0.7127	0.7012	0.7058	0.6930	0.6881	0.6719	0.7186	0.6915	0.7408	0.7277	0.7360	0.7194
NETWALK	0.7739	0.7641	0.7706	0.7581	0.7228	0.7122	0.7742	0.7523	0.7956	0.7885	0.7904	0.7823
ADDGRAPH	0.8005	0.8093	0.8149	0.8087	0.7796	0.7735	0.8024	0.7995	0.8133	0.8090	0.8205	0.8217
Multiplex Methods												
MNE	0.7994	0.7955	0.8050	0.7913	0.7108	0.7017	0.7532	0.7499	0.7541	0.7495	0.7813	0.7754
ML-GCN	0.7921	0.7886	0.7915	0.7907	0.7344	0.7263	0.7519	0.7439	0.7940	0.7918	0.8115	0.8072
ANOMULY	**0.8783**	**0.8729**	**0.8694**	**0.8610**	**0.8289**	**0.8195**	**0.8825**	**0.8754**	**0.8906**	**0.8852**	**0.8938**	**0.8871**
Improvement	9.71%	7.85%	6.68%	6.46%	6.32%	5.92%	9.98%	9.49%	9.50%	9.41%	8.93%	7.96%

Table 3: Performance comparison in simple networks (AUC).

Methods	Bitcoin		Amazon-S		DBLP-S	
Anomaly %	1%	5 %	1%	5 %	1%	5 %
GOUTLIER	0.7143	0.7091	0.6923	0.6614	0.7108	0.6995
CM-SKETCH	0.7146	0.7015	0.7049	0.6621	0.7084	0.6877
NETWALK	0.8375	0.8367	0.7483	0.7302	0.7779	0.7590
ADDGRAPH	0.8534	0.8416	0.7872	0.7828	0.7911	0.7932
ANOMULY	**0.8707**	**0.8661**	**0.8014**	**0.7943**	**0.8129**	**0.8236**

Table 4: Ablation study (AUC).

Methods	Amazon	DBLP	Ethereum
ANOMULY	**0.8289**	**0.8825**	**0.8906**
w/ MLP	0.8023	0.8606	0.8718
w/o Attention	0.7831	0.8219	0.8275
w/ Summation	0.8007	0.8571	0.8688

details are in Appendix C). Note that the DKPol and RM datasets are static multiplex networks, and we use them to show the effectiveness of ANOMULY in capturing content and structural features.

Baselines. Since there is no prior work on edge anomaly detection in multiplex networks, we first compare ANOMULY with single-layer edge anomaly detection methods: GOutlier [47] builds a generative model for edges in a node cluster. CM-Sketch [26] uses a Count-Min sketch for approximating global and local structural properties. NetWalk [20] uses a random walk to learn a unified embedding for each node and then dynamically clusters the nodes' embeddings. AddGraph [19] is an end-to-end approach that uses an extended GCN in temporal networks. Finally, we compare with two multiplex network embedding baselines, ML-GCN [84] and MNE [90]. We apply K-means clustering on their obtained node embeddings for anomaly detection [20].

Results on Multiplex Networks. We compare ANOMULY with the baseline methods on both dynamic and static multiplex networks with different percentages of anomalous edges (i.e., 1%, 5%). Table 2 reports the AUC for both the baselines and ANOMULY. Our method outperforms *all* baselines in all datasets and improves the best baseline results by 8.18% on average. There are three reasons for ANOMULY's superior performance: **(1)** it outperforms competitors for static datasets, because it can learn structural anomaly patterns in the network, rather than depending on pre-defined patterns/roles. **(2)** ANOMULY outperforms single-layer methods due to its attention mechanism that incorporates complementary information from different relation types. **(3)** ANOMULY outperforms multiplex methods as it is an end-to-end method and is optimized for anomaly detection. It is also a dynamic method and can take advantage of the temporal properties of the network.

Results on Single-layer Dynamic Networks. We also compare ANOMULY with single-layer method baselines on single-layer dynamic datasets. Table 3 summarizes the results. Once again, we see that ANOMULY outperforms all the baselines, even in single-layer graphs, by 2.46% on average. This is mainly due to *Snapshot Encoder*'s architecture, which enables our method to incorporate the outputs of GNN layers after each layer and recurrently update them over time using a GRU cell.

Ablation Study. Next, we conduct experiments to show that the ANOMULY architecture design is effective in boosting performance. We examine the effect of GRU cells by replacing them with a 2-layer MLP. We also investigate the effect of the attention mechanism by **(1)** removing the attention, learning node embeddings in each relation-type separately and **(2)** aggregating the information of different relation types by summation (without weights). Table 4 summarizes the results. We found that both our attention mechanism and the GRU cells are important for ANOMULY, producing significant performance boosts.

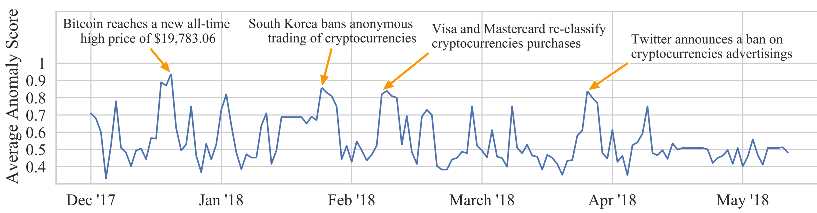


Figure 2: Event detection in Ethereum network.

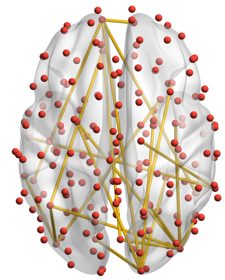


Figure 3: Anomalous edges in BN.

Additional experimental results on the effect of the training ratio can be found in Appendix E.

Effectiveness in Detecting Events. Next, we evaluate how well ANOMULY detects events in the Ethereum transaction network. In each timestamp, we calculate the anomaly score for all the edges in the snapshot. We then compute the average of the top-15 edge anomaly scores and report them in Figure 2. We find that the top-4 local optimums all coincide with major events annotated in the figure.

Case Study of Brain Networks. Behavioral disturbances in ADHD are thought to be caused by the dysfunction of spatially distributed, interconnected neural systems [107]. We used ANOMULY to detect anomalous connections in the brain network (BN) of people with ADHD. Anomalous functional correlations between BN of people with ADHD compared to those without can help us understand which brain regions are involved in ADHD. Our dataset [108] is derived from the functional magnetic resonance imaging (fMRI) of 40 individuals (20 individuals in the condition group, labeled ADHD, and 20 individuals in the control group, labeled TD) with the same methodology used by Lanciano et al. [30]. Here, each layer (relation type) is the BN of an individual person, where nodes are brain regions, and each edge measures the statistical association between the functionality of its endpoints. We present two results: **(1)** 74% of all detected anomalies are edges in the BNs of people in ADHD group. **(2)** 69% of all found anomalies in the ADHD group correspond to edges in the frontal and occipital cortex of the brain. Figure 3 illustrates the anomalous edges in the brain network of an individual in the ADHD group. These findings show an unexpected functional correlation of occipital and frontal lobes regions with other parts, which are consistent with previous studies on ADHD [32, 109]. More results and visualizations are reported in Appendix E.

5 Conclusion, Limitations, and Future Work

We present ANOMULY, an end-to-end unsupervised framework for detecting edge anomalies in dynamic multiplex networks. ANOMULY is based on a new architecture that employs GNN and GRU cells to take advantage of both temporal and structural properties and adds an attention mechanism that effectively incorporates information across different types of connections. Finally, it uses a negative sampling approach in training to overcome the lack of ground truth data. Extensive experiments show the power of ANOMULY to effectively identify temporal and structural anomalies in both single-layer and multiplex networks. Our case studies on brain networks and blockchain transaction networks show the usefulness of ANOMULY in a wide array of applications and domains. The success raises many interesting directions for future studies: **(1)** ANOMULY shows the potential to detect anomalous connections in human brains, which can help predict brain diseases or disorders. One future direction is to design an end-to-end model based on the ANOMULY architecture and directly optimize it to predict the presence of disease. **(2)** ANOMULY assigns the same importance to all snapshots, while earlier information might have less impact and importance than more recent information.

Acknowledgments and Disclosure of Funding

This research was enabled in part by support provided by WestGrid (https://www.westgrid.ca) and The Digital Research Alliance (https://alliancecan.ca/en). We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] Leman Akoglu and Christos Faloutsos. Anomaly, event, and fraud detection in large network datasets. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, page 773–774, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450318693. doi: 10.1145/2433396.2433496. URL `https://doi.org/10.1145/2433396.2433496`.

[2] Charu C. Aggarwal. Outlier analysis. In *EDBT*. Springer Cham, 2019. ISBN 978-3-319-83772-7. doi: https://doi.org/10.1007/978-3-319-47578-3.

[3] Leman Akoglu, Hanghang Tong, and Danai Koutra. Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery*, 29(3):626–688, May 2015. ISSN 1573-756X. doi: 10.1007/s10618-014-0365-y. URL `https://doi.org/10.1007/s10618-014-0365-y`.

[4] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z. Sheng, Hui Xiong, and Leman Akoglu. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2021. doi: 10.1109/TKDE.2021.3118815.

[5] Stephen Ranshous, Shitian Shen, Danai Koutra, Steve Harenberg, Christos Faloutsos, and Nagiza F. Samatova. Anomaly detection in dynamic networks: A survey. *WIREs Comput. Stat.*, 7(3):223–247, may 2015. ISSN 1939-5108. doi: 10.1002/wics.1347. URL `https://doi.org/10.1002/wics.1347`.

[6] Ninghao Liu, Xiao Huang, and Xia Hu. Accelerated local anomaly detection via resolving attributed networks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 2337–2343, 2017. doi: 10.24963/ijcai.2017/325. URL `https://doi.org/10.24963/ijcai.2017/325`.

[7] Zhen Peng, Minnan Luo, Jundong Li, Huan Liu, and Qinghua Zheng. Anomalous: A joint modeling approach for anomaly detection on attributed networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3513–3519. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi: 10.24963/ijcai.2018/488. URL `https://doi.org/10.24963/ijcai.2018/488`.

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

[9] Zhen Peng, Minnan Luo, Jundong Li, Luguo Xue, and Qinghua Zheng. A deep multi-view framework for anomaly detection on attributed networks. *IEEE Transactions on Knowledge and Data Engineering*, 34(6):2539–2552, 2022. doi: 10.1109/TKDE.2020.3015098.

[10] Kaize Ding, Qinghai Zhou, Hanghang Tong, and Huan Liu. Few-shot network anomaly detection via cross-network meta-learning. In *Proceedings of the Web Conference 2021*, WWW '21, page 2448–2456, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383127. doi: 10.1145/3442381.3449922. URL `https://doi.org/10.1145/3442381.3449922`.

[11] Jianyu Wang, Rui Wen, Chunming Wu, Yu Huang, and Jian Xiong. Fdgars: Fraudster detection via graph convolutional networks in online app review system. In *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19, page 310–316, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450366755. doi: 10.1145/3308560.3316586. URL `https://doi.org/10.1145/3308560.3316586`.

[12] Xueyuan Han, Thomas F. J.-M. Pasquier, Adam Bates, James Mickens, and Margo I. Seltzer. Unicorn: Runtime provenance-based detector for advanced persistent threats. In *27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020*. The Internet Society, 2020. URL `https://www.ndss-symposium.org/ndss-paper/unicorn-runtime-provenance-based-detector-for-advanced-persistent-threats/`.

[13] Stefano Cresci. A decade of social bot detection. *Communications of the ACM*, 63(10):72–83, 2020.

[14] Mohiuddin Ahmed, Abdun Naser Mahmood, and Md Rafiqul Islam. A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems*, 55:278–288, 2016.

[15] Siddharth Bhatia, Bryan Hooi, Minji Yoon, Kijung Shin, and Christos Faloutsos. Midas: Microcluster-based detector of anomalies in edge streams. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3242–3249, Apr. 2020. doi: 10.1609/aaai.v34i04.5724. URL `https://ojs.aaai.org/index.php/AAAI/article/view/5724`.

[16] Dhivya Eswaran and Christos Faloutsos. Sedanspot: Detecting anomalies in edge streams. In *2018 IEEE International conference on data mining (ICDM)*, pages 953–958. IEEE, 2018.

[17] Yen-Yu Chang, Pan Li, Rok Sosic, M. H. Afifi, Marco Schweighauser, and Jure Leskovec. F-fade: Frequency factorization for anomaly detection in edge streams. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, page 589–597, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450382977. doi: 10.1145/3437963.3441806. URL `https://doi.org/10.1145/3437963.3441806`.

[18] Dhivya Eswaran, Christos Faloutsos, Sudipto Guha, and Nina Mishra. Spotlight: Detecting anomalies in streaming graphs. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1378–1386, 2018.

[19] Li Zheng, Zhenpeng Li, Jian Li, Zhao Li, and Jun Gao. Addgraph: Anomaly detection in dynamic graph using attention-based temporal gcn. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4419–4425. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/614. URL `https://doi.org/10.24963/ijcai.2019/614`.

[20] Wenchao Yu, Wei Cheng, Charu C. Aggarwal, Kai Zhang, Haifeng Chen, and Wei Wang. Netwalk: A flexible deep embedding approach for anomaly detection in dynamic networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 2672–2681, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3220024. URL `https://doi.org/10.1145/3219819.3220024`.

[21] Jose Cadena, Feng Chen, and Anil Vullikanti. Graph anomaly detection based on steiner connectivity and density. *Proceedings of the IEEE*, 106(5):829–845, 2018. doi: 10.1109/JPROC.2018.2813311.

[22] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 07 2014. ISSN 2051-1310. doi: 10.1093/comnet/cnu016.

[23] Farnoosh Hashemi, Ali Behrouz, and Laks V.S. Lakshmanan. Firmcore decomposition of multilayer networks. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 1589–1600, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450390965. doi: 10.1145/3485447.3512205. URL `https://doi.org/10.1145/3485447.3512205`.

[24] Edoardo Galimberti, Francesco Bonchi, Francesco Gullo, and Tommaso Lanciano. Core decomposition in multilayer networks: Theory, algorithms, and applications. *ACM Trans. Knowl. Discov. Data*, 14(1), 2020. ISSN 1556-4681. doi: 10.1145/3369872.

[25] Teng Wang, Chunsheng Fang, Derek Lin, and S Felix Wu. Localizing temporal anomalies in large evolving graphs. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 927–935. SIAM, 2015.

[26] Stephen Ranshous, Steve Harenberg, Kshitij Sharma, and Nagiza F Samatova. A scalable approach for outlier detection in edge streams using sketch-based approximations. In *Proceedings of the 2016 SIAM international conference on data mining*, pages 189–197. SIAM, 2016.

[27] Jiaxuan You, Tianyu Du, and Jure Leskovec. Roland: Graph learning framework for dynamic graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 2358–2366, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539300. URL `https://doi.org/10.1145/3534678.3539300`.

[28] Bharat B Biswal, Maarten Mennes, Xi-Nian Zuo, Suril Gohel, Clare Kelly, Steve M Smith, Christian F Beckmann, Jonathan S Adelstein, Randy L Buckner, Stan Colcombe, et al. Toward discovery science of human brain function. *Proceedings of the National Academy of Sciences*, 107(10):4734–4739, 2010.

[29] Jonathan D. Power, Alexander L. Cohen, Steven M. Nelson, Gagan S. Wig, Kelly Anne Barnes, Jessica A. Church, Alecia C. Vogel, Timothy O. Laumann, Fran M. Miezin, Bradley L. Schlaggar, and Steven E. Petersen. Functional network organization of the human brain. *Neuron*, 72(4):665–678, Nov 2011. ISSN 1097-4199. doi: 10.1016/j.neuron.2011.09.006. URL `https://pubmed.ncbi.nlm.nih.gov/22099467`. S0896-6273(11)00792-6[PII].

[30] Tommaso Lanciano, Francesco Bonchi, and Aristides Gionis. Explainable classification of brain networks via contrast subgraphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery; Data Mining*, KDD '20, page 3308–3318, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3403383. URL `https://doi.org/10.1145/3394486.3403383`.

[31] Ali Behrouz, Farnoosh Hashemi, and Laks V. S. Lakshmanan. Firmtruss community search in multilayer networks. *arXiv preprint arXiv:2205.00742*, 2022. doi: 10.48550/ARXIV.2205.00742.

[32] Tanima Chatterjee, Réka Albert, Stuti Thapliyal, Nazanin Azarhooshang, and Bhaskar DasGupta. Detecting network anomalies using forman–ricci curvature and a case study for human brain networks. *Scientific Reports*, 11(1):8121, Apr 2021. ISSN 2045-2322. doi: 10.1038/s41598-021-87587-z. URL `https://doi.org/10.1038/s41598-021-87587-z`.

[33] Carlo Abrate and Francesco Bonchi. Counterfactual graphs for explainable classification of brain networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery; Data Mining*, KDD '21, page 2495–2504, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467154. URL `https://doi.org/10.1145/3447548.3467154`.

[34] Manlio De Domenico. Multilayer modeling and analysis of human brain networks. *Giga Science*, 6(5):gix004, 2017.

[35] Damiano Di Francesco Maesa, Andrea Marino, and Laura Ricci. Detecting artificial behaviours in the bitcoin users graph. *Online Social Networks and Media*, 3:63–74, 2017.

[36] Thai Pham and Steven Lee. Anomaly detection in bitcoin network using unsupervised learning methods. *arXiv preprint arXiv:1611.03941*, 2016.

[37] Tao-Hung Chang and Davor Svetinovic. Improving bitcoin ownership identification using transaction patterns analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(1):9–20, 2018.

[38] Somdip Dey. Securing majority-attack in blockchain using machine learning and algorithmic game theory: A proof of work. In *2018 10th computer science and electronic engineering (CEEC)*, pages 7–10. IEEE, 2018.

[39] Blaž Podgorelec, Muhamed Turkanović, and Sašo Karakatič. A machine learning-based method for automated blockchain transaction signing including personalized anomaly detection. *Sensors*, 20(1):147, 2019.

[40] Muneeb Ul Hassan, Mubashir Husain Rehmani, and Jinjun Chen. Anomaly detection in blockchain networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 2022.

[41] Amir Kafshdar Goharshady, Ali Behrouz, and Krishnendu Chatteriee. Secure credit reporting on the blockchain. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1343–1348, 2018. doi: 10.1109/Cybermatics_2018.2018.00231.

[42] Sabyasachi Chakraborty, Satyabrata Aich, Sim Jong Seong, and Hee-Cheol Kim. A blockchain based credit analysis framework for efficient financial systems. In *2019 21st International Conference on Advanced Communication Technology (ICACT)*, pages 56–60. IEEE, 2019.

[43] Ye Guo and Chen Liang. Blockchain application and outlook in the banking industry. *Financial innovation*, 2(1):1–12, 2016.

[44] Danny Nelson. Crypto criminals have already stolen $1.4b in 2020, says ciphertrace, June 2020. URL `https://www.coindesk.com/policy/2020/06/02/crypto-criminals-have-already-stolen-14b-in-2020-says-ciphertrace/`.

[45] Dorcas Ofori-Boateng, I Segovia Dominguez, C Akcora, Murat Kantarcioglu, and Yulia R Gel. Topological anomaly detection in dynamic multilayer blockchain networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 788–804. Springer, 2021.

[46] Haaroon Yousaf, George Kappos, and Sarah Meiklejohn. Tracing transactions across cryptocurrency ledgers. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 837–850, 2019.

[47] Charu C. Aggarwal, Yuchen Zhao, and Philip S. Yu. Outlier detection in graph streams. In *2011 IEEE 27th International Conference on Data Engineering*, pages 399–409, 2011. doi: 10.1109/ICDE.2011.5767885.

[48] Nicholas A Heard, David J Weston, Kiriaki Platanioti, and David J Hand. Bayesian anomaly detection methods for social networks. *The Annals of Applied Statistics*, 4(2):645–662, 2010.

[49] Weiren Yu, Charu C Aggarwal, Shuai Ma, and Haixun Wang. On anomalous hotspot discovery in graph streams. In *2013 IEEE 13th International Conference on Data Mining*, pages 1271–1276. IEEE, 2013.

[50] Leto Peel and Aaron Clauset. Detecting change points in the large-scale structure of evolving networks. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[51] Feng Chen and Daniel B Neill. Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1166–1175, 2014.

[52] Heng Wang, Minh Tang, Youngser Park, and Carey E Priebe. Locality statistics for anomaly detection in time series of graphs. *IEEE Transactions on Signal Processing*, 62(3):703–717, 2013.

[53] Minji Yoon, Bryan Hooi, Kijung Shin, and Christos Faloutsos. Fast and accurate anomaly detection in dynamic graphs with a two-pronged approach. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 647–657, 2019.

[54] Bryan Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, and Christos Faloutsos. Fraudar: Bounding graph fraud in the face of camouflage. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 895–904, 2016.

[55] Hao Yan, Qianzhen Zhang, Deming Mao, Ziyue Lu, Deke Guo, and Sheng Chen. Anomaly detection of network streams via dense subgraph discovery. In *2021 International Conference on Computer Communications and Networks (ICCCN)*, pages 1–9. IEEE, 2021.

[56] Jose Cadena, Feng Chen, and Anil Vullikanti. Graph anomaly detection based on steiner connectivity and density. *Proceedings of the IEEE*, 106(5):829–845, 2018.

[57] Jimeng Sun, Dacheng Tao, and Christos Faloutsos. Beyond streams and graphs: dynamic tensor analysis. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 374–383, 2006.

[58] Jimeng Sun, Yinglian Xie, Hui Zhang, and Christos Faloutsos. Less is more: Compact matrix decomposition for large sparse graphs. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 366–377. SIAM, 2007.

[59] Xian Teng, Yu-Ru Lin, and Xidao Wen. Anomaly detection in dynamic networks using multi-view time-series hypersphere learning. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 827–836, 2017.

[60] Wenchao Yu, Charu C Aggarwal, and Wei Wang. Temporally factorized network modeling for evolutionary network analysis. In *Proceedings of the Tenth ACM International conference on web search and data mining*, pages 455–464, 2017.

[61] Morteza Mardani, Gonzalo Mateos, and Georgios B Giannakis. Dynamic anomalography: Tracking network anomalies via sparsity and low rank. *IEEE Journal of Selected Topics in Signal Processing*, 7(1):50–66, 2012.

[62] Yixin Liu, Shirui Pan, Yu Guang Wang, Fei Xiong, Liang Wang, Qingfeng Chen, and Vincent CS Lee. Anomaly detection in dynamic graphs via transformer. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2021. doi: 10.1109/TKDE.2021.3124061.

[63] Ming Jin, Yixin Liu, Yu Zheng, Lianhua Chi, Yuan-Fang Li, and Shirui Pan. Anemone: Graph anomaly detection with multi-scale contrastive learning. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 3122–3126, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384469. doi: 10.1145/3459637.3482057. URL https://doi.org/10.1145/3459637.3482057.

[64] Yu Zheng, Ming Jin, Yixin Liu, Lianhua Chi, Khoa T Phan, and Yi-Ping Phoebe Chen. Generative and contrastive self-supervised learning for graph anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

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

[66] Jianfei Gao and Bruno Ribeiro. On the equivalence between temporal and static equivariant graph representations. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 7052–7076. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/gao22e.html.

[67] Hao Peng, Hongfei Wang, Bowen Du, Md Zakirul Alam Bhuiyan, Hongyuan Ma, Jianwei Liu, Lihong Wang, Zeyu Yang, Linfeng Du, Senzhang Wang, et al. Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting. *Information Sciences*, 521:277–290, 2020.

[68] Jinyin Chen, Xueke Wang, and Xuanheng Xu. Gc-lstm: Graph convolution embedded lstm for dynamic link prediction. *arXiv preprint arXiv:1812.04206*, 2018.

[69] Jia Li, Zhichao Han, Hong Cheng, Jiao Su, Pengyun Wang, Jianfeng Zhang, and Lujia Pan. Predicting path failure in time-evolving graphs. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1279–1289, 2019.

[70] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5363–5370, 2020.

[71] Aynaz Taheri, Kevin Gimpel, and Tanya Berger-Wolf. Learning to represent the evolution of dynamic graphs with recurrent models. In *Companion proceedings of the 2019 world wide web conference*, pages 301–307, 2019.

[72] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJiHXGWAZ.

[73] Cheng Yang, Chunchen Wang, Yuanfu Lu, Xumeng Gong, Chuan Shi, Wei Wang, and Xu Zhang. Few-shot link prediction in dynamic networks. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 1245–1255, 2022.

[74] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. Structured sequence modeling with graph convolutional recurrent networks. In *International conference on neural information processing*, pages 362–373. Springer, 2018.

[75] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3848–3858, 2019.

[76] Xiaoyang Wang, Yao Ma, Yiqi Wang, Wei Jin, Xin Wang, Jiliang Tang, Caiyan Jia, and Jian Yu. Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of The Web Conference 2020*, pages 1082–1092, 2020.

[77] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *IJCAI*, pages 3634–3640, 2018. URL https://doi.org/10.24963/ijcai.2018/505.

[78] Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. Representation learning for attributed multiplex heterogeneous network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1358–1368, 2019.

[79] Léo Pio-Lopez, Alberto Valdeolivas, Laurent Tichit, Élisabeth Remy, and Anaïs Baudot. Multiverse: a multiplex and multiplex-heterogeneous network embedding approach. *Scientific Reports*, 11(1):1–20, 2021.

[80] Yuchen Yan, Si Zhang, and Hanghang Tong. Bright: A bridging algorithm for network alignment. In *Proceedings of the Web Conference 2021*, pages 3907–3917, 2021.

[81] Shiyu Chang, Wei Han, Jiliang Tang, Guo-Jun Qi, Charu C Aggarwal, and Thomas S Huang. Heterogeneous network embedding via deep architectures. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 119–128, 2015.

[82] Yuanzhen Xie, Zijing Ou, Liang Chen, Yang Liu, Kun Xu, Carl Yang, and Zibin Zheng. Learning and updating node embedding on dynamic heterogeneous information network. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, page 184–192, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450382977. doi: 10.1145/3437963.3441745. URL https://doi.org/10.1145/3437963.3441745.

[83] Xiao Wang, Yuanfu Lu, Chuan Shi, Ruijia Wang, Peng Cui, and Shuai Mou. Dynamic heterogeneous information network embedding with meta-path based proximity. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[84] Ali Behrouz and Farnoosh Hashemi. Cs-mlgcn: Multiplex graph convolutional networks for community search in multiplex networks. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, CIKM '22, page 3828–3832, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392365. doi: 10.1145/3511808.3557572. URL https://doi.org/10.1145/3511808.3557572.

[85] Jiafeng Cheng, Qianqian Wang, Zhiqiang Tao, Deyan Xie, and Quanxue Gao. Multi-view attribute graph convolution networks for clustering. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 2973–2979, 2021.

[86] Changqing Zhang, Huazhu Fu, Qinghua Hu, Xiaochun Cao, Yuan Xie, Dacheng Tao, and Dong Xu. Generalized latent multi-view subspace clustering. *IEEE transactions on pattern analysis and machine intelligence*, 42(1):86–99, 2018.

[87] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rklz9iAcKQ.

[88] Chanyoung Park, Donghyun Kim, Jiawei Han, and Hwanjo Yu. Unsupervised attributed multiplex network embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5371–5378, 2020.

[89] Baoyu Jing, Chanyoung Park, and Hanghang Tong. Hdmi: High-order deep multiplex infomax. In *Proceedings of the Web Conference 2021*, pages 2414–2424, 2021.

[90] Hongming Zhang, Liwei Qiu, Lingling Yi, and Yangqiu Song. Scalable multiplex network embedding. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3082–3088. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi: 10.24963/ijcai.2018/428. URL https://doi.org/10.24963/ijcai.2018/428.

[91] Qifan Wang, Yi Fang, Anirudh Ravula, Ruining He, Bin Shen, Jingang Wang, Xiaojun Quan, and Dongfang Liu. Deep partial multiplex network embedding. In *Companion Proceedings of the Web Conference 2022*, WWW '22, page 1053–1062, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391306. doi: 10.1145/3487553.3524717. URL https://doi.org/10.1145/3487553.3524717.

[92] Ruchi Mittal and MPS Bhatia. Anomaly detection in multiplex networks. *Procedia Computer Science*, 125:609–616, 2018.

[93] P.V. Bindu, P. Santhi Thilagam, and Deepesh Ahuja. Discovering suspicious behavior in multilayer social networks. *Computers in Human Behavior*, 73:568–582, 2017. ISSN 0747-5632. doi: https://doi.org/10.1016/j.chb.2017.04.001. URL https://www.sciencedirect.com/science/article/pii/S0747563217302303.

[94] Monika Bansal and Dolly Sharma. Ranking and discovering anomalous neighborhoods in attributed multiplex networks. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, CoDS COMAD 2020, page 46–54, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450377386. doi: 10.1145/3371158.3371164. URL https://doi.org/10.1145/3371158.3371164.

[95] Asep Maulana and Martin Atzmueller. Centrality-based anomaly detection on multi-layer networks using many-objective optimization. In *2020 7th International Conference on Control, Decision and Information Technologies (CoDIT)*, volume 1, pages 633–638, 2020. doi: 10.1109/CoDIT49905.2020.9263819.

[96] Ling-Hao Chen, He Li, and Wenhao Yang. Anomman: Detect anomaly on multi-view attributed networks, 2022. URL https://arxiv.org/abs/2201.02822.

[97] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[98] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9267–9276, 2019.

[99] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1409.0473`.

[100] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28, 2014.

[101] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

[102] Jiaxuan You, Rex Ying, and Jure Leskovec. Design space for graph neural networks. In *NeurIPS*, 2020.

[103] Jungeun Kim and Jae-Gil Lee. Community detection in multi-layer graphs: A survey. *SIGMOD Rec.*, 44(3):37–48, dec 2015. ISSN 0163-5808. doi: 10.1145/2854006.2854013. URL `https://doi.org/10.1145/2854006.2854013`.

[104] Obaida Hanteer, Luca Rossi, Davide Vega D'Aurelio, and Matteo Magnani. From interaction to participation: The role of the imagined audience in social media community detection and an application to political communication on twitter. In *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '18, page 531–534. IEEE Press, 2018. ISBN 9781538660515.

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

[106] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517, 2016.

[107] Kristi R. Griffiths, Taylor A. Braund, Michael R. Kohn, Simon Clarke, Leanne M. Williams, and Mayuresh S. Korgaonkar. Structural brain network topology underpinning adhd and response to methylphenidate treatment. *Translational Psychiatry*, 11(1):150, Mar 2021. ISSN 2158-3188. doi: 10.1038/s41398-021-01278-x. URL `https://doi.org/10.1038/s41398-021-01278-x`.

[108] Jesse A Brown, Jeffrey D Rudie, Anita Bandrowski, John D Van Horn, and Susan Y Bookheimer. The ucla multimodal connectivity database: a web-based platform for brain connectivity matrix sharing and analysis. *Frontiers in neuroinformatics*, 6:28, 2012.

[109] Jian-zhe Wang, Tian-zi Jiang, Qing-jiu Cao, and Yufeng Wang. Characterizing anatomic differences in boys with attention-deficit/hyperactivity disorder with the use of deformation-based morphometry. *American Journal of Neuroradiology*, 28(3):543–547, 2007.

[110] Biao Jie, Mingxia Liu, Xi Jiang, and Daoqiang Zhang. Sub-network based kernels for brain network classification. In *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 622–629, 2016.

[111] Gang Chen, B Douglas Ward, Chunming Xie, Wenjun Li, Zhilin Wu, Jennifer L Jones, Malgorzata Franczak, Piero Antuono, and Shi-Jiang Li. Classification of alzheimer disease, mild cognitive impairment, and normal cognitive status with large-scale network analysis based on resting-state functional mr imaging. *Radiology*, 259(1):213, 2011.

[112] Chong-Yaw Wee, Pew-Thian Yap, Wenbin Li, Kevin Denny, Jeffrey N Browndyke, Guy G Potter, Kathleen A Welsh-Bohmer, Lihong Wang, and Dinggang Shen. Enriched white matter connectivity networks for accurate identification of mci patients. *Neuroimage*, 54(3): 1812–1822, 2011.

[113] Xuan Kan, Hejie Cui, Ying Guo, and Carl Yang. Effective and interpretable fmri analysis via functional brain network generation. *arXiv preprint arXiv:2107.11247*, 2021.

[114] Hejie Cui, Wei Dai, Yanqiao Zhu, Xiaoxiao Li, Lifang He, and Carl Yang. Brainnnexplainer: An interpretable graph neural network framework for brain network based disease analysis. *arXiv preprint arXiv:2107.05097*, 2021.

[115] Xuan Kan, Hejie Cui, Joshua Lukemire, Ying Guo, and Carl Yang. Fbnetgen: Task-aware gnn-based fmri analysis via functional brain network generation. *arXiv preprint arXiv:2205.12465*, 2022.

[116] Yanqiao Zhu, Hejie Cui, Lifang He, Lichao Sun, and Carl Yang. Joint embedding of structural and functional brain networks with graph neural networks for mental illness diagnosis. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 272–276. IEEE, 2022.

[117] Hejie Cui, Wei Dai, Yanqiao Zhu, Xiaoxiao Li, Lifang He, and Carl Yang. Interpretable graph neural networks for connectome-based brain disorder analysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 375–385. Springer, 2022.

[118] Jingyuan Zhang, Bokai Cao, Sihong Xie, Chun-Ta Lu, Philip S Yu, and Ann B Ragin. Identifying connectivity patterns for brain diseases via multi-side-view guided deep architectures. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 36–44. SIAM, 2016.

[119] Jiaxin Liu, Wei Zhao, Ye Hong, Sheng Gao, Xi Huang, Yingjie Zhou, Alzheimer's Disease Neuroimaging Initiative, et al. Learning features of brain network for anomaly detection. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 900–905. IEEE, 2020.

[120] The dblp team. dblp computer science bibliography. `https://dblp.uni-trier.de`, 2014.

[121] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022, mar 2003. ISSN 1532-4435.

[122] Xuemeng Zhai, Wanlei Zhou, Gaolei Fei, Weiyi Liu, Zhoujun Xu, Chengbo Jiao, Cai Lu, and Guangmin Hu. Null model and community structure in multiplex networks. *Scientific reports*, 8(1):1–13, 2018.

A Additional Related Work

Novelty of the *Snapshot Encoder* Architecture. In the literature, several studies combine GNNs with recurrent models such as GRU cells or Transformers. Here, we elaborate on how the *Snapshot Encoder*'s architecture is quite different from this prior work. As discussed in section 2, we can categorize previous work on combining GRU and GNNs into two groups. The first group replaces RNN's linear layer with a graph convolution layer [72, 74, 75]. The second group uses a GNN as a feature encoder and then deploys a sequence model on top of the GNN to encode temporal properties [67, 76, 77], which ignores the evolution of lower-level node embeddings. Also, these models are limited in their training strategy [27]. ROLAND [27] has addressed these limitations, but it is limited to single-layer graphs. Moreover, natural attempts to use multiplex graph neural networks [78, 79, 80, 84, 85, 86] in the ROLAND framework (e.g., replacing the GNN block with a multiplex GNN or GCN) lead to ignoring historical data in other relation types (layers). For example, assume that we want to use our attention mechanism (see section 3) in the ROLAND framework. Then, we need to use the attention mechanism in the GNN block, which means we incorporate the information about different relation types before embedding the *Embedding update* module. Accordingly, for each timestamp, we do not incorporate historical data on other relation types, which could produce undesirable performance.

Feature Learning and Anomaly Detection in Brain Networks. In recent years, several studies focused on analyzing brain networks to understand and distinguish healthy and diseased human brains [110, 111, 112]. Recently, due to the success of GNNs in analyzing graph-structured data, deep models have been proposed to predict brain diseases by learning the graph structures of brain networks[113, 114, 115, 116, 117]. In addition to predicting disease in brain networks, understanding the cause of the disease is important. To this end, several anomaly detection methods have been proposed to find anomalous connections, regions, or subgraphs in the brain, which can cause a disease [32, 118, 119]. However, all these anomaly detection models apply to single-layer networks only and do not naturally extend to multiplex networks, while a brain network generated from an individual can be noisy and incomplete [30, 31]. To the best of our knowledge, ANOMULY is the first method for detecting anomalous connections in multiplex brain networks.

Anomaly Detection in Blockchain Networks. Anomaly detection in blockchain transaction networks has recently attracted enormous attention [35, 36, 37, 38, 39, 40], due to the emergence of a huge assortment of financial systems' applications [41, 42, 43]. However, most existing work focuses on detecting illicit activity in a single blockchain network, while recent research shows that cryptocurrency criminals increasingly employ cross-cryptocurrency trades to hide their identity [44, 45]. Accordingly, Yousaf et al. [46] has recently shown that analyzing links across several blockchain networks is critical for identifying emerging criminal activity on the blockchain. To this end, Ofori-Boateng et al. [45] developed a new persistence summary and utilized it to detect events in dynamic multiplex blockchain networks. For additional related work on single blockchain transaction networks, we refer the reader to the extensive survey by Hassan et al. [40]. To the best of our knowledge, ANOMULY is the first method for detecting anomalous transactions in multiple blockchain networks.

B ANOMULY Algorithm

ANOMULY's algorithm appears in detail in Algorithm 1.

C Datasets

We use nine real-world public datasets [31, 45, 103, 104, 105, 106] whose domains cover social, co-authorship, blockchain, and co-purchasing networks. We summarize their statistics in Table 1.

Social Networks. RM [103] has 10 networks, each with 91 nodes. Nodes represent phones and an edge exists if two phones detect each other in a mobile network. Each network describes connections between phones in a month. DKPol [104] is collected during the month leading to the 2015 Danish parliamentary election on Twitter. Nodes are Twitter accounts of Danish politicians, and relations are "Retweet", "Reply", and "Topical Interaction" [104].

Algorithm 1 ANOMULY Algorithm

Input: A multiplex dynamic network $\mathcal{G} = \{\mathcal{G}^{(t)}\}_{t=1}^{T}$

Output: Node embeddings for all relation types $\{H_r^{(T)}\}_{r=1}^{\mathcal{L}}$

Initialize $\left\{\left\{H_r^{(0)^{(\ell)}}\right\}_{\ell=1}^{L}\right\}_{r=1}^{\mathcal{L}}$;

while not converged **do**

  for $t = 1, \ldots, T$ **do**

    for $r = 1, \ldots, \mathcal{L}$ **do**

      Let $\mathscr{L}_r^{(t)} = 0$;

      for $\ell = 1, \ldots, L$ **do**

        $\tilde{H}_r^{(t)^{(\ell)}} = \text{GNN}_r\left(H_r^{(\ell-1)}\right)$;

        $\hat{H}_r^{(t)^{(\ell)}} = \text{GRU}_r\left(\tilde{H}_r^{(t)^{(\ell)}}, H_r^{(t-1)^{(\ell)}}\right)$;

        $Z_r^{(t)^{(\ell)}} = \{\zeta_u^{(t)^{(\ell)}}\}_{u\in\mathcal{V}} = \left\{\sum_{r=1}^{L} \alpha_{r_u}^{(\ell)} \hat{\mathbf{h}}_{r_u}^{(t)^{(\ell)}}\right\}_{u\in\mathcal{V}}$;

        $H_r^{(t)^{(\ell)}} = \text{AGG}^{(\ell)}\left(\hat{H}_r^{(t)^{(\ell)}}, Z_r^{(t)^{(\ell)}}\right)$

      for $(u, v) \in \mathcal{E}_r^{(t)}$ **do**

        Sample (u', v') for $\varphi_r^{(t)}(u, v)$;

        $\mathscr{L}_r^{(t)} = \mathscr{L}_r^{(t)} + \max\left\{0, \gamma + \varphi_r^{(t)}(u, v) - \varphi_r^{(t)}(u', v')\right\}$;

      $\mathscr{L}_r^{(t)} = \mathscr{L}_r^{(t)} + \lambda \mathscr{L}_r^{reg}$;

    $\mathscr{L}^{(t)} = \frac{1}{\mathcal{L}}\left(\sum_{r=1}^{\mathcal{L}} \mathscr{L}_r^{(t)}\right)$;

    Minimize $\mathscr{L}^{(t)}$;

return $\{H_r^{(T)}\}_{r=1}^{\mathcal{L}}$

Co-purchasing Network. Amazon [106] is a co-purchasing network, where each node is an item and the type of connections are "Also-view" and "Also-bought". We focused on items with four categories, i.e., Beauty, Automotive, Patio Lawn and Garden, and Baby.

Collaboration Network. DBLP is a co-authorship network until 2014 from [120]. In this dataset, each node is a researcher, an edge shows collaboration, and each type of connection is a topic of research. For each collaboration, we consider the bag of words drawn from the titles of the paper and apply LDA topic modeling [121] to automatically identify 240 topics. We then cluster their non-zero elements into ten known research topics. Each layer in the network represents connections in one of the ten topics.

Blockchain Networks. Ethereum [45] is a blockchain transaction network over 576 days, where layers are different tokens, nodes are addresses of investors, and edges denote the transferred token values. Since the same address may trade multiple tokens, the address appears in networks of all the tokens it has traded. Ripple [45] is derived from the Ripple Credit Network and covers a timeline of Oct-2016 to Mar-2020. Similar to the Ethereum dataset, nodes are investors and edges represent transactions. Here layers (relation types) correspond to the five most issued fiat currencies on the Ripple network: JPY, USD, EUR, CCK, and CNY.

Brain Network. We use the ADHD-Brain dataset [108] derived from the functional magnetic resonance imaging (fMRI) of 40 individuals (20 individuals in the condition group, labeled ADHD, and 20 individuals in the control group, labeled TD) using the same methodology used by Lanciano et al. [30]. Here, each layer (relation type) is the brain network of an individual person, where nodes are brain regions, and each edge measures the statistical association between the functionality of its endpoints.

Single-layer Networks. DBLP-S is a subgraph of the multiplex DBLP network, where we collect a subset of researchers who work on data mining and related areas. Amozon-S, is a subgraph of the multiplex Amazon network, where we focus on the "Also-view" relation type. The Bitcoin dataset contains who-trusts-whom network of people who trade on the Alpha platforms [105].

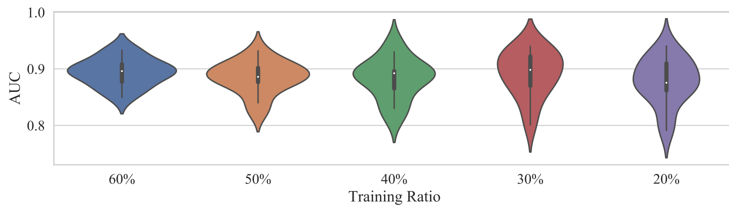


Figure 4: Stability over different training ratio on Ripple.

Note that all of the datasets are anonymized and do not contain any personally identifiable information or offensive content.

C.1 Inject Anomalous Edges in Multiplex Networks

Since the ground truth for anomaly detection is difficult to obtain [3], we follow the methodology used in existing studies[3, 19, 20] and inject anomalous edges into our datasets. However, existing methods only inject anomalous edges in a single-layer network and cannot add complex anomalies in multiplex networks. Accordingly, here we divide injected edges into two groups (50% each), **(1)** layer-independent anomalies, and **(2)** layer-dependent anomalies. For the first group, we use existing methods [3]. In multiplex networks, the rich information about node connections leads to repetitions, meaning edges between the same pair of nodes repeatedly appear in multiple layers. Repeated connections are more likely to be a strong tie and it may even suggest that its nodes belong to the same community [31, 122]. Accordingly, in the second type of injected anomalies, we inject random connections that do not appear in any relation type. That is, we first choose a random edge (u, v) such that $(u, v) \notin \mathcal{E}_k$ for all $k \in \mathcal{L}$, and then we inject this edge to a random relation type $r \in \mathcal{L}$. Since this connection does not appear in any relation type, it is more likely to be an anomaly. This type of anomaly helps us to understand whether ANOMULY can take advantage of complementary information of different relation types.

D Experimental Configuration

In the architecture of *Snapshot Encoder* we use 200 hidden dimensions for node states, GNN layers with skip-connection, sum aggregation, and batch-normalization. We tune hyper-parameters by cross-validation on a rolling basis, and search the hyper-parameters over **(i)** numbers of layers (1 layer to 5 layers); **(ii)** learning rate (0.001 to 0.01); and **(iii)** the margin γ (0.3 to 0.7). The values of other hyper-parameters are reported in Table 5.

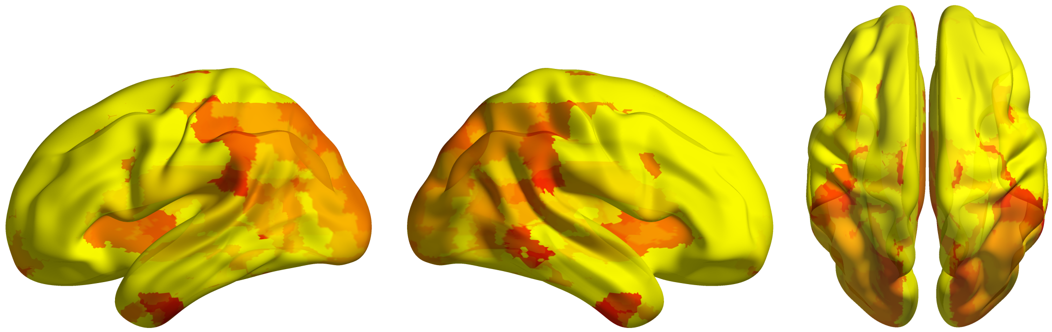
Table 5: The value of hyper-parameters of ANOMULY

Dataset	Multiplex Networks						Single-layer Networks		
	DKPol	RM	Amazon	DBLP	Ethereum	Ripple	Bitcoin	Amazon-S	DBLP-S
η	1	1	1	3	1	1	1	1	3
μ	0.3	0.25	0.5	0.5	0.3	0.3	0.3	0.5	0.5
λ	5×10^{-7}	5×10^{-7}	5×10^{-7}	5×10^{-7}	5×10^{-7}	5×10^{-7}	5×10^{-7}	5×10^{-7}	5×10^{-7}

E Additional Experimental Results

E.1 Parameter Sensitivity

We evaluate the effect of the training ratio of the dataset: we change the training ratio from 60% to 20% and report the results on the Ripple dataset in Figure 4. Decreasing the training ratio tends to increase both the average and maximum AUC, except for the 20% case, and decreases the minimum AUC for all training ratios. These results show that performance stays relatively stable, which demonstrates that our framework is robust in the presence of a small amount of training data.



(a) Left View (b) Right View (c) Up View

Figure 5: Distribution of anomalous edges in ADHD group.

E.2 Additional Results on Brain Networks

In this experiment, we investigate how anomalous edges found by ANOMULY are distributed in the brain. Figure 5 reports the average distribution of anomalous edges in the brain networks of people living with ADHD. Most anomalous edges found by ANOMULY have a vertex in the Occipital lobes. Moreover, the Temporal lobes are the brain regions with the most anomalous connections with the Occipital lobes. These findings can help to reveal new insights into understanding ADHD and the regions of the brain that are connected to its symptoms. These findings also show the potential of ANOMULY to extract features that can help predict brain diseases or disorders.

F Reproducibility

The code, datasets, and supplements are available at `https://github.com/ubc-systopia/ANOMULY`.