Temporal Anomalous Subgraph Discovery in Large-Scale Dynamic Financial Networks

Anonymous Author(s)

ABSTRACT

In this paper, we investigate the discovery of temporal anomalous subgraphs in large-scale financial networks, aiming to identify abnormal transaction behaviors among users over time. This task is crucial for the real-time detection of transaction anomalies in financial networks, such as money laundering and trading fraud. However, it poses significant challenges due to the diverse distribution of transactions, the dynamic nature of temporal networks, and the absence of theoretical foundation. To tackle these challenges, we introduce a novel Temporal Anomalous Subgraph Discovery (**TempASD**) algorithm with theoretical analysis. First, we propose a temporal candidate detection module that quickly pinpoints abnormal candidates by detecting anomalies in both the temporal structure and transaction distribution. Then, we introduce a carefully crafted reinforcement-learning-based refiner to optimize these candidates toward the most abnormal directions. We conducted extensive evaluations against thirteen advanced competitors. TempASD achieves an average improvement of 7× in abnormal degree compared to the state-of-the-art and is efficient in large-scale dynamic financial networks. The implementation details and code are available at https://github.com/FinancialPapercode/TemASD.

KEYWORDS

Temporal anomalous subgraph discovery, Dynamic financial networks, Reinforcement learning

1 INTRODUCTION

The extensive financial data generated from daily transactions, e.g., blockchain, in recent years has facilitated various financial data mining applications [12, 13], e.g., financial time-series forecasting [10, 32, 56], credit risk assessment [5, 20], and anomaly detection [6, 11, 14]. In particular, anomaly detection has been extensively studied recently since it can be used for identifying illegal transactions, such as money laundering and trading fraud [1, 6, 14, 21, 24, 34, 39, 42], thereby safeguarding the assets of users or corporations in the real world [14, 21, 42]. Unlike single-node anomaly discovery, detecting anomalous subgraphs can uncover abnormal structures among users, offering an intuitive way to trace irregular transactions [6, 14, 21].

Temporal anomalous subgraph discovery captures anomalies over time, enabling the timely identification of illegal transactions among users. This dynamic monitoring of suspicious activities is essential for maintaining the security of financial networks. Since temporal anomaly detection is crucial, an increasing number of researchers are focusing on temporal fraud detection in networks [3, 9, 27, 30, 40, 41, 52, 53]. They propose detecting anomalies based on structural and temporal information, which are distinguished from normal patterns in dynamic graphs.

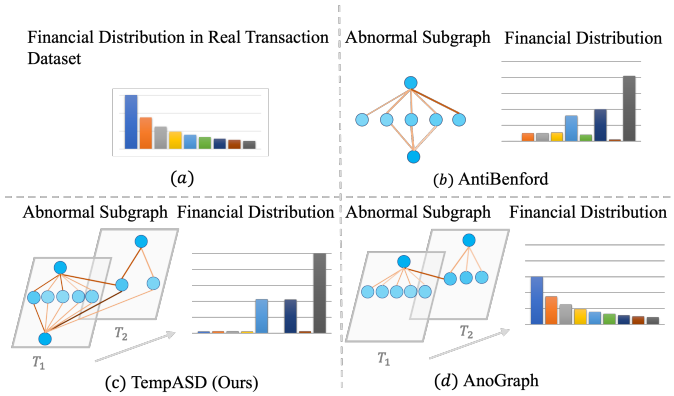

Figure 1: (a) Financial distribution in our transaction dataset, which follows Benford's law; Abnormal subgraph detected by (b) AntiBenford ; (c) TempASD (Ours); (d) AnoGraph. The "financial distribution" refers to the frequency distribution of the first digits of transaction amounts within the transaction subgraph. The height of each bar represents the proportion of transactions starting with a specific digit, ranging from 1 to 9, with each digit represented by a unique color.

However, since these methods are designed for general graphs, they fail to account for specific transaction details among users in financial networks. Specifically, even if some nodes frequently transfer money to other nodes at certain times, leading to structural and temporal anomalies, these transactions might still be normal. For example, university accounts may often deliver stipends or allowances to different student or teacher accounts. If transaction details are overlooked, these transactions could be incorrectly identified as anomalous subgraphs, despite being entirely legal in practice.

To fix these issues, the latest work [6] proposes to detect AntiBenford subgraphs for anomaly detection in static financial networks with transaction information involved. AntiBenford subgraphs are dense subgraphs whose transaction distribution deviates from Benford's law significantly. Benford's law [2, 31] states that the proportion of transaction numbers starting with the digit d follows a monotonically decreasing function of d, i.e., $\log_{10}(1 + 1/d)$. This pattern is observed in natural datasets, e.g., tax records, stock quotations, et al [6]. Figure 1 (a) visualizes the first digit distribution in our evaluation dataset named ETH-Jan-18 (see Section 5.1.1), confirming the presence of Benford's law in financial transactions.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

However, the latest work [6] can only identify static abnormal subgraphs (Figure 1 (b)) within a financial network since it is designed on non-temporal graphs, meaning it cannot detect changes and trends in user transaction behaviors over time. In this work, we theoretically define the temporal AntiBenford subgraphs, inspired by the static AntiBenford subgraphs [6]. This allows for the discovery of temporal abnormal subgraphs that deviate from Benford's law in dynamic financial networks, supported by a solid theoretical foundation. Figure 1 (c) displays the temporal abnormal subgraph discovered by TempASD in our evaluation dataset, with edge color indicating transaction frequency. This subgraph is dense over time, and its financial distribution, i.e., the distribution of the first digits in transaction amounts, significantly deviates from Benford's law. In contrast, the advanced temporal abnormal subgraph discovery method AnoGraph [3] detects subgraphs with structural and temporal anomalies (Figure 1 (d)), but their transaction distribution does not deviate from Benford's law.

Our contributions. Our primary objective is to effectively discover temporal anomalous subgraphs in dynamic financial networks. To achieve this, we first establish a theoretical foundation for identifying temporal abnormal subgraphs whose transaction distributions significantly deviate from Benford's law, i.e., temporal AntiBenford subgraphs. Building on this foundation, we introduce a pruning strategy to quickly pinpoint abnormal candidates based on both structure and transaction anomalies in the encoded vector space. To facilitate this, we propose a temporal subgraph encoder with a subgraph sampling-based algorithm. Subsequently, we propose a novel reinforcement learning-based algorithm to refine the abnormal candidates in the direction of maximal abnormality.

Secondly, we focus on efficiently discovering anomalies in large-scale dynamic financial networks for instant tracking. Specifically, we propose a top-p selection strategy to prune normal temporal subgraphs at an early stage, thereby saving computation in the temporal candidate detection module. Additionally, since the computations for temporal dense subgraphs and transaction information are independent for each node, we utilize multiprocessors for parallelization. Furthermore, we leverage the trained parameters in the reinforcement learning-based algorithm for fast anomaly refinement, as these parameters have already been learned during the offline training phase. TempASD demonstrates near-linear time complexity with respect to graph size.

To summarize, our contributions are:

- **Theoretical Analysis.** We are the first to conduct a theoretical analysis of temporal AntiBenford subgraphs (Section 3.2), which supports the discovery of temporal anomalous subgraphs in dynamic financial networks.
- **Effective Algorithms.** TempASD introduces a temporal candidate detection module together with a temporal anomaly refinement module for the effective discovery of temporal anomalies.
- **Efficient Discovery.** TempASD proposes a pruning strategy with parallelization for fast discovery and achieves near-linear time complexity with respect to graph size.
- **Comprehensive Evaluations.** We conducted extensive experiments on four real transaction datasets, comparing with thirteen methods. TempASD shows an average improvement of 7× in the abnormal degree compared to the state-of-the-art.

2 RELATED WORK

Static anomaly discovery. Non-temporal anomaly detection methods are designed for static graphs, where the structure and relationships between nodes remain unchanged over time. CPM [33] proposes to identify connections between cliques that share nodes for community structure discovery. BigClam [48] detects communities using a nonnegative matrix factorization method but fails to capture varying edge distributions among vertices. Holoscope [21] detects contrast patterns between vertices by analyzing both topology and spikes in graphs. FlowScope [14] efficiently detects the complete flow of money with multipartite graphs, and can scale to large networks. AntiBenford [6] identifies the densest subgraphs in the financial network and filters out those that are statistically normal with respect to the transaction distribution. CLARE [44] locates anomalous communities in the networks and uses a reinforcement-learning framework to refine these anomalies. AS-GAE [54] extracts anomalies with an unsupervised framework. GCAD [59] identifies anomalous subgraphs as those that are rare and significantly different from the majority. BioGNN [8] leverages a bi-level optimization framework within a graph neural network to enhance anomaly detection in graphs. SIGNET [23] identifies motifs that are most discriminative in graph samples to explain graph anomalies. However, these methods are designed for static graphs and fail to account for the dynamic changes in transaction distributions among nodes and edges over time.

Temporal anomaly discovery. There are several advanced methods for discovering temporal anomalous subgraphs. Netspot [30] introduces an efficient technique for detecting anomalous regions in temporal graphs. StreamSpot [27] utilizes clustering for efficient temporal anomaly detection in heterogeneous graphs, which is memory-efficient. MTHL [40] employs a multi-view algorithm to learn a hypersphere around normal patterns, then uses the learned hypersphere to distinguish normal and abnormal ones. A stochastic process-based method [53] is designed to detect changes in graph streams. Netwalk [52] presents a random walk-based algorithm for identifying anomalies in dynamic graphs. DeepSphere [41] introduces a deep autoencoder-based algorithm for dynamic anomaly detection. RustGraph [9] jointly learns structural and temporal dependency in dynamic graphs for anomaly detection. AnoGraph [3] utilizes a sketch-based method for anomaly detection in temporal graphs. However, these methods are designed for general graphs and do not account for the specific anomalies associated with the amounts transferred between nodes in financial transactions. Besides, there are methods designed for temporal anomaly detection at the node level [19, 46] and the edge level [4, 57], which are beyond the scope of this paper.

Temporal densest subgraph discovery. A divide-and-conquer method [50] utilizes pruning strategies to efficiently mine diversified temporal subgraph patterns. FIDES [25] and FIDES$^+$ [26] connect the temporal densest subgraph discovery with a variant of the Prize Collecting Steiner Tree problem and offer efficient solutions for the discovery. FAST-GA [37] employs a greedy method for temporal densest subgraph discovery. TopkDBSOL [7] shows that density bursting subgraphs with long durations can be decomposed into indecomposable density bursting subgraphs with equal or

larger burstiness, then presents an efficient way to detect indecomposable density bursting subgraphs in an online mode. EMU [22] introduces a stochastic method that extends the conventional EM algorithm for detecting the densest temporal subgraphs in dynamic graphs. A hierarchical core maintenance algorithm [18] is designed for large dynamic graphs. TopLC [16] presents a DFS-based search algorithm for discovering diversified lasting cohesive subgraphs in temporal networks. MBC [35] efficiently mines bursting cores in temporal graphs. OTCD [49] introduces a scalable temporal k-core query algorithm using a novel structure named temporal edge list. However, these methods primarily focus on detecting the dense subgraphs based on the structural and temporal information and overlook the amounts of transactions in their model designs.

3 PRELIMINARY

In this section, we first give a problem definition. Then, we give a formal theoretical analysis for temporal anomalous subgraphs.

3.1 Problem Definition

Definition 1 (Temporal Financial Network). *A financial network is represented as a graph $G(V, \mathcal{E})$, where V denotes the set of nodes and $(u, v, t, z) \in \mathcal{E}$ represents a temporal edge indicating a transaction from node u to v at time t with an amount z transferred.*

Definition 2 (Benford's law[2, 6, 31]). *Benford's law is the distribution of the first digit d of numbers appearing in a wide variety of numerical data., i.e.,* $\log_{10}(1 + 1/d)$.

Problem Definition: Temporal Anomalous Subgraph Discovery. Given a temporal financial network $G(V, \mathcal{E})$, the objective is to identify the top-a temporal anomalous subgraphs, where a is a hyperparameter defined by users (refer to Tables 2 and 3 in the experiment section). Each subgraph is defined based on equal time intervals using a time window, and involves a subset of nodes $S \subseteq V$ such that (i) it induces a high average number of temporal edges, which is calculated by dividing the total number of temporal edges within the subgraph during a time window by the number of nodes in the subgraph, and (ii) the empirical histogram of the first digit of these edges' weights, i.e., transaction amounts, significantly deviates from Benford's law statistically.

Note that the key difference between detecting anomalous subgraphs in static graphs (such as in the latest work [6]) and in temporal graphs (such as our TempASD) is that the latter captures subgraphs with abnormal transaction behaviors as they change over time. In contrast, the former overlooks temporal information and only detects anomalies in static graphs, which may be misleading, as those anomalies might not persist when considering the temporal dimension.

3.2 Temporal AntiBenford Subgraph Theory

The existing theory on abnormal subgraphs in financial networks mainly focuses on static subgraphs, such as AntiBenford subgraphs [6]. In this work, we provide a formal theoretical analysis of temporal AntiBenford subgraphs, facilitating the identification of temporal anomalous subgraphs in dynamic financial networks. Our theorems build upon the work in [6], which operates in a static graph setting. We extend these static results to the temporal domain through a theoretical analysis.

Given a temporal financial network $G(V, \mathcal{E})$, we consider the set of its edges as i.i.d. samples from Benford's distribution as [6], and we denote this hypothesis as H_0.

Theorem 3.1. *Let $h(S)$ denote the number of induced edges in the temporal subgraph formed by $S \subseteq V$, let $0 < \epsilon < 1$ be an accuracy parameter, and let $\delta \stackrel{def}{=} \min_d p_d = p_9 = 0.048$, which is the lowest digit probability in Benford's law. Suppose the null hypothesis H_0 holds, with high probability, for all temporal subgraphs $S \subseteq V$ with temporal average degree $\frac{2h(S)}{|S||T_S|}$ at least $C_\epsilon \log n$ (where $C_\epsilon = \frac{36}{\delta\epsilon^2}$ is a constant, $n = |V|$, and $T_S = \{t|(u, v, t, z) \in \mathcal{E}, u, v \in S\}$), the number of temporal edges ($\frac{X_{S,d}}{|T_S|}$) with the first digit d in temporal graph $G[S]$ is strongly concentrated around its expected value $\mathbb{E}\left[\frac{X_{S,d}}{|T_S|}\right]$, which equals $\frac{p_d h(S)}{|T_S|}$, for all $d = 1, \cdots, 9$.*

The intuition behind Theorem 3.1 is that when a subgraph has a sufficiently high temporal average degree, the observed distribution of first digits will strongly concentrate around its expected value. This ensures that deviations from Benford's Law can be flagged as potential anomalies. We use this theorem to derive Corollary 3.2, and the detailed proof can be found in Section B of the appendix.

Corollary 3.2. *Let $\psi(S) \stackrel{def}{=} \frac{1}{|S|}\sum_{d=1}^{9} \frac{\left(\frac{X_{S,d}}{|T_S|} - \frac{p_d h(S)}{|T_S|}\right)^2}{\frac{p_d h(S)}{|T_S|}}$, which is the average deviation of the distribution of the first digits from Benford's law. Suppose the null hypothesis H_0 holds, it expects that for temporal subgraphs $S \subseteq V$, $\max_{S \subseteq V} \psi(S) \leq \rho_T^\star$ holds, where $\rho_T^\star$ represents the temporal density of the temporal densest subgraph in $G[S]$.*

Theorem 3.3. *A temporal AntiBenford subgraph is a subset of nodes $S \subseteq V$ where $\psi(S) \gg \frac{h(S)}{|S||T_S|}$.*

The proofs for all theorems are provided in the appendix.

4 THE TEMPASD MODEL

4.1 Framework

The overall framework of the TempASD model is depicted in Figure 2. The process begins by encoding temporal subgraphs using temporal graph neural networks (Sec. 4.2). Next, it narrows down the pool of temporal anomalous candidates using a pruning strategy for fast discovery (Sec. 4.3). Finally, these candidates are refined with a novel reinforcement learning-based algorithm (Sec. 4.4).

4.2 Temporal Subgraph Encoder

Encoding temporal subgraphs is non-trivial due to their diverse structures and varying financial distributions over time. We propose to encode temporal subgraphs by learning their hidden features in terms of both *temporal graph structure* and *financial distribution*. The *temporal graph structure* reflects connectivity and transaction frequency between nodes, while the *financial distribution* captures the empirical histogram of transaction amounts' first digits.

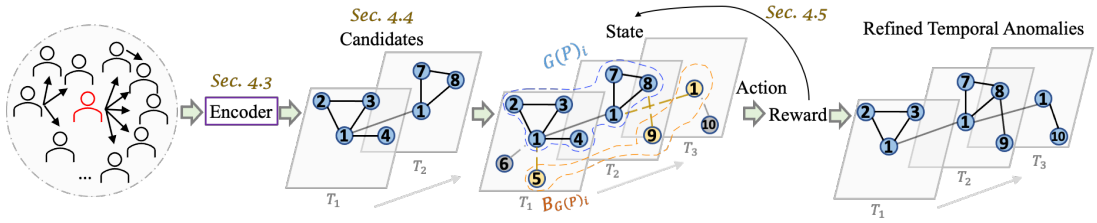

Figure 2: The overall framework of our proposed TempASD model, where $G[S_{P_i}]$ represents the candidates obtained through the temporal candidate detection module in Sec. 4.3, and $B_{S_{P_i}}$ is the boundary of $G[S_{P_i}]$.

Temporal Graph Neural Networks for Financial Subgraphs. We propose to encode the financial transaction information between nodes using temporal graph neural networks. It allows us to effectively capture the dynamic nature of financial interactions over time, enhancing the detection and analysis of temporal anomalies in financial networks. TempASD accommodates any type of temporal graph neural networks (temporal GNN), e.g., [15, 36, 38, 55], without restrictions. For simplicity, we select GConvLSTM [38] as the base encoder in this paper, where the weights in the temporal GNN are transaction frequencies between nodes. GConvLSTM is selected for its ability to handle sequential data with relational dependencies and capture long-term temporal patterns, which are essential in financial networks.

We propose to train the graph structures and temporal interactions among temporal subgraphs using a carefully crafted temporal graph learning algorithm. It improves the model's ability to differentiate temporal anomalies in financial networks by leveraging both structural and temporal information. We divide the temporal graph into snapshots based on equal time intervals using a time window. These snapshots allow us to analyze the network's evolution over time by examining the temporal edges and their weights within each snapshot. The impact of the duration of time intervals will be demonstrated in Section C.3 of appendix. We aim to locate embeddings of subgraphs with similar structures and financial distributions close to each other, while separating the embeddings of subgraphs with differing structures and financial distributions. To construct positive subgraph pairs, for each temporal anomalous subgraph $G[S^i_{a_t}]$ in the training set $\mathcal{G}[S_{a_t}]$, we sample two nested temporal subgraphs $G[S^{b^+}] \subset G[S^a]$ from $G[S^i_{a_t}]$, where $G[S^{b^+}]$ is constructed by randomly removing some transactions from each snapshot of the subgraph $G[S^a]$. In contrast, negative subgraph pairs are formed from another different anomalous subgraph $G[S^j_{a_t}]$ in the set $\mathcal{G}[S_{a_t}]$, and $G[S^a] \nsubseteq G[S^{b^-}]$, $G[S^{b^-}] \subseteq G[S^j_{a_t}]$.

The similarity $\text{sim}(G[S^a], G[S^{b^+}])$ between positive subgraph pairs $G[S^a]$ and $G[S^{b^+}]$ is calculated as:

$$\text{sim}(G[S^a], G[S^{b^+}]) = \sum_{i=1}^{|S^a_{T_0}|} \sum_{q=1}^{T_V} \sum_{k\neq q}^{T_V} \frac{e(G[S^a]_i^{T_q})^\intercal e(G[S^{b^+}]_{\mathcal{N}_i}^{T_k})}{||e(G[S^a]_i^{T_q})||\ ||e(G[S^{b^+}]_{\mathcal{N}_i}^{T_k})||} - \text{KL}\left(w(G[S^a]), w(G[S^{b^+}])\right), \tag{1}$$

where $e(G[S^a]_i^{T_q})$ represents the encoded vector of the i-th node of the subgraph $G[S^a]$ at the snapshot T_q. Similarly, $e(G[S^{b^+}]_{\mathcal{N}_i}^{T_k})$ denotes the encoded vector of the neighborhood of node i of the subgraph $G[S^{b^+}]$ at the snapshot T_k, where $e(G[S^{b^+}]_{\mathcal{N}_i}^{T_k})$ is calculated by the readout$(\cdot)$ function over encoded vectors of all neighbors of node i of the subgraph $G[S^{b^+}]$ at the snapshot T_k. $|S^a_{T_0}|$ is the number of nodes of the subgraph $G[S^a]$ at the snapshot T_0. And T_V denotes the number of sampled snapshots for each temporal subgraph. Additionally, KL $(\cdot)$ denotes the KL-divergence between the financial distributions of two subgraphs $G[S^a]$ and $G[S^{b^+}]$. Subgraphs with similar financial distributions result in lower KL scores.

Similarly, we calculate the similarity between negative pairs as:

$$\text{sim}(G[S^a], G[S^{b^-}]) = \sum_{i=1}^{|S^a_{T_0}|} \sum_{q=1}^{T_V} \sum_{k\neq q}^{T_V} \frac{e(G[S^a]_i^{T_q})^\intercal e(G[S^{b^-}]_{\mathcal{N}_j}^{T_k})}{||e(G[S^a]_i^{T_q})||\ ||e(G[S^{b^-}]_{\mathcal{N}_j}^{T_k})||} - \text{KL}\left(w(G[S^a]), w(G[S^{b^-}])\right), \tag{2}$$

where $e(G[S^{b^-}]_{\mathcal{N}_j}^{T_k})$ denotes the encoded vector of the neighborhood of randomly sampled node j of the subgraph $G[S^{b^-}]$ at the snapshot T_k. Since $G[S^a]$ and $G[S^{b^-}]$ are totally different temporal subgraphs with no overlaps, we cannot use the same node i for the negative pair.

The loss function is optimized by maximizing the similarities between positive pairs and minimizing the similarities between negative pairs, as shown in Equation 3:

$$\mathcal{L} = -\log \frac{\exp(\text{sim}(G[S^a], G[S^{b^+}])/\tau)}{\sum_n \exp(\text{sim}(G[S^a], G[S^{b^-}])/\tau)}, \tag{3}$$

where τ is the temperature parameter.

Based on Theorem A.6 from [45], we derive Theorem 4.1, which demonstrates that our encoder is more effective after sufficient training.

Theorem 4.1. *Let f_1 represent the temporal subgraph encoder we proposed that incorporates financial distribution, and let f_2 be the temporal subgraph encoder without considering financial distribution as [47]. After adequately training f_1 and f_2, we have $I(f_1(\mathcal{G}[S_1]); y) > I(f_2(\mathcal{G}[S_1]); y)$, where y denotes the graph label and $I(\cdot)$ represents mutual information.*

Theorem 4.2. *The time complexity of temporal subgraph encoder is $O(deg_{\text{avg}} \cdot |V| + k_1(|U| + |Z|))$, where deg_{avg} denotes the average degree of nodes, $|U|$ and $|Z|$ represent the number of positive and negative pairs, and k_1 is constant.*

4.3 Temporal Candidate Detection

Firstly, we form the temporal dense subgraph for each vertex in the observation window W_r by greedily removing vertices with the minimum temporal degree as inspired by [17]. The observation window W_r is a sliding window that updates over time. To facilitate the discovery process, we filter the top-p temporal abnormal subgraphs from the obtained dense subgraphs if their transaction distributions significantly deviate from Benford's law, as measured

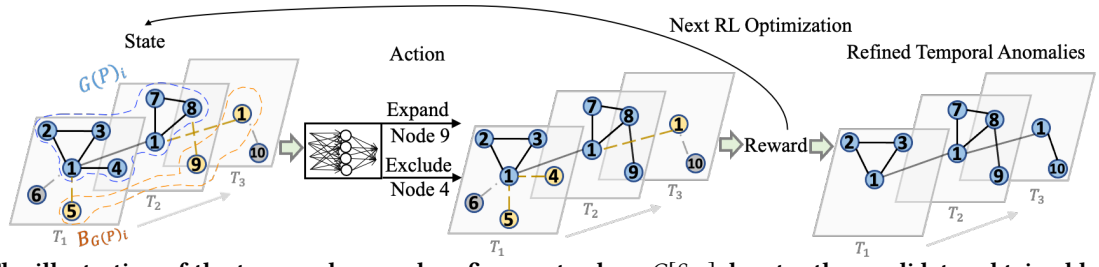

Figure 3: The illustration of the temporal anomaly refinement, where $G[S_{P_i}]$ denotes the candidates obtained by the temporal candidate detection in Sec. 4.3, and $B_{S_{P_i}}$ denotes the boundary of the subgraph $G[S_{P_i}]$. This module further refines the candidates from Sec. 4.3 towards the most anomalous direction.

by the $\mathcal{X}^2(\widehat{S})$-statistic, where $\widehat{S} \subseteq V$. This statistic measures how much the distribution of the subgraph $G[\widehat{S}]$ deviates from Benford's law as follows:

$$\mathcal{X}^2(\widehat{S}) = \sum_{d=1}^{9} \frac{\left(\frac{X_{\widehat{S},d}}{|T_{\widehat{S}}|} - \mathbb{E}\left(\frac{X_{\widehat{S},d}}{|T_{\widehat{S}}|}\right)\right)^2}{\mathbb{E}\left(\frac{X_{\widehat{S},d}}{|T_{\widehat{S}}|}\right)}, \tag{4}$$

where $\frac{X_{\widehat{S},d}}{|T_{\widehat{S}}|}$ represents the number of temporal edges in $G[\widehat{S}]$ with transaction amounts that have a first digit of $d \in \{1, \cdots, 9\}$, and $\mathbb{E}\left(\frac{X_{\widehat{S},d}}{|T_{\widehat{S}}|}\right)$ denotes the expected number of such edges based on Benford's distribution. Since the computations of temporal dense subgraphs and their $\mathcal{X}^2(\widehat{S})$-statistics are independent for different vertices, we perform these calculations in parallel across multiple processors. This step significantly improves efficiency when handling large-scale temporal networks.

Next, the temporal GNN in Sec. 4.2 is employed to encode the temporal subgraph $G[\widehat{S}] \in \mathcal{G}[\widehat{S}]$ into a vector space. The subgraphs in $\mathcal{G}[\widehat{S}]$ whose embeddings closely match the encoded vectors of the training anomalous subgraphs $\mathcal{G}[S_{a_t}]$ are then detected as candidate temporal abnormal subgraphs $\mathcal{G}[S_P]$.

Theorem 4.3. *The time complexity of temporal candidate detection is $O(|V_{W_r}| \cdot |\mathcal{E}_{W_r}| + d_p \cdot p)$, where $|V_{W_r}|$ is the number of vertices during the observation window W_r, $|\mathcal{E}_{W_r}|$ represents the number of edges during W_r, d_p denotes the dimension of the feature vector, and p is the number of candidates.*

4.4 Temporal Anomaly Refinement

State. The state at the i-th step contains the abnormal candidates $G[S_{P_i}]$ obtained from Section 4.3 and its boundary $B_{S_{P_i}}$. We propose selecting nodes with the top transaction frequencies as the boundary $B_{S_{P_i}}$. Specifically, in Figure 3, for the candidate subgraph $G[S_{P_i}]$ from the temporal candidate detection module in Section 4.3, we obtain its connected nodes: node 1 at the snapshot T_3, 5 and 6 at the snapshot T_1, 9 at the snapshot T_2. Among them, since node 1 at T_3, 5 at T_1, and 9 at T_2 have the top-3 transaction frequencies compared with node 6 at T_1, they are selected as the boundary $B_{S_{P_i}}$. Overall, the boundary $B_{S_{P_i}}$ should include the nodes connecting and have the most aggregate transaction frequencies with nodes in $G[S_{P_i}]$, as defined below:

$$B_{S_{P_i}} = \arg\max_{I \subseteq V \setminus G[S_{P_i}]_{T_i} : |I| \le m} \sum_{T_i \in |T|} \sum_{s \in I} \sum_{c \in G[S_{P_i}]_{T_i}} r_{s,c,T_i}, \tag{5}$$

where r_{s,c,T_i} represents the transaction frequency between nodes s and c at the snapshot T_i, and $|I| \le m$ denotes that the number of nodes in the boundary cannot exceed m.

The node representation of each node is generated by the trained temporal GNN in Section 4.2.

Action. We define two actions, exclusion and expansion. The action space for exclusion involves the nodes in the subgraphs $G[S_{P_i}]$. And the action space for expansion contains nodes in the boundary $B_{S_{P_i}}$. We utilize a Multilayer Perceptron (MLP) to learn features from nodes in the action spaces and softmax these features of nodes to select the optimal node with the highest probability. Besides, the $\psi(S_{P_{i+1}})$ and $\mathcal{X}^2(S_{P_{i+1}})$ values should become larger after taking the specific action.

Reward. To guide the model towards the most abnormal direction, we use the gain of abnormal degree from specific actions as the reward, optimizing as shown in Figure 3. For instance, the model selects actions like expanding node 9 at T_2 and excluding node 4 at T_1 for higher rewards. We first define the abnormal degree for the subgraph $G[S_P]$ at the i-th step as:

$$d_a(S_{P_i}) = \frac{\psi(S_{P_i})}{\rho_T}, \tag{6}$$

where $\rho_T = \frac{h(S)}{|S||T_s|}$ denotes the temporal density of the subgraph $G[S_P]$ at the i-th step.

Based on the Theorem 3.3, a temporal abnormal subgraph should satisfy $\psi(S) \gg \rho_T$. Therefore, we optimize the problem by policy gradient to maximize the gain of abnormal degrees after taking the specific actions, i.e., reward $= d_a(S_{P_{i+1}}) - d_a(S_{P_i})$.

Theorem 4.4. *The time complexity of refinement is $O(|\mathcal{G}[S_{a_t}]| \cdot t)$, where t denotes the number of states in the optimization process and $|\mathcal{G}[S_{a_t}]|$ represents the number of training anomalous subgraphs.*

5 EXPERIMENTS

5.1 Experimental Setup

5.1.1 Datasets. Four real financial datasets, namely PlusTokenPonzi, ETH-Jan-18, ETH-Jan-19, and Blur, are used to evaluate models. The PlusTokenPonzi dataset [1] is a real Ethereum blockchain transaction dataset that involves labeled money laundering activities from EthereumHeist [43]. This dataset has 6122 ground truth money laundering labels. The ETH datasets consist of real transaction data obtained from the Ethereum blockchain, available through Google BigQuery[2]. The Blur dataset is a real transaction dataset

[1] https://github.com/lindan113/EthereumHeist?tab=readme-ov-file
[2] https://www.kaggle.com/bigquery/ethereum-blockchain

from the NFT marketplace [58]. Statistic details are in Table 1. Details about these datasets are available in the appendix.

Table 1: Statistics of experimented datasets.

Datasets	# Nodes	# Edges	# Transactions
PlusTokenPonzi	38,324	58,731	64,858
ETH-Jan-2018	1,761,571	2,749,707	4,279,799
ETH-Jan-2019	2,199,347	3,331,594	6,128,061
Blur	175,071	1,103,791	3,539,773

5.1.2 Preprocessing. Following AntiBenford [6], transactions with values less than 1 unit are excluded during preprocessing. Since our goal is to discover anomalous subgraphs in temporal networks, we retain the timestamps associated with the transactions.

The temporal graph is divided into multiple snapshots at equal time intervals, with a sequential relationship between these snapshots. For each time interval, we represent multiple transactions between two nodes as a single edge. Each edge is associated with both the transaction frequency and the financial distribution, which includes the probabilities of transaction amounts starting with each digit d. This forms the input graph for our analysis.

As ETH and Blur datasets lack inherent anomaly labels, we use the advanced methods [16] and [6] as a reference. Specifically, we identify the densest temporal subgraphs satisfying $\psi(S) \gg \rho_T$ (see Theorem 3.3), and randomly select 80 of them for the training set. Since the PlusTokenPonzi dataset includes anomaly labels itself, we randomly pick 20% of the ground truth data as the training set.

5.1.3 Competitors. We compare our proposed model with thirteen other methods. More details are provided in Section 2.

Non-temporal anomaly detection methods: **Holoscope** [21], **FlowScope** [14] and **AntiBenford** [6] propose non-learning-based methods for anomalous subgraph discovery. While **CLARE** [44], **AS-GAE** [54], **GCAD** [59], and **SIGNET** [23] utilize learning-based methods for anomaly detection.

Temporal anomaly detection methods: To detect anomalies in dynamic graphs, **DeepSphere** [41] combines deep autoencoders with a hypershere learning algorithm. **RustGraph** [9] detects anomalies by jointly learning structural-temporal dependency in temporal graphs. **AnoGraph** [3] proposes a sketch algorithm to detect anomalies in streaming graphs.

Temporal densest subgraph discovery methods: **FAST-GA** [37] proposes a greedy approach to detect temporal dense subgraphs. **OTCD** [49] and **TopLC** [16] propose to utilize scalable temporal cores for temporal cohesive subgraph mining.

5.1.4 Hyperparameter Settings. We perform hyperparameter tuning using grid search. We tune the parameters as follows: the duration of time interval for snapshots in {5, 10, 20, 30} minutes, the maximum size m of nodes in the boundary in {100, 150, 200, 250}, the range of observation window W_r for temporal dense subgraphs in {30, 60, 90, 120} minutes, the size of the top-p candidates in {100, 150, 200, 250}, the number of sampling pairs in {5, 10, 15, 20}, and the number of training subgraphs in {20, 50, 80, 110}.

5.1.5 Performance Metrics. We evaluate performance using standard metrics from [6] and [17], including the $\mathcal{X}^2$-statistic, temporal subgraph density ρ_T and $\psi(S)$, which is the average $\mathcal{X}^2$ per node of the subgraph $G[S]$. The $\mathcal{X}^2(S)$ metric evaluates the deviation of the subgraph's financial distribution from Benford's law, while $\psi(S)$ calculates the average deviation, represented as $\frac{\mathcal{X}^2(S)}{|S|}$. This average is crucial because large $\mathcal{X}^2(S)$ values in larger subgraphs may lead to deceptively small $\psi(S)$ values. Thus, $\psi(S)$ provides a fair comparison across subgraphs of varying sizes.

According to Theorem 3.3, a temporal subgraph with nodes S is considered abnormal if $\psi(S) \gg \rho_T$, we use the abnormal degree $d_a(S) = \frac{\psi(S)}{\rho_T}$ as a metric. A higher $d_a(S)$ indicates a more abnormal temporal subgraph.

Furthermore, since the PlusTokenPonzi dataset includes ground truth data, we evaluate the overlapping similarity between the anomalies detected by each method and the ground truth on the PlusTokenPonzi dataset. For this, we adopt the ONMI (Overlapping Normalized Mutual Information) metric, as used in [28, 44], which is an overlapping variant of the NMI score.

5.2 Effectiveness Evaluation

We evaluate the effectiveness of all methods by comparing their top-a results in Tables 2, 3, ranked by $\psi(S)$. Note that the results for the ETH-Jan-2019 and Blur datasets have been included in the appendix due to space limitations. We do not choose $d_a(S) = \frac{\psi(S)}{\rho_T}$ because it returns subgraphs with very low density ρ_T, which are not dense and fail to meet the criteria for temporal abnormal subgraphs. To ensure stability, we report the average results over five runs. We observe:

(1) Our proposed TempASD achieves the best performance in detecting anomalies when compared with the ground truth on the PlusTokenPonzi dataset in Table 2. ONMI measures the similarity between detected and ground truth subgraphs, with higher scores indicating greater overlap. Specifically, TempASD achieves the highest ONMI scores compared to other methods, indicating its superior accuracy in identifying anomalies.

(2) TempASD demonstrates the highest abnormal degree across all datasets compared to other baseline methods. On average, it is 7× more effective than the state-of-the-art in terms of abnormal degree. This confirms TempASD's capability to detect the temporal abnormal subgraphs in dynamic financial networks.

(3) TempASD achieves the highest $\psi(S)$ values across all datasets compared to other baselines. Although its $\mathcal{X}^2(S)$ values at the top 5, 15, and 20 on the Blur dataset are not the highest, their average $\mathcal{X}^2(S)$ values, i.e., $\psi(S)$ values, are the highest. Since $\psi(S)$ values measure the average $\mathcal{X}^2(S)$ per node, i.e., $\frac{\mathcal{X}^2(S)}{|S|}$, this metric is more fair for comparing subgraphs with varying sizes.

(4) The temporal density (ρ_T) values of TempASD are comparable to those of other baselines, even though some are not the highest. However, the abnormal degree values of TempASD consistently remain the highest. Some baselines, such as FlowScope, may have some of the highest temporal density values, but their abnormal degree values are very low, indicating that they are not abnormal subgraphs at all.

(5) The abnormal degree values decrease as the a values increase. This occurs because a larger a includes more subgraphs with lower abnormal degrees. Nevertheless, TempASD can still identify temporal subgraphs with a higher abnormal degree.

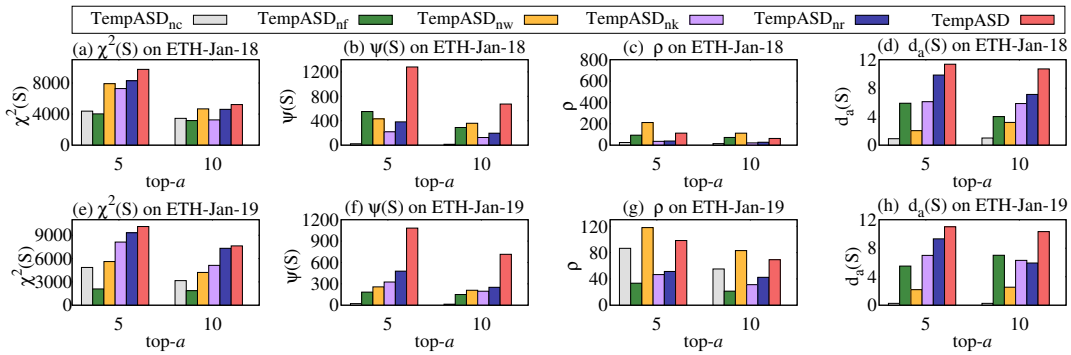

Figure 4: The ablation study w.r.t. four evaluation metrics.

Table 2: Effectiveness on PlusTokenPonzi. For each method, we report average results over five runs. The best method should extract subgraphs with higher abnormal degree $d_a(S)$ and relatively higher density ρ_T. Our method TempASD meets it.

top-a	Metrics	Holoscope	FlowScope	AntiBenford	CLARE	AS-GAE	GCAD	SIGNET	FAST-GA	TopLC	OTCD	DeepSphere	RustGraph	AnoGraph	TempASD
5	$\mathcal{X}^2(S)$	1020.57	4278.46	4965.96	1983.10	1453.70	724.71	2311.75	1583.32	1031.54	1238.73	263.62	995.25	1099.51	**7682.23**
	$\psi(S)$	20.85	32.03	16.16	12.14	13.64	39.18	27.76	26.52	34.18	71.98	34.75	4.10	12.35	**903.71**
	ρ_T	13.83	28.33	14.98	12.44	18.44	32.39	33.63	25.29	53.02	**85.05**	28.92	5.40	28.97	81.34
	$d_a(S)$	1.51	1.13	1.08	0.98	0.74	1.21	0.83	1.05	0.64	0.85	1.20	0.76	0.43	**11.11**
	ONMI	0.30	0.02	0.21	0.56	0.20	0.40	0.00	0.28	0.00	0.23	0.34	0.00	0.01	**0.78**
10	$\mathcal{X}^2(S)$	816.36	3743.96	4575.90	2943.27	818.51	359.44	1242.36	889.37	911.85	1172.41	147.69	721.27	1077.05	**6790.91**
	$\psi(S)$	14.20	21.62	11.75	8.46	8.98	26.23	18.58	21.20	22.91	45.54	18.43	2.74	8.43	**456.24**
	ρ_T	8.20	16.99	12.56	6.78	14.91	44.00	17.32	18.52	36.82	**58.98**	14.50	3.95	24.82	43.85
	$d_a(S)$	1.73	1.27	0.94	1.25	0.60	0.60	1.07	1.15	0.62	0.77	1.27	0.69	0.34	**10.40**
	ONMI	0.27	0.01	0.13	0.53	0.20	0.40	0.00	0.22	0.00	0.18	0.29	0.00	0.00	**0.64**
15	$\mathcal{X}^2(S)$	619.63	3469.80	4049.66	4310.50	595.26	272.53	1131.72	795.96	901.25	946.46	104.94	563.65	2653.54	**6711.95**
	$\psi(S)$	10.97	15.55	9.81	6.50	7.16	19.34	15.00	18.79	17.18	33.18	12.81	2.03	5.92	**305.46**
	ρ_T	5.92	12.71	10.93	5.55	13.63	39.77	11.57	12.82	27.60	**43.47**	14.88	2.96	18.63	38.84
	$d_a(S)$	1.85	1.22	0.90	1.17	0.52	0.49	1.30	1.47	0.62	0.76	0.86	0.69	0.32	**7.87**
	ONMI	0.21	0.00	0.09	0.46	0.18	0.39	0.00	0.18	0.00	0.13	0.27	0.00	0.00	**0.50**
20	$\mathcal{X}^2(S)$	561.68	3095.10	3750.38	3241.47	479.28	238.11	895.65	700.76	870.44	774.77	88.75	424.04	2489.22	**6482.48**
	$\psi(S)$	8.97	12.07	8.21	5.17	6.04	15.37	12.65	15.68	13.76	25.75	10.90	1.55	5.55	**229.75**
	ρ_T	4.79	10.04	10.04	4.99	9.91	31.83	12.13	10.17	21.79	33.18	12.48	2.23	17.47	**33.25**
	$d_a(S)$	1.87	1.20	0.82	1.04	0.61	0.48	1.04	1.54	0.63	0.78	0.87	0.69	0.32	**6.91**
	ONMI	0.20	0.00	0.01	0.44	0.12	0.31	0.00	0.11	0.00	0.12	0.20	0.00	0.00	**0.50**

Table 3: Effectiveness on ETH-Jan-2018. For each method, we report average results over five runs.

top-a	Metrics	Holoscope	FlowScope	AntiBenford	CLARE	AS-GAE	GCAD	SIGNET	FAST-GA	TopLC	OTCD	DeepSphere	RustGraph	AnoGraph	TempASD
5	$\mathcal{X}^2(S)$	309.29	3966.68	7900.45	1745.85	449.84	724.71	1724.03	1598.60	1312.87	954.85	145.41	3332.36	2150.12	**9754.80**
	$\psi(S)$	33.87	92.69	2.39	21.51	4.04	7.11	7.59	30.17	151.27	111.33	23.89	38.96	542.23	**1280.87**
	ρ_T	22.77	**317.13**	0.94	19.70	23.86	17.78	11.67	68.25	192.32	101.31	68.45	51.21	137.36	112.69
	$d_a(S)$	1.49	0.29	2.55	1.09	0.17	0.40	0.65	0.44	0.79	1.10	0.35	0.76	3.95	**11.37**
10	$\mathcal{X}^2(S)$	694.76	2635.44	4001.75	1044.07	287.68	359.44	996.69	1684.03	824.03	520.43	78.38	2954.33	2295.02	**5225.04**
	$\psi(S)$	23.51	50.36	1.70	11.63	2.58	5.98	4.41	25.33	109.42	66.77	14.52	32.55	468.27	**673.59**
	ρ_T	13.65	**260.18**	2.98	30.58	10.78	12.57	10.50	58.98	178.72	91.95	60.15	44.67	107.73	62.92
	$d_a(S)$	1.72	0.19	0.57	0.38	0.24	0.48	0.42	0.43	0.61	0.73	0.24	0.73	4.35	**10.71**
15	$\mathcal{X}^2(S)$	674.76	2202.56	2702.57	801.36	257.98	272.53	693.55	1607.24	601.84	354.44	56.21	2178.36	1761.74	**3566.73**
	$\psi(S)$	18.43	35.82	1.40	3.86	2.03	4.84	3.09	20.22	90.44	44.59	10.76	39.60	239.52	**454.19**
	ρ_T	25.22	**261.43**	2.27	21.36	8.89	8.52	8.61	50.40	160.00	71.04	63.78	70.80	89.09	43.77
	$d_a(S)$	0.73	0.14	0.62	0.18	0.23	0.57	0.36	0.40	0.57	0.63	0.17	0.56	2.69	**10.38**
20	$\mathcal{X}^2(S)$	591.19	2042.13	2029.30	616.65	260.12	238.11	529.02	1602.92	458.06	271.04	46.37	1264.23	1355.93	**2702.99**
	$\psi(S)$	14.87	30.88	1.21	7.64	1.52	2.58	2.42	20.17	68.05	33.49	8.59	27.51	131.49	**342.24**
	ρ_T	22.13	**264.16**	1.75	20.29	11.20	9.31	7.03	50.28	127.59	55.11	65.41	37.92	71.02	33.12
	$d_a(S)$	0.67	0.12	0.69	0.38	0.14	0.28	0.34	0.40	0.53	0.61	0.13	0.73	1.85	**10.33**

5.3 Ablation Study

The Blur and PlusTokenPonzi datasets, as well as the results for the top 15 and top 20, are excluded due to similar behavior patterns. TempASD$_{\mathbf{nr}}$ omits temporal anomaly refinement (Section 4.4). TempASD$_{\mathbf{nw}}$ disregards weights, i.e., transaction frequency, for the temporal GNN (Section 4.2). TempASD$_{\mathbf{nk}}$ excludes the KL-divergence of financial distributions between pairs (Section 4.2). TempASD$_{\mathbf{nc}}$ substitutes temporal dense subgraphs with k-ego nets (Section 4.3). TempASD$_{\mathbf{nf}}$ randomly selects connected nodes as the boundary nodes (Section 4.4). From Figure 4, we observe:

(1) TempASD outperforms TempASD$_{\mathbf{nr}}$ on both datasets, demonstrating the significance of temporal anomaly refinement in improving candidates towards the abnormal direction.

(2) TempASD$_{\mathbf{nw}}$ achieves higher density ρ_T than TempASD, but it gets a very low abnormal degree $d_a(S)$, which means that the subgraph is not abnormal. This underscores the effect of incorporating transaction frequency in the temporal GNN.

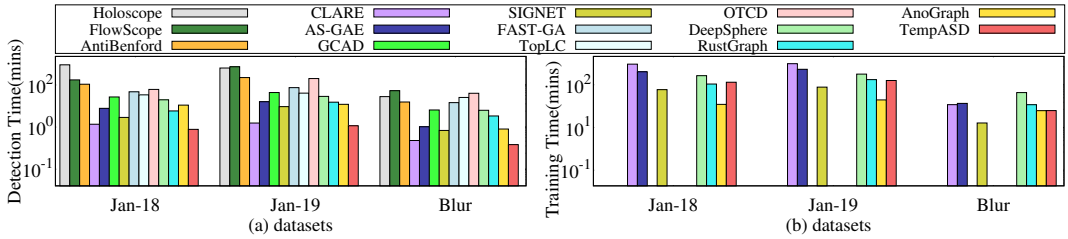

Figure 5: (a) Detection time; (b) training time. Note that non-learning-based methods require zero training time.

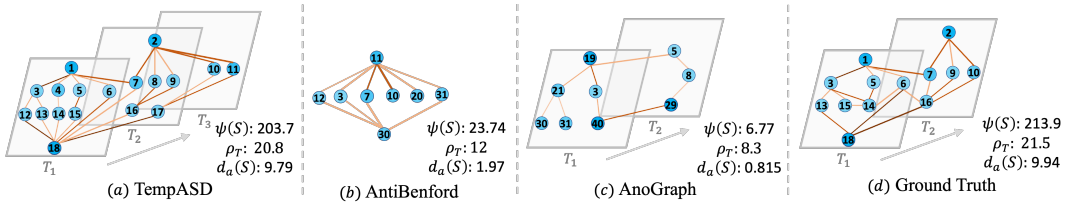

Figure 6: The abnormal subgraph detected by (a) TempASD; (b) AntiBenford; (c) AnoGraph, with the ground truth shown in (d). The color of the edges represents the frequency of transactions.

(3) TempASD outperforms TempASD$_{\mathbf{nk}}$ on both datasets, highlighting the effect of using KL-divergence in measuring the transaction similarity between subgraph pairs.

(4) TempASD significantly outperforms TempASD$_{\mathbf{nc}}$ on both datasets, highlighting the critical role of temporal dense subgraphs in candidate discovery.

(5) TempASD outperforms TempASD$_{\mathbf{nf}}$ on both datasets, highlighting the importance of using the transaction frequency to select boundary nodes.

5.4 Efficiency Evaluation

We use Intel Xeon Platinum 8370C processor 32-core 2.8GHz CPU with NVIDIA GeForce RTX 4090 24G GPU for evaluation.

Figure 5 (a) presents the detection time across three datasets, while Figure 5 (b) shows the training time. Due to space limitations, we have moved the results of the PlusTokenPonzi dataset to the appendix. Non-learning-based methods have zero training times as they do not involve a training phase. TempASD demonstrates the lowest detection time across the three datasets and has comparable training time to other baselines.

5.5 Case Study

We display the detected temporal abnormal subgraphs by TempASD in the PlusTokenPonzi dataset, which includes the ground truth, as shown in Figure 6 (a). We use the default time interval for snapshots in this case study, i.e., ten minutes. TempASD successfully identifies temporal dense subgraphs across three time intervals, T_1, T_2, and T_3, with a high $\psi(S)$ value, indicating that the financial transaction distribution significantly deviates from Benford's law. Furthermore, the abnormal degree $d_a(S)$ is the highest among the three models. The percentage overlap between detected and ground truth subgraphs is computed as $\frac{N_{\text{overlap}}}{N_{\text{gt}}+N_{\text{pm}}-N_{\text{overlap}}}$, where N_{overlap} represents overlapping nodes between the ground truth (N_{gt}) and model-detected nodes (N_{pm}). The overlap between the subgraph detected by TempASD and the ground truth is 72.2%, demonstrating that TempASD accurately detects the abnormal subgraph.

In contrast, AntiBenford, which is designed for static financial networks, can only detect non-temporal dense subgraphs, as shown in Figure 6 (b). The overlap between the subgraph discovered by AntiBenford and the ground truth is only 16.7%. This limitation motivated us to establish the theoretical foundation for detecting abnormal subgraphs in a temporal setting.

Although AnoGraph is designed for temporal networks, it does not incorporate transaction information, leading to detected temporal subgraphs that do not deviate from Benford's law (with a very low $\psi(S)$ value), which means that the transaction information of the detected temporal subgraph is normal. Additionally, the overlap between the subgraph detected by AnoGraph and the ground truth is only 10%.

6 LIMITATIONS AND THREATS TO VALIDITY

Anomalies intentionally crafted to conform to Benford's Law potentially evade detection by methods that rely solely on this principle. However, our approach overcomes this by incorporating structural and temporal features, including subgraph density, transaction patterns, and temporal evolution. These complementary insights make our method robust against sophisticated evasion attempts.

7 CONCLUSIONS

In this paper, we address the challenge of detecting temporal anomalous subgraphs in dynamic financial networks. We begin with a formal theoretical analysis of temporal abnormal subgraphs, i.e., temporal AntiBenford subgraphs. Building on this, we introduce a novel model named TempASD. This model first encodes temporal subgraphs using temporal graph neural networks. Next, it incorporates a temporal candidate detection module for obtaining candidates. Finally, we present a temporal anomaly refinement module, which employs a reinforcement learning-based algorithm designed to optimize toward the most abnormal directions. We conduct extensive evaluations against thirteen advanced models, demonstrating that the TempASD model outperforms state-of-the-art models in both effectiveness and efficiency. Additionally, TempASD scales smoothly to very large dynamic financial networks.

REFERENCES

[1] Abhishek Awasthi. 2012. Clustering algorithms for anti-money laundering using graph theory and social network analysis. (2012).

[2] Frank Benford. 1938. The law of anomalous numbers. *Proceedings of the American philosophical society* (1938), 551–572.

[3] Siddharth Bhatia, Mohit Wadhwa, Kenji Kawaguchi, Neil Shah, Philip S Yu, and Bryan Hooi. 2023. Sketch-based anomaly detection in streaming graphs. In *SIGKDD*. 93–104.

[4] 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 *CIKM*. 3747–3756.

[5] Ning Chen, Bernardete Ribeiro, and An Chen. 2016. Financial credit risk assessment: a recent review. *Artificial Intelligence Review* 45 (2016), 1–23.

[6] Tianyi Chen and Charalampos Tsourakakis. 2022. Antibenford subgraphs: Unsupervised anomaly detection in financial networks. In *SIGKDD*.

[7] Lingyang Chu, Yanyan Zhang, Yu Yang, Lanjun Wang, and Jian Pei. 2019. Online density bursting subgraph detection from temporal graphs. *PVLDB* 12, 13 (2019), 2353–2365.

[8] Yuan Gao, Junfeng Fang, Yongduo Sui, Yangyang Li, Xiang Wang, Huamin Feng, and Yongdong Zhang. 2024. Graph Anomaly Detection with Bi-level Optimization. In *The Web Conference 2024*. 4383–4394.

[9] Jianhao Guo, Siliang Tang, Juncheng Li, Kaihang Pan, and Lingfei Wu. 2023. RustGraph: Robust Anomaly Detection in Dynamic Graphs by Jointly Learning Structural-Temporal Dependency. *TKDE* (2023).

[10] Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. 2018. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *WSDM*. 261–269.

[11] Xuanwen Huang, Yang Yang, Yang Wang, Chunping Wang, Zhisheng Zhang, Jiarong Xu, Lei Chen, and Michalis Vazirgiannis. 2022. Dgraph: A large-scale financial dataset for graph anomaly detection. *NeurIPS* 35 (2022), 22765–22777.

[12] Arijit Khan and Cuneyt Gurcan Akcora. 2022. Graph-based Management and Mining of Blockchain Data. In *CIKM*. 5140–5143.

[13] Boris Kovalerchuk and Evgenii Vityaev. 2005. Data mining for financial applications. *Data Mining and Knowledge Discovery Handbook* (2005).

[14] Xiangfeng Li, Shenghua Liu, Zifeng Li, Xiaotian Han, Chuan Shi, Bryan Hooi, He Huang, and Xueqi Cheng. 2020. Flowscope: Spotting money laundering based on graphs. In *AAAI*, Vol. 34. 4731–4738.

[15] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *ICLR*.

[16] Longlong Lin, Pingpeng Yuan, Rong-Hua Li, and Hai Jin. 2021. Mining Diversified Top-r r Lasting Cohesive Subgraphs on Temporal Networks. *IEEE Transactions on Big Data* 8, 6 (2021), 1537–1549.

[17] Longlong Lin, Pingpeng Yuan, Rong-Hua Li, Chunxue Zhu, Hongchao Qin, Hai Jin, and Tao Jia. 2024. QTCS: Efficient Query-Centered Temporal Community Search. *PVLDB* 17, 6 (2024), 1187–1199.

[18] Zhe Lin, Fan Zhang, Xuemin Lin, Wenjie Zhang, and Zhihong Tian. 2021. Hierarchical core maintenance on large dynamic graphs. *PVLDB* 14, 5 (2021), 757–770.

[19] Chenlei Liu, Yuhua Xu, and Zhixin Sun. 2024. Directed dynamic attribute graph anomaly detection based on evolved graph attention for blockchain. *KAIS* 66, 2 (2024), 989–1010.

[20] Qiang Liu, Zhaocheng Liu, Haoli Zhang, Yuntian Chen, and Jun Zhu. 2021. Mining cross features for financial credit risk assessment. In *CIKM*. 1069–1078.

[21] Shenghua Liu, Bryan Hooi, and Christos Faloutsos. 2017. Holoscope: Topology-and-spike aware fraud detection. In *CIKM*. 1539–1548.

[22] Xuanming Liu, Tingjian Ge, and Yinghui Wu. 2020. A stochastic approach to finding densest temporal subgraphs in dynamic graphs. *TKDE* 34, 7 (2020), 3082–3094.

[23] Yixin Liu, Kaize Ding, Qinghua Lu, Fuyi Li, Leo Yu Zhang, and Shirui Pan. 2024. Towards Self-Interpretable Graph-Level Anomaly Detection. *NeurIPS* 36 (2024).

[24] Lin-Tao Lv, Na Ji, and Jiu-Long Zhang. 2008. A RBF neural network model for anti-money laundering. In *2008 International conference on wavelet analysis and pattern recognition*, Vol. 1. IEEE, 209–215.

[25] Shuai Ma, Renjun Hu, Luoshu Wang, Xuelian Lin, and Jinpeng Huai. 2017. Fast computation of dense temporal subgraphs. In *ICDE*. IEEE, 361–372.

[26] Shuai Ma, Renjun Hu, Luoshu Wang, Xuelian Lin, and Jinpeng Huai. 2019. An efficient approach to finding dense temporal subgraphs. *TKDE* 32, 4 (2019), 645–658.

[27] Emaad Manzoor, Sadegh M Milajerdi, and Leman Akoglu. 2016. Fast memory-efficient anomaly detection in streaming heterogeneous graphs. In *SIGKDD*. 1035–1044.

[28] Aaron F McDaid, Derek Greene, and Neil Hurley. 2011. Normalized mutual information to evaluate overlapping community finding algorithms. *arXiv preprint arXiv:1110.2515* (2011).

[29] Michael Mitzenmacher and Eli Upfal. 2017. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press.

[30] Misael Mongiovi, Petko Bogdanov, Razvan Ranca, Evangelos E Papalexakis, Christos Faloutsos, and Ambuj K Singh. 2013. Netspot: Spotting significant anomalous regions on dynamic networks. In *SIAM SDM*. SIAM, 28–36.

[31] Simon Newcomb. 1881. Note on the frequency of use of the different digits in natural numbers. *American Journal of mathematics* 4, 1 (1881), 39–40.

[32] Thien Hai Nguyen, Kiyoaki Shirai, and Julien Velcin. 2015. Sentiment analysis on social media for stock movement prediction. *ESWA* 42, 24 (2015).

[33] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. 2005. Uncovering the overlapping community structure of complex networks in nature and society. *nature* 435, 7043 (2005), 814–818.

[34] Ebberth L Paula, Marcelo Ladeira, Rommel N Carvalho, and Thiago Marzagão. [n. d.]. Deep learning anomaly detection as support fraud investigation in brazilian exports and anti-money laundering. In *2016 15th ieee international conference on machine learning and applications (icmla)*.

[35] Hongchao Qin, Rong-Hua Li, Ye Yuan, Guoren Wang, Lu Qin, and Zhiwei Zhang. 2022. Mining bursting core in large temporal graphs. *PVLDB* (2022).

[36] Benedek Rozemberczki, Paul Scherer, Yixuan He, George Panagopoulos, Alexander Riedel, Maria Astefanoaei, Oliver Kiss, Ferenc Beres, Guzman Lopez, Nicolas Collignon, et al. 2021. Pytorch geometric temporal: Spatiotemporal signal processing with neural machine learning models. In *CIKM*. 4564–4573.

[37] Polina Rozenshtein, Nikolaj Tatti, and Aristides Gionis. 2017. Finding dynamic dense subgraphs. *TKDD* 11, 3 (2017), 1–30.

[38] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. 2018. Structured sequence modeling with graph convolutional recurrent networks. In *ICONIP*. Springer, 362–373.

[39] Jun Tang and Jian Yin. 2005. Developing an intelligent data discriminating system of anti-money laundering based on SVM. In *2005 International conference on machine learning and cybernetics*, Vol. 6. IEEE.

[40] Xian Teng, Yu-Ru Lin, and Xidao Wen. 2017. Anomaly detection in dynamic networks using multi-view time-series hypersphere learning. In *CIKM*. 827–836.

[41] Xian Teng, Muheng Yan, Ali Mert Ertugrul, and Yu-Ru Lin. 2018. Deep into hypersphere: Robust and unsupervised anomaly discovery in dynamic networks. In *IJCAI*.

[42] Su-Nan Wang and Jian-Gang Yang. 2007. A money laundering risk evaluation method based on decision tree. In *2007 international conference on machine learning and cybernetics*, Vol. 1. IEEE, 283–286.

[43] Jiajing Wu, Dan Lin, Qishuang Fu, Shuo Yang, Ting Chen, Zibin Zheng, and Bowen Song. 2024. Toward Understanding Asset Flows in Crypto Money Laundering Through the Lenses of Ethereum Heists. *IEEE Transactions on Information Forensics and Security* 19 (2024), 1994–2009.

[44] Xixi Wu, Yun Xiong, Yao Zhang, Yizhu Jiao, Caihua Shan, Yiheng Sun, Yangyong Zhu, and Philip S Yu. 2022. Clare: A semi-supervised community detection algorithm. In *SIGKDD*. 2059–2069.

[45] Yucheng Wu, Leye Wang, Xiao Han, and Han-Jia Ye. [n. d.]. Graph Contrastive Learning with Cohesive Subgraph Awareness. In *The Web Conference 2024*.

[46] Bingbing Xu, Huawei Shen, Bingjie Sun, Rong An, Qi Cao, and Xueqi Cheng. 2021. Towards consumer loan fraud detection: Graph neural networks with role-constrained conditional random field. In *AAAI*, Vol. 35. 4537–4545.

[47] Yiming Xu, Bin Shi, Teng Ma, Bo Dong, Haoyi Zhou, and Qinghua Zheng. 2023. CLDG: Contrastive learning on dynamic graphs. In *ICDE*. 696–707.

[48] Jaewon Yang and Jure Leskovec. 2013. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *WSDM*.

[49] Junyong Yang, Ming Zhong, Yuanyuan Zhu, Tieyun Qian, Mengchi Liu, and Jeffrey Xu Yu. 2023. Scalable time-range k-core query on temporal graphs. *PVLDB* 16, 5 (2023), 1168–1180.

[50] Yi Yang, Da Yan, Huanhuan Wu, James Cheng, Shuigeng Zhou, and John CS Lui. 2016. Diversified temporal subgraph pattern mining. In *SIGKDD*. 1965–1974.

[51] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *NeurIPS* 33 (2020), 5812–5823.

[52] Wenchao Yu, Wei Cheng, Charu C Aggarwal, Kai Zhang, Haifeng Chen, and Wei Wang. 2018. Netwalk: A flexible deep embedding approach for anomaly detection in dynamic networks. In *SIGKDD*. 2672–2681.

[53] Daniele Zambon, Cesare Alippi, and Lorenzo Livi. 2018. Concept drift and anomaly detection in graph streams. *TNNLS* 29, 11 (2018), 5592–5605.

[54] Zheng Zhang and Liang Zhao. 2022. Unsupervised Deep Subgraph Anomaly Detection. In *ICDM*. 753–762.

[55] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. 2019. T-GCN: A temporal graph convolutional network for traffic prediction. *TITS* 21, 9 (2019), 3848–3858.

[56] Yu Zhao, Huaming Du, Ying Liu, Shaopeng Wei, Xingyan Chen, Fuzhen Zhuang, Qing Li, and Gang Kou. 2022. Stock movement prediction based on bi-typed hybrid-relational market knowledge graph via dual attention networks. *TKDE* (2022).

[57] 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*, Vol. 3. 7.

[58] Chenyu Zhou, Hongzhou Chen, Hao Wu, Junyu Zhang, and Wei Cai. 2024. ARTEMIS: Detecting Airdrop Hunters in NFT Markets with a Graph Learning System. In *The Web Conference 2024*. 1824–1834.

[59] Zhong Zhuang, Kai Ming Ting, Guansong Pang, and Shuaibin Song. 2023. Subgraph Centralization: A Necessary Step for Graph Anomaly Detection. In *SIAM SDM*. 703–711.

A MODEL ANALYSIS

A.1 Model Details

The framework of the **temporal subgraph encoder** is shown in Figure 7, which includes temporal graph neural networks for modeling the *temporal graph structure* and the *financial distribution*. The encoder parameters are trained by optimizing the loss function $\mathcal{L}$ in Equation 3.

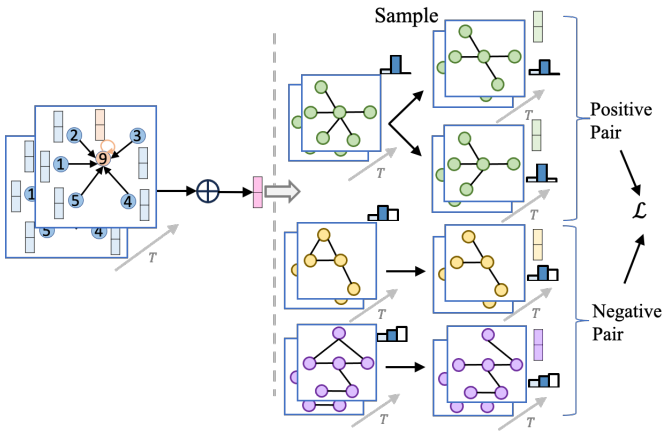

Figure 7: The illustration of the temporal subgraph encoder.

The overall process of the temporal subgraph encoder is shown in Algorithm 1. The key idea is to train both positive and negative subgraph pairs to encode abnormal subgraphs with consideration for both graph structure and financial distribution. Firstly, it generates positive and negative subgraph pairs by sampling from the training anomalous subgraphs. Since our model is semi-supervised, we use the few-labeled abnormal subgraphs as the training samples to learn the hidden features behind anomalies in financial networks. Second, the temporal graph neural network is applied to the sampled subgraphs for graph connectivity and transaction frequency modeling. Then, the parameters ϕ are optimized by maximizing the vector similarities between positive subgraph pairs and minimizing the vector similarities between negative subgraph pairs.

Algorithm 1: Temporal Subgraph Encoder

Input : The training anomalous subgraphs $\mathcal{G}[S_{a_t}]$, a temporal financial network $G(V, \mathcal{E})$
Output: The trained parameters ϕ in the abnormal subgraph encoder

while *not converge* **do**
  Generate the positive and negative samples based on $\mathcal{G}[S_{a_t}]$
  Encode sampled subgraphs by temporal graph neural networks
  Update parameters ϕ by minimizing the loss $\mathcal{L}$ based on Equation 3
return The trained parameters ϕ

The framework of the **temporal candidate detection** process is depicted in Figure 8, which includes the construction of temporal dense subgraphs, early pruning based on $\mathcal{X}^2(\widehat{S})$ (the deviation from Benford's law), and vector space matching. These steps collectively ensure the efficient identification of temporal abnormal candidates.

The overall process of our proposed temporal candidate detection is shown in Algorithm 2. The main idea is to identify the most likely temporal abnormal subgraphs that significantly deviate from Benford's law based on the training samples. Firstly, the construction of temporal dense subgraphs is performed for all nodes. Then, the top-p abnormal subgraphs are selected based on $\mathcal{X}^2(\widehat{S})$ scores. Following this, the temporal GNN in Sec. 4.2 is utilized to encode $\mathcal{G}[\widehat{S}]$ and the training anomalous subgraphs $\mathcal{G}[S_{a_t}]$. Finally, the subgraphs $\mathcal{G}[\widehat{S}]$ whose encoded vectors are similar to the encoded vectors of $\mathcal{G}[S_{a_t}]$ are matched as temporal abnormal candidates $\mathcal{G}[S_P]$.

Algorithm 2: Temporal Candidate Detection

Input : The training anomalous subgraphs $\mathcal{G}[S_{a_t}]$, a temporal financial network $G(V, \mathcal{E})$
Output: Temporal abnormal candidates $\mathcal{G}[S_P]$

$\mathcal{G}[S_P] \leftarrow \emptyset$
for *Every node* $v \in V$ **do**
  Obtain the temporal dense subgraph of v
  Calculate $\mathcal{X}^2(\widehat{S})$ of the subgraph $G[\widehat{S}]$ using Equation 4
Select the top-p temporal abnormal subgraphs based on $\mathcal{X}^2(\widehat{S})$
Encode the selected subgraphs using temporal GNN
for *Every subgraph* $G[S_{a_t}] \in \mathcal{G}[S_{a_t}]$ **do**
  Encode $G[S_{a_t}]$ using temporal GNN
  Find the subgraphs $\mathcal{G}[\widehat{S}]$ whose encoded vectors matched with the encoded vector of $G[S_{a_t}]$
  $\mathcal{G}[S_P] = \mathcal{G}[S_P] \cup \mathcal{G}[\widehat{S}]$
return $\mathcal{G}[S_P]$

The overall process of **temporal anomaly refinement** is shown in Algorithm 3. The key idea is to refine the subgraphs by maximizing the rewards after taking specific actions. Firstly, it constructs the boundaries of the training subgraphs. Then, we obtain the node representations of all nodes in the action spaces of exclusion and expansion. Next, the actions are chosen for exclusion and expansion. After that, the abnormal degrees are calculated for the next step. We optimize the parameters by maximizing the rewards. Finally, the anomalous candidates are refined by the trained model.

A.2 Notations of Our Model

The notations of our method are in Table 4.

A.3 Model Discussion

Here we give a discussion of the TempASD model as follows:

• The TempASD model is *effective*. The reason is that we detect candidates by matching the temporal subgraphs with the training anomalies in the encoded vector space, then use a temporal anomaly refinement module to optimize the detected candidates towards the most abnormal direction.

• The TempASD model is *efficient*. The overall time complexity of the TempASD model is $O(deg_{\text{avg}} \cdot |V| + |V_{W_r}| \cdot |\mathcal{E}_{W_r}| + |\mathcal{G}[S_{a_t}]| \cdot t)$. Since $|\mathcal{G}[S_{a_t}]|$ and t are constants, the time complexity can be reduced to $O(deg_{\text{avg}} \cdot |V| + |V_{W_r}| \cdot |\mathcal{E}_{W_r}|)$. Since the observation

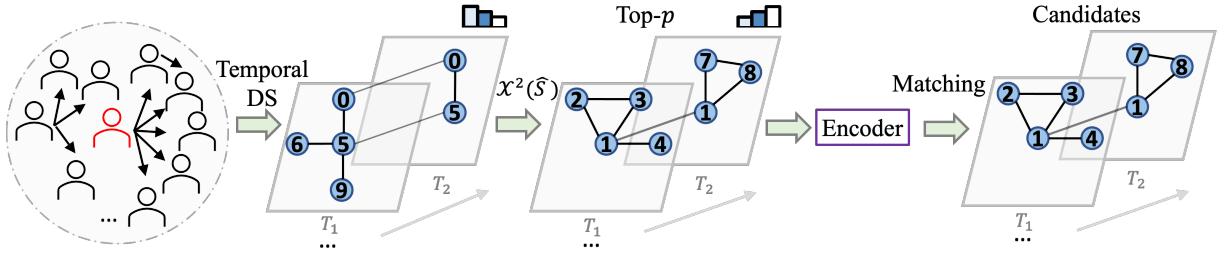

Figure 8: The illustration of temporal candidate detection, where $\mathcal{X}^2(\widehat{S})$ represents the deviation of $G[\widehat{S}]$ from Benford's law, and DS denotes dense subgraph. The encoder refers to the trained parameters in Sec. 4.2, which embeds the temporal subgraphs into vector space. This module reduces the pool of candidates significantly.

Algorithm 3: Temporal Anomaly Refinement

Input : The training anomalous subgraphs $\mathcal{G}[S_{a_t}]$, the temporal financial network $G(V, \mathcal{E})$
Output : The anomalous subgraphs $\mathcal{G}[S_a]$

Obtain the candidates $\mathcal{G}[S_P]$ by Algorithm 1
while *not converge* **do**
  `/* Training the model by maximizing the rewards */`
  Construct the boundaries of the training subgraphs based on Equation 5
  Choose the action for both exclusion and expansion
  Calculate the abnormal degree in Equation 6
  Optimize the parameters by maximizing the rewards after taking specific actions
`/* Refining anomalous subgraphs */`
Generate the anomalous subgraphs $\mathcal{G}[S_a]$ via refining the candidates $\mathcal{G}[S_P]$ by the trained model above
return $\mathcal{G}[S_a]$

Table 4: Notations

Notation	Description
$G(V, \mathcal{E})$	A temporal financial network
$\mathcal{G}[S_{a_t}]$	The training temporal anomalous subgraphs
$\mathcal{G}[S_P]$	The temporal abnormal candidates
$G[\widehat{S}]$	The sub-optimal temporal subgraph
ρ_T	The temporal density
$e(G[S^a])$	The encoded vector of the subgraph $G[S^a]$
$w(G[S^a])$	The financial distribution of the subgraph $G[S^a]$
r_{s,c,T_i}	Transaction frequency between nodes s and c at the snapshot T_i

window W_r is typically short (e.g., around 1 hour in our experiment), the numbers of vertices and edges (e.g., around 7,000 vertices and 15,000 edges) are usually not large. In our experiment, we have $|V_{W_r}| \cdot |\mathcal{E}_{W_r}| < k_2 \cdot deg_{\text{avg}} \cdot |V|$, where k_2 is a constant. Thus, the complexity is dominated by $O(deg_{\text{avg}} \cdot |V|)$, which is near-linear w.r.t. graph size.

• The TempASD model is *scalable*, demonstrating a manageable increase in time complexity proportional to the size of the financial network and the time dimension span. It can guarantee the running time of the model when applying it to larger financial networks and longer-span transactions.

• The TempASD model is *parallelizable* in the temporal candidate detection module since the computations of the temporal dense subgraphs, and $\mathcal{X}^2(\widehat{S})$ values are independent among different vertices. They can be executed in parallel with multiprocessors to reduce the running time significantly.

• The TempASD model can be *generalized* to other types of networks and domains. Firstly, it can be extended to detect temporal anomalous subgraphs in other types of networks, e.g., social networks and citation networks. Specifically, the calculation of the top-p subgraphs with large $\mathcal{X}^2(\widehat{S})$ should be removed in the temporal candidate detection module, as it is only applicable to financial networks that follow Benford's law. Moreover, the reward in the temporal anomaly refinement module should be tailored to the downstream application, for example, considering the gain in density or F1 score. Secondly, the TempASD model can be extended to handle influential community search over temporal graphs. The temporal candidate detection module should be modified by adapting the top-p subgraphs with large $\mathcal{X}^2(\widehat{S})$ to temporal subgraphs with large importance values. Then the reward in the temporal anomaly refinement module should be adapted to the gain of the importance values after the specific actions.

B PROOF OF THEOREMS

We use the Chernoff bound [29] in the proofs.

Theorem B.1. (Chernoff bound). *Consider a set of mutually independent binary random variables* $\{X_1, \cdots, X_q\}$. *Let* $X = \sum_{i=1}^{q} X_i$ *be their sum. Then, for* $0 < \epsilon < 1$ *we have* $\Pr[|X - \mathbb{E}[X]| \geq \epsilon \mathbb{E}[X]] \leq 2e^{-\epsilon^2 \mathbb{E}[X]/3}$.

B.1 Proof of Theorems 3.1

Proof. By applying the Chernoff bound (see Theorem B.1), for any $0 < \epsilon < 1$, we have:

$$\Pr\left[\left|\frac{X_{S,d}}{|T_s|} - \frac{p_d h(S)}{|T_s|}\right| \geq \epsilon \frac{p_d h(S)}{|T_s|}\right] \leq 2e^{-\frac{\epsilon^2 p_d h(S)}{3|T_s|}}.$$

Since $\frac{2h(S)}{|S||T_s|} \geq \frac{36 \log n}{\delta \epsilon^2}$, we get $\frac{\epsilon^2 p_d h(S)}{|T_s|} \geq \frac{\epsilon^2 \delta h(S)}{|T_s|} \geq 18|S| \log n$.

Hence, $\Pr\left[\left|\frac{X_{S,d}}{|T_s|} - \frac{p_d h(S)}{|T_s|}\right| \geq \epsilon \frac{p_d h(S)}{|T_s|}\right] \leq 2e^{-6|S| \log n} \leq 2n^{-6|S|}$.

Table 5: Effectiveness on ETH-Jan-2019 dataset. For each method, we report average results over five runs.

top-a	Metrics	Holoscope	FlowScope	AntiBenford	CLARE	AS-GAE	GCAD	SIGNET	FAST-GA	TopLC	OTCD	DeepSphere	RustGraph	AnoGraph	TempASD
5	$\mathcal{X}^2(S)$	2374.97	8018.53	9094.98	3375.30	5542.81	790.27	1459.99	1718.37	3801.75	1387.80	3887.27	2143.54	3163.21	**10121.43**
	$\psi(S)$	23.52	318.55	67.94	60.60	40.38	5.24	11.98	45.33	168.56	86.54	145.18	69.43	206.08	**1082.29**
	ρ_T	22.03	**313.34**	15.14	53.92	63.29	9.37	12.53	52.20	102.50	106.80	54.48	51.53	196.23	98.33
	$d_a(S)$	1.07	1.02	4.49	1.12	0.64	0.56	0.96	0.87	1.64	0.81	2.67	1.35	1.05	**11.01**
10	$\mathcal{X}^2(S)$	1334.05	4492.26	4800.36	3110.18	3692.44	409.04	703.02	967.54	3441.38	916.11	2230.29	1079.25	2893.23	**7632.98**
	$\psi(S)$	18.81	168.96	35.38	44.06	24.71	4.62	6.33	29.86	137.50	47.47	102.45	45.91	139.05	**714.52**
	ρ_T	16.14	**157.11**	9.46	41.70	44.83	6.24	13.97	37.79	89.57	58.58	42.25	44.92	153.89	69.19
	$d_a(S)$	1.17	1.08	3.74	1.06	0.55	0.74	0.45	0.79	1.54	0.81	2.43	1.02	0.90	**10.33**
15	$\mathcal{X}^2(S)$	1193.95	3191.97	4183.12	2888.44	2587.29	382.13	668.01	703.64	2580.54	624.84	1804.71	729.30	2772.76	**3899.43**
	$\psi(S)$	16.67	117.46	24.16	36.78	20.32	2.73	6.33	23.79	96.57	31.62	73.94	34.27	107.55	**562.35**
	ρ_T	11.17	**161.21**	89.06	39.34	34.91	8.72	9.92	36.79	71.95	39.02	34.07	39.84	127.28	55.44
	$d_a(S)$	1.49	0.73	0.27	0.93	0.58	0.31	0.64	0.65	1.34	0.81	2.17	0.86	0.84	**10.14**
20	$\mathcal{X}^2(S)$	1051.14	2699.62	**3681.89**	2549.08	1518.63	253.45	52.34	566.54	2130.31	474.70	1283.73	567.17	2446.47	1983.41
	$\psi(S)$	13.91	99.94	18.32	30.80	11.92	2.79	0.51	20.09	75.29	23.77	57.93	27.33	86.01	**298.44**
	ρ_T	8.86	**135.20**	67.53	33.33	26.18	6.38	4.15	22.76	58.88	29.33	27.19	24.03	101.46	32.98
	$d_a(S)$	1.57	0.74	0.27	0.92	0.46	0.44	0.12	0.88	1.28	0.81	2.13	1.14	0.85	**9.05**

Next, for all digits d and temporal subgraphs with temporal average degree $\frac{2h(S)}{|S||T_S|} = \Omega(\log n)$, we use the double union bound and get:

$$\Pr\left[\exists d \in \{1,\cdots,9\}, S \subseteq V : \frac{X_{S,d}}{|T_S|} \notin (1+\epsilon)\mathbb{E}\left[\frac{X_{S,d}}{|T_S|}\right]\right]$$
$$\leq \sum_{d=1}^{9}\sum_{k=2}^{n}\binom{n}{k}n^{-6k} \leq 9\sum_{k=2}^{n}(\frac{en}{k})^k n^{-6k} = o(1)\,.$$

Consequently, with high probability $1-o(1)$, the number of temporal edges that begin with digit d in $G[S]$ with a sufficiently large average temporal degree is strongly concentrated around the expectation $\mathbb{E}\left[\frac{X_{S,d}}{|T_S|}\right]$. □

B.2 Proof of Corollary 3.2

Proof. Theorem 3.1 indicates that for all sufficiently large temporal subgraphs, we expect $\frac{X_{S,d}}{|T_S|}$ to be closely concentrated around its true expectation. With high probability, the following inequality holds:

$$\psi(S) = \frac{1}{|S|}\sum_{d=1}^{9}\frac{\left(\frac{X_{S,d}}{|T_S|}-\frac{p_d h(S)}{|T_S|}\right)^2}{\frac{p_d h(S)}{|T_S|}} \leq \frac{1}{|S|}\sum_{d=1}^{9}\epsilon^2\frac{p_d h(S)}{|T_S|} = \frac{\epsilon^2}{2}\frac{2h(S)}{|S||T_S|}.$$

Thus, $\psi(S) \leq \frac{h(S)}{|S||T_S|}$ since $0 < \epsilon < 1$. Hence, we have $\max_{S\subseteq V}\psi(S) \leq \rho_T^\star$. □

B.3 Proof of Theorems 3.3

Proof. Corollary 3.2 indicates that under the null hypothesis H_0, which corresponds to a temporal *normal* subgraph, the inequality $\psi(S) \leq \frac{h(S)}{|S||T_S|}$ holds. Therefore, for temporal *anomalous* subgraph, i.e., temporal AntiBenford subgraph, we have $\psi(S) \gg \frac{h(S)}{|S||T_S|}$. □

B.4 Lemma 1

Lemma 1. Optimizing the loss function in Equation 3 of the paper is equivalent to maximizing $I(f_e(\mathcal{G}[S_1]); f_e(\mathcal{G}[S_2]))$, leading to the maximization of $I(f_e(\mathcal{G}[S_1]); \mathcal{G}[S_2])$, where I represents mutual information.

Proof. Since the objective of the encoder is to maximize the embedding similarity between positive subgraph pairs and minimize the embedding similarity between negative pairs, the loss function for a pair of subgraphs $G[S_1]$ and $G[S_2]$ with embeddings $e(G[S_1])$ and $e(G[S_2])$ is equivalent to:

$$\mathcal{L}_{1,2} = -\log\frac{\exp(-D(G[S_1], G[S_2]))}{\sum_{i,j}\exp(-D(G[S_i], G[S_j]))}\,. \tag{7}$$

And minimizing Equation 7 above is equivalent to maximizing a lower bound of the mutual information between the learned embeddings of two subgraphs as proven in [51], which is equivalent to maximizing the mutual information between their embeddings $I(f_e(\mathcal{G}[S_1]); f_e(\mathcal{G}[S_2]))$. According to Lemma A.5 in [45], it leads to the maximization of $I(f_e(\mathcal{G}[S_1]); \mathcal{G}[S_2])$. □

B.5 Proof of Theorems 4.4

Proof. The encoder f_1 can differentiate financial subgraphs with different financial distributions that f_2 cannot. It means that f_1 can capture more information of subgraphs $\mathcal{G}[S_1]$ than f_2:

$$H(\mathcal{G}[S_1]) \geq H(f_1(\mathcal{G}[S_1])) > H(f_2(\mathcal{G}[S_1]))\,, \tag{8}$$

where $H(\cdot)$ represents information entropy. Since $f_1(\mathcal{G}[S_1])$ and $f_2(\mathcal{G}[S_1])$ are functions of $\mathcal{G}[S_1]$, we have

$$I(f_1(\mathcal{G}[S_1]); \mathcal{G}[S_1]) > I(f_2(\mathcal{G}[S_1]); \mathcal{G}[S_1])\,. \tag{9}$$

According to Theorem A.6 in [45] and Lemma 1 above, it holds that $I(f_1(\mathcal{G}[S_1]); y) > I(f_2(\mathcal{G}[S_1]); y)$. □

B.6 Proof of Theorems 4.5

Proof. The temporal subgraph encoder comprises two main components: the temporal GNN module and the subgraph pair optimization phase. The time complexity of the temporal GNN module is $O(deg_{\text{avg}}\cdot|V|)$, where deg_{avg} denotes the average degree of nodes. For the subgraph pair optimization phase, the time complexity is $O(k_1(|U|+|Z|))$, where $|U|$ and $|Z|$ represent the number of positive and negative pairs, respectively, and k_1 is a constant. Hence, the overall time complexity of the temporal subgraph encoder is $O(deg_{\text{avg}}\cdot|V| + k_1(|U|+|Z|))$. □

Table 6: Effectiveness on Blur dataset. For each method, we report average results over five runs.

top-a	Metrics	Holoscope	FlowScope	AntiBenford	CLARE	AS-GAE	GCAD	SIGNET	FAST-GA	TopLC	OTCD	DeepSphere	RustGraph	AnoGraph	TempASD
5	$\mathcal{X}^2(S)$	562.79	168.43	640.74	**4575.33**	591.73	193.67	116.91	808.49	406.99	592.01	1693.09	1521.74	1280.05	4523.17
	$\psi(S)$	1.45	0.78	2.71	23.76	1.09	2.07	1.17	77.23	54.46	35.63	21.55	137.78	100.90	**852.31**
	ρ_T	1.32	0.77	1.17	16.57	1.73	1.49	1.89	53.97	80.78	48.63	20.47	51.53	105.29	**107.91**
	$d_a(S)$	1.10	1.01	2.31	1.43	0.63	1.38	0.62	1.43	0.67	0.73	1.05	2.67	0.96	**7.90**
10	$\mathcal{X}^2(S)$	192.32	121.93	343.40	2754.44	415.31	190.90	99.12	404.51	275.23	333.08	1086.23	825.58	663.49	**3004.62**
	$\psi(S)$	0.90	0.72	1.94	15.13	0.63	1.47	0.69	52.32	41.58	23.37	14.78	83.41	51.71	**671.56**
	ρ_T	1.54	0.55	0.96	10.10	2.63	2.05	1.74	46.77	55.40	49.65	**102.48**	44.92	54.51	89.19
	$d_a(S)$	0.58	1.30	2.02	1.50	0.24	0.72	0.40	1.12	0.75	0.47	0.14	1.86	0.95	**7.53**
15	$\mathcal{X}^2(S)$	119.74	87.78	259.30	**2000.33**	352.49	185.26	93.83	411.00	246.75	187.86	744.70	590.93	450.49	1789.21
	$\psi(S)$	0.61	0.30	1.59	11.35	0.51	0.70	0.88	35.27	37.08	20.06	11.79	66.22	35.02	**476.43**
	ρ_T	1.20	0.72	0.85	7.63	2.22	1.55	1.36	39.33	41.60	39.78	**73.80**	39.84	38.05	66.33
	$d_a(S)$	0.51	0.41	1.86	1.49	0.23	0.45	0.65	0.90	0.89	0.50	0.16	1.66	0.92	**7.18**
20	$\mathcal{X}^2(S)$	99.71	54.27	223.07	**1538.24**	287.88	191.28	81.82	201.60	97.57	138.23	651.28	514.28	390.08	962.36
	$\psi(S)$	0.59	0.32	1.39	9.02	0.37	0.82	0.46	27.08	21.09	22.44	9.70	58.68	26.58	**210.13**
	ρ_T	1.13	0.43	0.80	5.98	2.92	0.89	1.52	34.72	22.97	31.88	**55.60**	24.03	28.80	35.46
	$d_a(S)$	0.52	0.76	1.74	1.51	0.13	0.92	0.30	0.78	0.92	0.70	0.17	2.44	0.92	**5.93**

B.7 Proof of Theorems 4.6

Proof. The temporal candidate detection process consists of two main steps: temporal dense subgraph construction and vector space matching. The time complexity of temporal dense subgraph construction is $O(|V_{W_r}| \cdot |\mathcal{E}_{W_r}|)$, where $|V_{W_r}|$ is the number of vertices during the observation window W_r, and $|\mathcal{E}_{W_r}|$ represents the number of edges during W_r. For vector space matching, the time complexity is $O(d_p \cdot p)$, where d_p denotes the dimension of the feature vector, and p is the number of candidates. Therefore, the overall time complexity of temporal candidate detection is $O(|V_{W_r}| \cdot |\mathcal{E}_{W_r}| + d_p \cdot p)$. □

B.8 Proof of Theorems 4.7

Proof. The refinement module optimizes trainable hyperparameters by maximizing the rewards associated with training anomalous subgraphs, $\mathcal{G}[S_{a_t}]$, where each subgraph corresponds to t states in the optimization process. Consequently, the overall time complexity of the refinement process is $O(|\mathcal{G}[S_{a_t}]| \cdot t)$, where t denotes the number of states in the optimization process, and $|\mathcal{G}[S_{a_t}]|$ represents the number of training anomalous subgraphs. □

C SUPPLEMENTARY EXPERIMENTS

Experimental Environment. Our experiments are conducted on the server running Intel Xeon Platinum 8370C processor 32-core 2.8GHz CPU with NVIDIA GeForce RTX 4090 24G GPU.

C.1 Dataset Description

• The PlusTokenPonzi dataset [3] is a real Ethereum blockchain transaction dataset that involves money laundering activities from EthereumHeist [43]. It includes detailed information such as transaction timestamps and amounts, service provider address labels, hierarchical levels of laundering addresses, etc. This is the first publicly available dataset with ground truth data on money laundering.

[3]https://github.com/lindan113/EthereumHeist?tab=readme-ov-file

• The ETH datasets consist of real transaction data obtained from the Ethereum blockchain, available through Google BigQuery[4]. These datasets include token transactions between source and sink addresses, including transaction amounts and timestamps. They are also used in the advanced method AntiBenford [6].

• The Blur dataset is a real transaction dataset from the NFT marketplace [58], compiled using the Etherscan API[5]. It covers NFT transactions among addresses from October 19, 2022, to April 1, 2023.

C.2 Effectiveness Evaluation

Tables 5 and 6 present the effectiveness results for the ETH-Jan-2019 and Blur datasets. They show that our proposed TempASD achieves the highest abnormal degree values.

C.3 Sensitivity Analysis

We evaluate the effects of the TempASD model w.r.t. different hyperparameters. Since the trends of hyperparameters on PlusToken-Ponzi, ETH-Jan-19 and Blur datasets are similar to ETH-Jan-18, we use ETH-Jan-18 to show the effects of hyperparameters. Similarly, since the trends of $\mathcal{X}^2$, $\psi(S)$ and ρ_T are similar to $d_a(S)$, we show the results of $d_a(S)$ in Figure 9.

Duration of Time Interval for Snapshots. Figure 9 (a) indicates that increasing the time interval results in the best performance at an interval of 10 minutes. However, further increases cause performance to decline due to the loss of more detailed transaction information.

Maximum Size m of Nodes in the Boundary. Figure 9 (b) demonstrates that when m exceeds 150, it introduces more noise, leading to degraded results.

Range of Observation Window W_r for Temporal Dense Subgraphs. Figure 9 (c) illustrates that expanding W_r beyond 60 results in larger temporal subgraphs with lower abnormal degrees, which negatively impacts performance.

Size of the Top-p Candidates. Figure 9 (d) reveals that increasing the number of candidates beyond 150 results in more sub-optimal subgraphs, thereby reducing performance.

Number of Sampling Pairs. Figure 9 (e) shows that increasing the number of sampling pairs beyond 10 leads to degraded performance due to potentially increased noise.

[4]https://www.kaggle.com/bigquery/ethereum-blockchain
[5]https://etherscan.io/

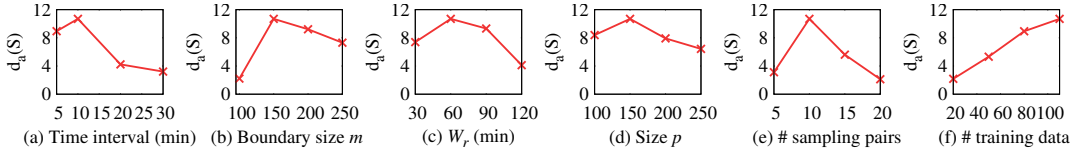

Figure 9: Hyperparameter effects of TempASD on ETH-Jan-18 dataset.

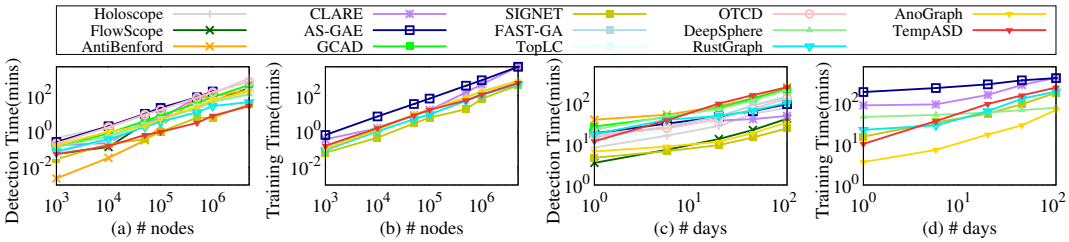

Figure 10: Scalability on (a–b) the size of graph nodes; (c–d) the span of time dimension.

Number of Training subgraphs. Figure 9 (f) shows that increasing the number of training subgraphs improves results up to 80. However, the improvement diminishes when the number reaches 100. Due to the scarcity of training subgraphs, we did not further increase this number in our experiment.

C.4 Efficiency Comparison

Figure 11 illustrates the detection and training times for the PlusTokenPonzi dataset. It can be observed that our proposed TempASD achieves the shortest detection time, while also maintaining relatively lower training time.

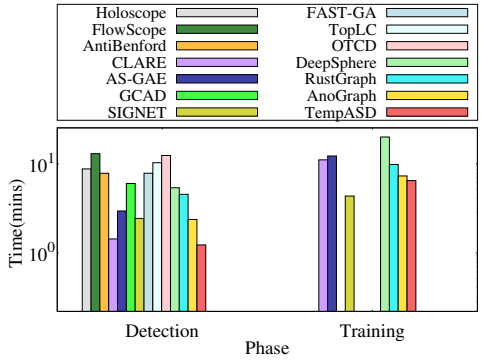

Figure 11: Detection time and training time on the PlusTokenPonzi dataset. Note that non-learning-based methods require zero training time.

C.5 Scalability Study

Five datasets ranging from 1 week to 1 year in 2019 are utilized to analyze the scalability over time in Section C.5.2. Further details can be found in Table 7.

Table 7: Statistics of experimented datasets.

Datasets	# Nodes	# Edges	# Transactions
ETH-1 Week	779,237	1,016,481	1,438,641
ETH-1 Month	2,199,347	3,331,594	6,128,061
ETH-3 Month	2,877,055	8,504,999	17,529,925
ETH-6 Month	5,018,047	21,572,152	43,885,527
ETH-1 Year	6,820,719	38,917,136	85,055,054

C.5.1 Scalability w.r.t. the size of graph nodes. Figures 10 (a–b) display the detection and training times in relation to the size of graph nodes for all methods. Our proposed TempASD exhibits an approximately linear relationship with the size of nodes in financial networks on a log-log scale, indicating that its time complexity is polynomial.

C.5.2 Scalability w.r.t. the span of time dimension. Figures 10 (c–d) display detection and training times w.r.t. the number of days. The node size is fixed at 500,000 to ensure that the changes are solely due to variations in the time dimension. The trend for TempASD is approximately linear in relation to the number of days in dynamic financial networks.