Fast and Accurate Anomaly Detection in Dynamic Graphs with a Two-Pronged Approach

Minji Yoon†, Bryan Hooi‡, Kijung Shin§, Christos Faloutsos∗
∗ School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA
† School of Computing, National University of Singapore, Singapore
‡ School of Electrical Engineering, KAIST, South Korea
minjiy@cs.cmu.edu, bhooi@andrew.cmu.edu, kijungs@kaist.ac.kr, christos@cs.cmu.edu

1 INTRODUCTION

Network-based systems including computer networks and social network services have been a focus of various attacks. In computer networks, distributed denial of service (DDOS) attacks use a number of machines to make connections to a target machine to block their availability. In social networks, users pay spammers to “Like” or “Follow” their page to manipulate their public trust. By abstracting those networks to a graph, we can detect those attacks by finding suddenly emerging anomalous signs in the graph.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Conference, August 4–8, 2019, Anchorage, AK, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6201-6/19/08...
https://doi.org/10.1145/3292500.3330946

Various approaches have been proposed to detect anomalies in graphs, the majority of which focus on static graphs [3, 6, 11–13, 20, 24]. However, many real-world graphs are dynamic, with timestamps indicating the time when each edge was inserted/deleted. Static anomaly detection methods, which are solely based on static connections, miss useful temporal signals of anomalies.

Several approaches [8, 9, 22] have been proposed to detect anomalies on dynamic graphs (we review these in greater detail in Section 2 and Table 1). However, they are not satisfactory in terms of accuracy and speed. Accurately detecting anomalies in near real-time is important in order to cut back the impact of malicious activities and start recovery processes in a timely manner.

In this paper, we propose AnomRank, a fast and accurate online algorithm for detecting anomalies in dynamic graphs with a two-pronged approach. We classify anomalies in dynamic graphs into two types: AnomalyS and AnomalyW. AnomalyS denotes suspicious changes to the structure of the graph, such as through the addition of edges between previously unrelated nodes in spam attacks. AnomalyW indicates anomalous changes in the weight (i.e., number of edges) between connected nodes, such as suspiciously frequent connections in port scan attacks.

Various node score functions have been proposed to map each node of a graph to an importance score: PageRank [17], HITS and its derivatives (SALSA) [13, 15], Fiedler vector [7], etc. Our intuition is that anomalies induce sudden changes in node scores. Based on this intuition, AnomRank focuses on the 1st and 2nd order derivatives of node scores to detect anomalies with large changes in node scores within a short time. To detect AnomalyS and AnomalyW effectively, we design two versions of node score functions based on characteristics of these two types of anomalies. Then, we define two novel metrics, AnomRankS and AnomRankW, which measure the 1st and 2nd order derivatives of our two versions of node score functions, as a barometer of anomalousness. We theoretically analyze the effectiveness of each metric on the corresponding anomaly type, and provide rigid guarantees on the accuracy of AnomRank. Through extensive experiments with real-world and synthetic graphs, we demonstrate the superior performance of AnomRank over existing methods. Our main contributions are:

- **Online, two-pronged approach**: We introduce AnomRank, an online detection method, for two types of anomalies (AnomalyS, AnomalyW) in dynamic graphs.
- **Theoretical guarantees**: We prove the effectiveness of our proposed metrics, AnomRankS and AnomRankW, theoretically (Theorems 1 and 2).
We discuss previous work on detecting anomalous entities (nodes, edges, events, etc.) on static and dynamic graphs. See [4] for an extensive survey on graph-based anomaly detection.

Anomalous Node Detection can be described under the following categories:

- **Anomalous Node Detection**: [3] extracts egonet-based features and finds empirical patterns with respect to the features. Then, it identifies nodes whose egonets deviate from the patterns. [27] groups nodes that share many neighbors and spots nodes that cannot be assigned to any community.

- **Anomalous Edge Detection**: [6] encodes the input graph based on similar connectivity between nodes, then spots edges whose removal significantly reduces the total encoding cost. [24] factorizes the adjacency matrix and flags edges which introduce high reconstruction error as outliers.

- **Anomalous Subgraph Detection**: [11] and [20] measure the anomalousness of nodes and edges, then find a dense subgraph consisting of many anomalous nodes and edges.

Anomaly detection in dynamic graphs can also be described under the following categories:

- **Anomalous Node Detection**: [23] approximates the adjacency matrix of the current snapshot based on incremental matrix factorization. Then, it spots nodes corresponding to rows with high reconstruction error. [25] computes nodes features (degree, closeness centrality, etc) in each graph snapshot. Then, it identifies nodes whose features are notably different from their previous values and the features of nodes in the same community.

- **Anomalous Edge Detection**: [8] detects edges that connect sparsely-connected parts of a graph. [18] spots anomaly edges based on their occurrence, preferential attachment and mutual neighbors.

- **Anomalous Subgraph Detection**: [5] spots near-bipartite cores where each node is connected to others in the same core densely within a short time. [12] and [21] detect groups of nodes who form dense subgraphs in a temporally synchronized manner. [22] identifies dense subtensors created within a short time.

- **Event Detection**: [1, 9, 10, 14] detect the following events: sudden appearance of many unexpected edges [1], sudden appearance of a dense subgraph [9], sudden drop in the similarity between two consecutive snapshots [14], and sudden prolonged spikes and lightweight stars [10].

Our proposed AnomRank is an anomalous event detection method with fast speed and high accuracy. It can be easily extended to localize culprits of anomalies into nodes and substructures (Section 4.4), and it detects various types of anomalies in dynamic graphs in a real-time. Table 1 compares AnomRank to existing methods.

### Table 1: AnomRank out-features competitors: comparison of our proposed AnomRank and existing methods for anomaly detection in dynamic graphs.

| Property                        | Oddball [3] | MetricFDR [10] | CC, CS [5, 12] | DenseAlert [22] | SpotLight [9] | SedanSpot [8] | AnomRank |
|---------------------------------|-------------|----------------|----------------|-----------------|----------------|----------------|-----------|
| Real-time detection*            | ✔           | ✔              | ✔              | ✔               | ✔              | ✔              | ✔         |
| Allow edge deletions            | ✔           | ✔              | ✔              | ✔               | ✔              | ✔              | ✔         |
| Structural anomalies            | ✔           | ✔              | ✔              | ✔               | ✔              | ✔              | ✔         |
| Edge weight anomalies           | ✔           | ✔              | ✔              | ✔               | ✔              | ✔              | ✔         |

*compute 1M edges within 5 seconds.

### 3 Preliminaries

Table 2 gives a list of symbols and definitions.

Various node score functions have been designed to estimate importance (centrality, etc.) of nodes in a graph: PageRank [17]; HITS [13] and its derivatives (SALSA) [15]; Fiedler vector [7]; all the centrality measures from social network analysis (eigenvector-, degree-, betweenness-centrality [26]). Among them, we extend PageRank to design our node score functions in Section 4.2 because (a) it is fast to compute, (b) it led to the ultra-successful ranking.
Thus, as the graph evolves under normal behavior with the insertion and deletion of edges, node scores evolve smoothly. In contrast, anomalies such as network attacks or rating manipulation often lead to large changes in node scores. Our key intuition is that such abrupt gains or losses are reflected in the 1st and 2nd derivatives of node scores: large 1st derivative identifies large changes, while large 2nd derivative identifies abrupt changes in the trend of the data, thereby distinguishing changes from normal users who evolve according to smooth trends. Thus, tracking 1st and 2nd order derivatives helps detect anomalies in dynamic graphs.

Changes in a dynamic graph are classified into two types: structure changes and edge weight changes. Since these two types of changes affect node scores of the graph differently (more details in Section 4.2), we need to handle them separately. Thus, first, we classify anomalies in dynamic graphs into two types: ANOMALYS and ANOMALYW (Section 4.1). Then we design two node score functions based on characteristics of these two types of anomalies, respectively (Section 4.2). Next, we define two novel metrics for anomalousness using 1st and 2nd order derivatives of our node scores, and verify the effectiveness of each metric on the respective type of anomalies theoretically (Section 4.3). Based on these analyses, we introduce our method ANOMRANK, a fast and accurate anomaly detection algorithm in dynamic graphs (Section 4.4).

### 4.1 Anomalies in Dynamic Graphs

We classify anomalies in dynamic graphs into two types: ANOMALYS and ANOMALYW.

#### 4.1.1 ANOMALYS

It is suspicious if a number of edges are inserted/deleted among nodes which are previously unrelated/related. Hence, ANOMALYS denotes a massive change with regard to the graph structure. One example of ANOMALYS is spam in mail network graphs: a spammer sends mail to many unknown individuals, generating out-edges toward previously unrelated nodes. Data exfiltration attacks in computer network graphs are another example: attackers transfer a target’s data stealthily, generating unseen edges around a target machine to steal information. As illustrated in Figure 2, we define a structure change as follows:

**Definition 1 (Structure Change).** If a node $u$ changes the destination of $\Delta m$ of its out-edges from previous neighbors $v_1, \ldots, v_{\Delta m}$ to new neighbors $v'_1, \ldots, v'_{\Delta m}$, we call the change a structure change of size $\Delta m$.

With abnormally large $\Delta m$, a structure change becomes an ANOMALYS. To detect ANOMALYS, we need to focus on the existence of edges between two nodes, rather than the number of occurrences of edges between two nodes.

---

**Table 2: Table of symbols.**

| Symbol | Definition |
|--------|------------|
| $G$    | (undirected and (un)weighted input graph) |
| $\Delta G$ | update in graph |
| $n, m$ | numbers of nodes and edges in $G$ |
| $\tilde{A}$ | $(n \times n)$ row-normalized adjacency matrix of $G$ |
| $\tilde{B}$ | $(n \times n)$ row-normalized adjacency matrix of $G + \Delta G$ |
| $\Delta A$ | $(n \times n)$ difference between $\tilde{A}^T$ and $\tilde{B}^T (= \tilde{B}^T - \tilde{A}^T)$ |
| $c$    | damping factor of PageRank |
| $b_s$  | $(n \times 1)$ uniform starting vector |
| $b_w$  | $(n \times 1)$ out-edge proportional starting vector |
| $\tilde{A}_u$ | $(n \times n)$ row-normalized unweighted adjacency matrix |
| $\tilde{A}_w$ | $(n \times n)$ row-normalized weighted adjacency matrix |

**Figure 2:** Two-pronged approach: Changes in dynamic graphs are classified into two types, structure change and edge weight change.
4.1.2 AnomalyW. In dynamic graphs, an edge between two nodes could occur several times. Edge weight is proportional to the number of edge occurrences. AnomalyW denotes a massive change of edge weights in a graph. One example of AnomalyW is port scan attacks in computer network graphs: to scan ports in a target IP address, attackers repeatedly connect to the IP address, thus increasing the number of edge occurrences to the target node. On Twitter, high edge density on a user-keyword graph could indicate bot-like behavior, e.g., bots posting about the same content repeatedly. As illustrated in Figure 2, we define an edge weight change as follows:

**Definition 2 (Edge Weight Change).** If a node adds/subtracts Δm out-edges to neighbor node v, we call the change an edge weight change of size Δm.

With abnormally large Δm, an edge weight change becomes an AnomalyW. In contrast to AnomalyS, here we focus on the number of occurrences of each edge, rather than only the presence or absence of an edge.

4.2 Node Score Functions for Detecting AnomalyS and AnomalyW

To detect AnomalyS and AnomalyW, we first define two node score functions, ScoreS and ScoreW, which we use to define our anomalousness metrics in Section 4.3.

4.2.1 ScoreS. We introduce node score ScoreS, which we use to catch AnomalyS. Define the row-normalized unweighted adjacency matrix ˜A, a starting vector b, which is an all-1/n vector of length n (the number of nodes), and the damping factor c.

**Definition 3 (ScoreS).** ScoreS node score vector p is defined by the following iterative equation:

\[ p = c \tilde{A} b + (1 - c) b \]

For this (unweighted) case, ScoreS is the same as PageRank, but we refer to it as ScoreS for consistency with our later definitions. Note that the number of edge occurrences between nodes is not considered in ScoreS. Using Lemma 1, we can compute ScoreS incrementally at fast speed, in dynamic graphs.

4.2.2 ScoreW. Next, we introduce the second node score, ScoreW, which we use to catch AnomalyW. To incorporate edge weight, we use the weighted adjacency matrix A instead of ˜A. However, this is not enough on its own: imagine an attacker node who adds a massive number of edges, all toward a single target node, and the attacker has no other neighbors. Since A is row-normalized, this attacker appears no different in A as if they only added a single edge toward the same target. Hence, to catch such attackers, we also introduce an out-degree proportional starting vector b, i.e., setting the initial scores of each node proportional to its outdegree.

**Definition 4 (ScoreW).** ScoreW node score vector p is defined by the following iterative equation:

\[ p = c A p + (1 - c) b \]

A(i, j) is the edge weight from node i to node j. bw(i) is \( \frac{m_i}{m} \), where mi denotes the total weight of all edges of node i, and m denotes the total weight of the graph.

Next, we show how ScoreW is computed incrementally in a dynamic graph. Assume that a change ΔG happens in graph G in time interval Δt, inducing changes ΔAw and Δbw in the adjacency matrix and the starting vector, respectively.

**Lemma 2 (Dynamic ScoreW).** Given updates ΔAw and Δbw in a graph during Δt, an updated score vector \( p_w(t + Δt) \) is computed incrementally from a previous score vector \( p_w(t) \) as follows:

\[
p_w(t + Δt) = p_w(t) + \sum_{k=0}^{∞} (c(ΔAw + Δbw))^k cΔAw p_w(t) + (1 - c) \sum_{k=0}^{∞} (c(ΔAw + Δbw))^k Δbw
\]

**Proof.** For brevity, \( p_w^0 = p_w(t + Δt) \) and \( p_w^0 = p_w(t) \).

\[
p_w^0 = (1 - c) \sum_{k=0}^{∞} c^k (ΔAw + Δbw)^k b + (1 - c) \sum_{k=0}^{∞} c^k (ΔAw + Δbw)^k b
\]

\[
= p_w^0 + \sum_{k=0}^{∞} c(ΔAw + Δbw)^k cΔAw p_w^0 + (1 - c) \sum_{k=0}^{∞} c(ΔAw + Δbw)^k Δbw
\]

In the third line, we use Lemma 1.

Note that, for small changes in a graph, the starting vectors of the last two terms, \( cΔAw p_w(t) \) and \( Δbw \), have much smaller L1 lengths than the original starting vector b, so they can be computed at fast speed.

4.2.3 Suitability. We estimate changes in ScoreS induced by a structure change (Definition 1) and compare the changes with those in ScoreW to prove the suitability of ScoreS for detecting AnomalyS.

**Lemma 3 (Upper Bound for Structure Change in ScoreS).** When a structure change of size Δm happens around a node u with k out-neighbors, \|ΔA\|1 is upper-bounded by \( \frac{2Δm}{k} \).

**Proof.** In ΔAu, only the u-th column has nonzeros. Thus, \( \|ΔA\|1 = \|ΔAu\|1 \), ΔAu is normalized by k as the total number of out-neighbors of node u is k. For out-neighbors v1, v2, ..., vk who lose edges, ΔAvk(vj, u) = -1/k. For out-neighbors v′1, v′2, ..., v′k, who earn edges, ΔAv′k(v′j, u) = 1/k. Then \( \|ΔA\|1 = \|ΔAu\|1 = \|ΔA_v\|1 = \frac{Δm}{k} + \frac{Δm}{k} = \frac{2Δm}{k} \).

When a structure change is presented in ScoreW, \( \|Δbw\|1 = 0 \) since there is no change in the number of edges. Moreover, \( \|ΔAw\|1 = \frac{2Δm}{m} \) since each row in Aw is normalized by the total sum of out-edge weights, mw, which is larger than the total number of out-neighbors k. In other words, a structure change generates larger changes in ScoreS \( \frac{2Δm}{m} \) than ScoreW \( \frac{2Δm}{mw} \). Thus ScoreS is more suitable to detect AnomalyS than ScoreW.

Similarly, we estimate changes in ScoreW induced by an edge weight change (Definition 2) and compare the changes with those in ScoreS to prove the suitability of ScoreW for detecting AnomalyW.

**Lemma 4 (Upper Bound for Edge Weight Change in ScoreW).** When an edge weight change of size Δm happens around a node u with mw out-edge weights in a graph with m total edge weights, \( \|ΔAw\|1 \) and \( \|Δbw\|1 \) are upper bounded by \( \frac{2Δm}{mw} \) and \( \frac{2Δm}{m} \), respectively.
Proof. In $\Delta A_w$, only the $u$-th column has nonzeros. Then $\|\Delta A_w\|_1 = \|\Delta A_w(u)\|_1$. Node $u$ has $m_u$ edges toward each out-neighbor $v_i (i = 1, \ldots, k)$. Thus the total sum of out-edge weights, $m_u$, is $\sum_{i=1}^{k} m_u$. Since an weighted adjacency matrix is normalized by the total out-edge weights, $\tilde{A}_w(v_i, u) = \frac{m_u}{m_u + \Delta m}$ After $\Delta m$ edges are added from node $v_u$ to node $v_i$, $\Delta A_w(v_i, u) = \frac{m_u}{m_u + \Delta m} - \frac{m_u}{m_u}$ for $i \neq k$, $\Delta A_w(v_i, u) = \frac{m_u}{m_u + \Delta m}$ for $i = k$. Then $\|\Delta A_w\|_1 = \|\Delta A_w(u)\|_1$ is bounded as follows:

$$\|\Delta A_w\|_1 = \|\Delta A_w(u)\|_1 = \sum_{i=1}^{k} m_u \left( \frac{1}{m_u} - \frac{1}{m_u + \Delta m} \right) + \frac{\Delta m}{m_u + \Delta m}$$

$$= \frac{2 \Delta m}{m_u + \Delta m} \leq \frac{2 \Delta m}{\Delta m}$$

$b_w(i) = m_u$ where $m_u$ is the total number of out-edges of node $v_i$. After $\Delta m$ edges are added from node $v_u$ to node $v_i$, $\Delta b_w(i) = \frac{m_u}{m_u + \Delta m} - \frac{m_u}{m_u}$ for $i \neq u$, $\Delta b_w(i) = \frac{m_u}{m_u + \Delta m} - \frac{m_u}{m_u}$ for $i = u$. Then $\|\Delta b_w\|_1$ is bounded as follows:

$$\|\Delta b_w\|_1 = \sum_{i=1}^{k} m_u \left( \frac{1}{m_u} - \frac{1}{m_u + \Delta m} \right) + \frac{\Delta m}{m_u + \Delta m} = \frac{2 \Delta m}{m_u + \Delta m} \leq \frac{2 \Delta m}{m_u}$$

In contrast, when an edge weight change is presented in $\text{SCORES}_S$, $\|\Delta A_w\|_1 = 0$ since the number of out-neighbors is unchanged. Note that $\|\Delta b_w\|_1 = 0$ since $b_w$ is fixed in $\text{SCORES}_S$. In other words, $\text{ANOmalYW}$ does not induce any change in $\text{SCORES}_S$.

4.3 Metrics for \text{ANOmalYYS} and \text{ANOmalYW}

Next, we define our two novel metrics for evaluating the anomalousness at each time in dynamic graphs.

4.3.1 \text{ANOmalRANKS}. First, we discretize the first order derivative of $\text{SCORES}_S$ vector $p_s$ as follows:

$$p'_s = \frac{p_s(t + \Delta t) - p_s(t)}{\Delta t}$$

Similarly, the second order derivative of $p_s$ is discretized as follows:

$$p''_s = \frac{(p_s(t + \Delta t) - p_s(t)) - (p_s(t) - p_s(t - \Delta t))}{\Delta t^2}$$

Next, we define a novel metric $\text{ANOmalRANKS}$ which is designed to detect $\text{ANOmalYYS}$ effectively.

Definition 5 (ANOmalRANKS). Given $\text{SCORES}_S$ vector $p_s$, \text{ANOmalRANKS} $a_s$ is an $(n \times 2)$ matrix $[p'_s, p''_s]$, concatenating 1st and 2nd derivatives of $p_s$. The $\text{ANOmalRANKS}$ score is $[a_s]_1$.

Next, we study how ANOmalRANKS scores change under the assumption of a normal stream, or an anomaly, thus explaining how it distinguishes anomalies from normal behavior. First, we model a normal graph stream based on Lipschitz continuity to capture smoothness:

Assumption 1 ($[p_s(t)]_1$ in Normal Stream). In a normal graph stream, $[p_s(t)]_1$ is Lipschitz continuous with positive real constants $K_1$ and $K_2$ such that:

$$[p'_s]_1 \leq K_1 \text{ and } [p''_s]_1 \leq K_2$$

In Lemma 5, we back up Assumption 1 by upper-bounding $[p'_s]_1$ and $[p''_s]_1$. For brevity, all proofs of this subsection are given in Supplement A.3.

Lemma 5 (Upper bound of $[p'_s]_1$). Given damping factor $c$ and updates $\Delta A_s$ in the adjacency matrix during $\Delta t$, $[p'_s]_1$ is upper-bounded by $\frac{c}{1 - c} \frac{\|\Delta A_s\|_1}{\Delta t}$.

Proof. Proofs are given in Supplement A.3.

We bound the $L1$ length of $p'_s$ in terms of $L1$ length of $\Delta A_s$, and $\Delta A_n$, where $\Delta A_s$ denotes the changes in $A_s$ from time $(t - \Delta t)$ to time $t$, and $\Delta A_n$ denotes the changes in $A_n$ from time $t$ to $(t + \Delta t)$.

Lemma 6 (Upper bound of $[p''_s]_1$). Given damping factor $c$ and sequential updates $\Delta A_s$ and $\Delta A_n$, $[p''_s]_1$ is upper-bounded by $\frac{c}{1 - c} \frac{\|\Delta A_s\|_1}{\Delta t} + \frac{c}{1 - c} \frac{\|\Delta A_n\|_1}{\Delta t}$.

Proof. Proofs are given in Supplement A.3.

Normal graphs have small changes thus having small $\|\Delta A_s\|_1$. This results in small values of $[p'_s]_1$. In addition, normal graphs change gradually thus having small $\|\Delta A_n\|_1$. This leads to small values of $[p''_s]_1$. Then, $\text{ANOmalRANKS}$ score $[a_s]_1 = \max([p'_s]_1, [p''_s]_1)$ has small values in normal graph streams under small upper bounds.

Observation 1 (ANOmalYYS in ANOmalRANKS). ANOmalYYS involves sudden structure changes, inducing large ANOmalRANKS scores.

ANOmalYYS happens with massive changes ($\frac{\Delta m}{\Delta t}$) abruptly ($\frac{\Delta m}{\Delta t}$). In the following Theorem 1, we explain Observation 1 based on large values of $\frac{\Delta m}{\Delta t}$ and $\frac{\Delta m}{\Delta t}$ in ANOmalYYS.

Theorem 1 (Upper bounds of $[p'_s]_1$ and $[p''_s]_1$ with ANOmalYYS). When ANOmalYYS occurs with large $\frac{\Delta m}{\Delta t}$ and $\frac{\Delta m}{\Delta t}$, $L1$ lengths of $p'_s$ and $p''_s$ are upper-bounded as follows:

$$[p'_s]_1 \leq \frac{c}{1 - c} \frac{2 \Delta m}{\Delta t}$$

$$[p''_s]_1 \leq \frac{c}{1 - c} \frac{2 \Delta m^2}{\Delta t} + 2 \left( \frac{c}{1 - c} \right)^2 \frac{\Delta m}{\Delta t}$$

Proof. Proofs are given in Supplement A.3.

Based on Theorem 1, ANOmalYYS has higher upper bounds of $[p'_s]_1$ and $[p''_s]_1$ than normal streams. This gives an intuition for why ANOmalYYS results in high ANOmalRANKS scores (Figure 1(c)). We detect ANOmalYYS successfully based on ANOmalRANKS scores in real-world graphs (Figure 3).

4.3.2 ANOmalRANKW. We discretize the first and second order derivatives $p'_w$ and $p''_w$ of $p_w$ as follows:

$$p'_w = \frac{p_w(t + \Delta t) - p_w(t)}{\Delta t}$$

$$p''_w = \frac{(p_w(t + \Delta t) - p_w(t)) - (p_w(t) - p_w(t - \Delta t))}{\Delta t^2}$$

Then we define the second metric ANOmalRANKW which is designed to find ANOmalYYS effectively.

Definition 6 (ANOmalRANKW). Given $\text{SCOREW}$ vector $p_w$, ANOmalRANKW $a_w$ is an $(n \times 2)$ matrix $[p'_w, p''_w]$, concatenating 1st and 2nd derivatives of $p_w$. The ANOmalRANKW score is $[a_w]_1$.

We model smoothness of $[p'_w(t)]_1$ in a normal graph stream using Lipschitz continuity in Assumption 2. Then, similar to what we have shown in the previous Section 4.3.1, we show upper bounds of $[p'_w]_1$ and $[p''_w]_1$ in Lemmas 7 and 8 to explain Assumption 2.
We first calculate updates \( \|p_w(t)\|_1 \) in Normal Stream. In a normal graph stream, \( \|p_w(t)\|_1 \) is Lipschitz continuous with positive real constants \( C_1 \) and \( C_2 \) such that,

\[
\|p_w'(t)\|_1 \leq C_1 \quad \text{and} \quad \|p_w''(t)\|_1 \leq C_2
\]

Lemma 7 (Upper bound of \( \|p_w''(t)\|_1 \)). Given damping factor \( c \), updates \( \Delta A_w \) in the adjacency matrix, and updates \( \Delta b_w \) in the starting vector during \( \Delta t \), \( \|p_w''(t)\|_1 \) is upper-bounded by \( \frac{1}{\Delta t} \left( \|\Delta A_w\|_1 + \|\Delta b_w\|_1 \right) \).

Proof. Proofs are given in Supplement A.3.

In the following lemma, \( \|p_w''\|_{\max} \) denotes the upper bound of \( \|p_w''(t)\|_1 \) which we show in Lemma 6. \( \Delta A_w \) is the changes in \( A_w \) from time \( t \) to \( (t+\Delta t) \), and \( \Delta b_w \) is the changes in \( b_w \) from time \( t \) to \( (t+\Delta t) \). \( \|p_w''(t)\|_1 \) is upper-bounded by \( \frac{1}{\Delta t} \left( \|\Delta A_w\|_1 + \|\Delta b_w\|_1 \right) \).

Lemma 8 (Upper bound of \( \|p_w''(t)\|_1 \)). Given damping factor \( c \), sequencing updates \( \Delta A_w \) and \( \Delta A_w \), and sequencing updates \( \Delta b_w \) and \( \Delta b_w \), the upper bound of \( \|p_w''(t)\|_1 \) is presented as follows:

\[
\|p_w''\|_{\max} + \frac{1}{\Delta t} (\|\Delta A_w\|_1 + c \|\Delta b_w\|_1) \leq \|p_w''(t)\|_1
\]

Proof. Proofs are given in Supplement A.3.

Normal graph streams have small changes (small \( \|\Delta A_w\|_1 \) and small \( \|\Delta b_w\|_1 \)) and evolve gradually (small \( \|\Delta b_w - \Delta b_w\|_1 \)). Then, normal graph streams have small ANOMRANKW scores under small upper bounds of \( \|p_w''(t)\|_1 \) and \( \|p_w''(t)\|_1 \).

Observation 2 (ANOMRANKW in ANOMRANKW). ANOMRANKW involves sudden edge weight changes, inducing large ANOMRANKW.

We explain Observation 2 by showing large upper bounds of \( \|p_w''(t)\|_1 \) and \( \|p_w''(t)\|_1 \) induced by large values of \( \|\Delta A_w\|_1 \) and \( \|\Delta A_w\|_1 \) in ANOMRANKW.

Theorem 2 (Upper bounds of \( \|p_w'(1)\|_1 \) and \( \|p_w'(1)\|_1 \) with ANOMRANKW). When ANOMRANKW occurs with large \( \|\Delta A_w\|_1 \) and \( \|\Delta A_w\|_1 \), L1 lengths of \( p_w \) and \( p_w \) are upper-bounded as follows:

\[
\|p_w'(1)\|_1 \leq c \left( \frac{2}{1 - c} \right) \frac{2 \Delta m}{\Delta T} + \frac{2 \Delta m}{\Delta T}
\]

\[
\|p_w''(1)\|_1 \leq \frac{c}{1 - c} \frac{2 \Delta^2 m}{k} \frac{2 \Delta m}{\Delta T} + \frac{c}{1 - c} \frac{2 \Delta m}{\Delta T} + \frac{2 \Delta m}{\Delta T} + \frac{2 \Delta m}{\Delta T} + \frac{2 \Delta m}{\Delta T}
\]

Proof. Proofs are given in Supplement A.3.

With high upper bounds of \( \|p_w'(1)\|_1 \) and \( \|p_w'(1)\|_1 \), shown in Theorem 2, ANOMRANKW has high ANOMRANKW scores (Figure 1(c)). We detect ANOMRANKW successfully based on ANOMRANKW scores in real-world graphs (Figure 3).

4.4 Algorithm

Algorithm 1 describes how we detect anomalies in a dynamic graph. We first calculate updates \( \Delta A_w \) in the unweighted adjacency matrix, updates \( \Delta A_w \) in the weighted adjacency matrix, and updates \( \Delta b_w \) in the out-edge proportional starting vector (Line 1). These computations are proportional to the number of edge changes, taking a few milliseconds for small changes. Then, ANOMRANK updates SCORES and SCOREW vectors using the update rules in Lemmas 1 and 2 (Line 2). Then ANOMRANK calculates an anomaly score given SCORES and SCOREW in Algorithm 2. ANOMRANK computes ANOMRANKS and ANOMRANKW, and returns the maximum L1 length between them as the anomaly score.

Normalization: As shown in Theorems 1 and 2, the upper bounds of ANOMRANKS and ANOMRANKW are based on the number of out-neighbors \( k \) and the number of out-edge weights \( m_w \). This leads to skew in anomalousness score distributions since many real-world graphs have skewed degree distributions. Thus, we normalize each node’s ANOMRANKS and ANOMRANKW scores by subtracting its mean and dividing by its standard deviation, which we maintain along the stream.

Explainability and Attribution: ANOMRANK explains the type of anomalies by comparing ANOMRANKS and ANOMRANKW: higher scores of ANOMRANKS suggest that ANOMRANKS has happened, and vice versa. High scores of both metrics indicate a large edge weight change that also alters the graph structure. Furthermore, we can localize culprits of anomalies by ranking ANOMRANK scores of each node in the score vector, as computed in Lines 1 and 2 of Algorithm 2. We show this localization experimentally in Section 5.5.

\Delta \text{ selection: Our analysis and proofs, hold for any value of } \Delta. The choice of \( \Delta \) is outside the scope of this paper, and probably best decided by a domain expert: large \( \Delta \) is suitable for slow (‘low temperature’) attacks; small \( \Delta \) spots fast and abrupt attacks. In our experiments, we chose \( \Delta = 1 \) hour, and 1 day, respectively, for a computer-network intrusion setting, and for a who-emails-whom network.

5 EXPERIMENTS

In this section, we evaluate the performance of ANOMRANK compared to state-of-the-art anomaly detection methods on dynamic graphs. We aim to answer the following questions:

1. Q1. Practicality. How fast, accurate, and scalable is ANOMRANK compared to its competitors? (Section 5.2)

2. Q2. Effectiveness of two-pronged approach. How do our two metrics, ANOMRANKS and ANOMRANKW, complement each other in real-world and synthetic graphs? (Section 5.3)


5.3 Effectiveness of Two-Pronged Approach

In this experiment, we show the effectiveness of our two-pronged approach using real-world and synthetic graphs.

5.3.1 Real-World Graph. We measure anomaly scores based on four metrics: AnomRankW, AnomRankS, SedanSpot, and DenseAlert, on the ENRON dataset. In Figure 4, AnomRankW and SedanSpot show similar trends, while AnomRankW detects different events as anomalies on the same dataset. DenseAlert shows similar trends with the sum of AnomRankW and AnomRankS, while missing several anomalous events. This is also reflected in the low accuracy of DenseAlert on the DARPA dataset in Figure 3. The anomalies detected by AnomRankW and AnomRankW coincide with major events in the ENRON timeline\(^2\) as follows:

\(^2\)http://www.agsm.edu.au/bobm/teaching/BE/Enron/timeline.html
Anomaly Score

May 2000

AnomalyW

1

AnomalyS

2

AnomRankS

3

SedanSpot

4

DenseAlert

5

6

9

10

Observation 1. AnomRankS and AnomRankW spot different types of anomalous events.

Observation 2. DenseAlert and SedanSpot detect a subset of the anomalies detected by AnomRank.

5.3.2 Synthetic Graph. In our synthetic graph generated by RTM method, we inject two types of anomalies to examine the effectiveness of our two metrics. Details of the injections are as follows:

- **InjectionS**: We choose 50 timestamps uniformly at random: at each chosen timestamp, we select 8 nodes uniformly at random, and introduce all edges between these nodes in both directions.
- **InjectionW**: We choose 50 timestamps uniformly at random: at each chosen timestamp, we select two nodes uniformly at random, and add 70 edges from the first to the second.

A clique is an example of AnomalyS with unusual structure pattern, while high edge weights are an example of AnomalyW. Hence, InjectionS and InjectionW are composed of AnomalyS and AnomalyW, respectively.

Then we evaluate the precision of the top-50 highest anomaly scores output by the AnomRankS metric and the AnomRankW metric. We also evaluate each metric on the DARPA dataset based on their top-250 anomaly scores. In Table 3, AnomRankS shows higher precision on InjectionS than AnomRankW, while AnomRankW has higher precision on InjectionW and DARPA. In Section 4.2.3, we showed theoretically that AnomalyS induces larger changes in AnomRankS than AnomRankW, explaining the higher precision of AnomRankS than AnomRankW on Injections. We also showed that adding additional edge weights has no effect on AnomRankS, explaining that AnomRankS does not work on InjectionW. For the DARPA dataset, AnomRankW shows higher accuracy than AnomRankS. DARPA contains 2.7M attacks, and 90% of the attacks (2.4M attacks) are DOS attacks generated from only 2-3 source IP addresses toward 2-3 target IP addresses. These attacks are of AnomalyW type with high edge weights. Thus AnomRankW shows higher precision on DARPA than AnomRankS.

5.4 Effectiveness of Two-Derivatives Approach

In this experiment, we show the effectiveness of 1st and 2nd order derivatives of ScoreS and ScoreW in detecting anomalies in dynamic graphs. For brevity, we show the result on ScoreW; result on ScoreS is similar. Recall that AnomRankW score is defined as the L1 length of \( a_w = [p'_w, p''_w] \) where \( p'_w \) and \( p''_w \) are the 1st and 2nd order derivatives of ScoreW, respectively. We define two metrics, AnomRankW-1ST and AnomRankW-2ND, which denote the L1 lengths of \( p'_w \) and \( p''_w \), respectively. By estimating precision using AnomRankW-1ST and AnomRankW-2ND individually, we examine the effectiveness of each derivative using the same injection scenarios and evaluation approach as Section 5.3.2.

In Table 4, AnomRankW-1ST shows higher precision on the DARPA dataset, while AnomRankW-2ND has higher precision on injection scenarios. AnomRankW-1ST detects suspiciously large anomalies based on L1 length of 1st order derivatives, while AnomRankW-2ND detects abruptness of anomalies based on L1 length of 2nd order derivatives. Note that combining 1st and 2nd order derivatives leads to better precision. This shows that 1st and 2nd order derivatives complement each other.
anomaly in the DARPA dataset, nodes (IP addresses) are sorted in order of their AnomRank scores. Outliers with significantly large scores correspond to IP addresses which are likely to be engaged in network intrusion attacks. At the 15th snapshot (\( T = 15 \)) when Back DOS attacks occur, the attacker IP (135.008.060.182) and victim IP (172.016.114.050) have the largest AnomRank scores. In the 133th snapshot (\( T = 133 \)) where Nmap probing attacks take place, the victim IP (172.016.112.050) has the largest score.

6 CONCLUSION

In this paper, we proposed a two-pronged approach for detecting anomalous events in a dynamic graph.

Our main contributions are:

- **Online, Two-Pronged Approach** We introduced AnomRank, a novel and simple detection method in dynamic graphs.
- **Theoretical Guarantees** We present theoretical analysis (Theorems 1 and 2) on the effectiveness of AnomRank.
- **Practicality** In Section 5, we show that AnomRank outperforms state-of-the-art baselines, with up to 49.5× faster speed or 35% higher accuracy. AnomRank is fast, taking about 2 seconds on a graph with 4.5 million edges.

Our code and data are publicly available\(^3\).

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grants No. CNS-1314632 and IIS-1408924. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, or other funding parties. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] Charu C Aggarwal, Yuchen Zhao, and Philip S Yu. 2011. Outlier detection in graph streams. In ICDE.
[2] Leman Akoglu, Mary McGlohon, and Christos Faloutsos. 2008. RTM: Laws and a recursive generator for weighted time-evolving graphs. In KDD.
[3] Leman Akoglu, Mary McGlohon, and Christos Faloutsos. 2010. Oddball: Spotting anomalies in weighted graphs. In PAKDD.
[4] Leman Akoglu, Hanghang Tong, and Danai Koutra. 2015. Graph based anomaly detection and description: a survey. DMKD 29, 3 (2015), 626–688.
[5] Alex Beutel, WanHong Xu, Venkatesan Guruswami, Christopher Palow, and Christos Faloutsos. 2013. Copycatch: stopping group attacks by spotting lockstep behavior in social networks. In WWW.
[6] Deepayan Chakrabarti. 2004. Autopart: Parameter-free graph partitioning and outlier detection. In PKDD.
[7] F.R.K. Chung. 1994. Spectral Graph Theory. Number no. 92 in CBMS Regional Conference Series. Conference Board of the Mathematical Sciences.
[8] Dhiyya Eswaran and Christos Faloutsos. 2018. SEDANSPT: Detecting Anomalies in Edge Streams. In KDD.
[9] Dhiyya Eswaran, Christos Faloutsos, Sudipto Guha, and Nina Mishra. 2018. SpotLight: Detecting Anomalies in Streaming Graphs. In KDD.
[10] Keith Henderson, Tina Eliassi-Rad, Christos Faloutsos, Leman Akoglu, Lei Li, Koji Maruhashi, B Aditya Prakash, and Hanghang Tong. 2010. Metric forensics: a multi-level approach for mining volatile graphs. In KDD.
[11] Bryan Hooi, Kijung Shin, Hyun Ah Song, Alex Beutel, Neil Shah, and Christos Faloutsos. 2017. Graph-based fraud detection in the face of camouflage. TKDD 11, 4 (2017), 44.
[12] Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, and Shiqiang Yang. 2016. Catching synchronized behaviors in large networks. A graph mining approach. TKDD 10, 4 (2016), 35.
[13] Jon M Kleinberg, 1999. Authoritative sources in a hyperlinked environment. JACM 46, 5 (1999), 604–632.
[14] Danai Koutra, Neil Shah, Joshina T Vogelstein, Brian Gallagher, and Christos Faloutsos. 2016. DeltaCon: principled massive-graph similarity function with attribution. TKDD 10, 3 (2016), 28.
[15] Ronny Lempel and Shlomo Moran. 2001. SALSA: the stochastic approach for link-structure analysis. ACM Trans. Inf. Syst. 19, 2 (2001), 131–160.
[16] Richard Lippmann, Robert K Cunningham, David J Fried, Isaac Graf, Kris R Kendall, Seth E Webster, and Marc A Zissman. 1999. Results of the DARPA 1998 Offline Intrusion Detection Evaluation. In Recent advances in intrusion detection, Vol. 99, 829–835.
[17] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The PageRank citation ranking: Bringing order to the web. Technical Report. Stanford InfoLab.
[18] Stephen Ranshous, Steve Harenberg, Kohiti Sharma, and Nagiza F Samatova. 2016. A scalable approach for outlier detection in edge streams using sketch-based approximations. In SDM.
[19] Jitesh Shetty and Jafar Abbri. 2004. The Enron email dataset database schema and brief statistical report. Information sciences institute technical report, University of Southern California 4, 1 (2004), 120–128.
[20] Kijung Shin, Tina Eliassi-Rad, and Christos Faloutsos. 2018. Patterns and anomalies in k-cores of real-world graphs with applications. KDD-S 4, 3 (2018), 677–710.
[21] Kijung Shin, Bryan Hooi, and Christos Faloutsos. 2018. Fast, Accurate, and Flexible Algorithms for Dense Subtensor Mining. TKDD 12, 3 (2018), 28.
[22] Kijung Shin, Bryan Hooi, Jisu Kim, and Christos Faloutsos. 2017. DenseAlert: Incremental dense-subtensor detection in tensor streams. In KDD.
[23] Jinseng Sun, Daecheng Tao, and Christos Faloutsos. 2006. Beyond streams and graphs: dynamic tensor analysis. In KDD.
[24] Hanghang Tong and Ching-Yung Lin. 2011. Non-negative residual matrix factorization with application to graph anomaly detection. In SDM.
[25] Teng Wang, Chunsheng Fang, Derek Lin, and S Felix Wu. 2015. Localizing temporal anomalies in large evolving graphs. In SDM.
[26] Stanley Wasserman and Katherine Faust. 1994. Social network analysis: Methods and applications. Cambridge university press.
[27] Xiaowei Xu, Nurcan Yuruk, Zhidan Feng, and Thomas Al Schweiger. 2007. Scan: a structural clustering algorithm for networks. In KDD.
[28] Minji Yoon, Wooseong Jin, and U Kang. 2018. Fast and Accurate Random Walk with Restart on Dynamic Graphs with Guarantees. In WWW.
[29] Minji Yoon, Jinhong Jung, and U Kang. 2018. TPA: Fast, Scalable, and Accurate method for Approximate Random Walk with Restart on Billion Scale Graphs. In ICDE.
A SUPPLEMENT

A.1 Experimental Setting

All experiments are carried out on a 3 GHz Intel Core i5 iMac, 16 GB RAM, running OS X 10.13.6. We implemented ANOMRank and SedanSpot in C++, and we used an open-sourced implementation of DenseAlert, provided by the authors of [22]. To show the best trade-off between speed and accuracy, we set the sample size to 50 for SedanSpot and follow other parameter settings as suggested in the original paper [8]. For ANOMRank, we set the damping factor c to 0.5, and stop iterations for computing node score vectors when L1 changes of node score vectors are less than $10^{-3}$.

A.2 Dataset

**DARPA** [16] has 4.5M IP-IP communications between 9.4K source IP and 2.3K destination IP over 87.7K minutes. Each communication is a directed edge (srcIP, dstIP, timestamp, attack) where the attack label indicates whether the communication is an attack or not. We aggregate edges occurring in every hour, resulting in a stream of 1463 graphs. We annotate a graph snapshot as anomalous if it contains at least 50 attack edges. Then there are 288 ground truth anomalies (23.8% of total). We use the first 256 graphs for initializing means and variances needed during normalization (as described in Section 4.4).

**ENRON** [19] contains 50K emails from 151 employees over 3 years in the ENRON Corporation. Each email is a directed edge (sender, receiver, timestamp). We aggregate edges occurring in every day duration, resulting in a stream of 1139 graphs. We use the first 256 graphs for initializing means and variances.

**RTM method** [2] generates time-evolving graphs with repeated Kronecker products. We use the publicly available code[5]. The generated graph is a directed graph with 1K nodes and 81K edges over 2.7K timesteps. We use the first 300 timestamps for initializing means and variances.

A.3 Proofs

We prove upper bounds on the 1st and 2nd derivatives of ScoreS and ScoreW, showing their effectiveness in detecting ANOMALYs and ANOMALYW.

**Proof of Lemma 5 (Upper bound of $\|p_\omega^{\prime}\|_1$).**

For brevity, $p_\omega^\prime \leftarrow p_\omega(t + \Delta t), p_\omega^{\prime\prime} \leftarrow p_\omega(t)$. By Lemma 1, $\|p_\omega^{\prime\prime} - p_\omega^{\prime}\|_1$ is presented as follows:

$$\|p_\omega^{\prime\prime} - p_\omega^{\prime}\|_1 = \|\sum_{k=0}^{\infty} c^k (\hat{A}_\omega + \Delta A_\omega)^k c (\Delta A_\omega p_\omega^{\prime\prime})\|_1$$

$$\leq c \sum_{k=0}^{\infty} \|c^k (\hat{A}_\omega + \Delta A_\omega)^k\|_1 \|\Delta A_\omega p_\omega^{\prime\prime}\|_1$$

$$\leq \frac{c}{1-c} \|\Delta A_\omega p_\omega^{\prime}\|_1 \leq \frac{c}{1-c} \|\Delta A_\omega\|_1$$

$$\|p_\omega^{\prime}\|_1 = \|p_\omega^{\prime\prime} - p_\omega^{\prime}\|_1 \leq \frac{c}{1-c} \|\Delta A_\omega\|_1$$

Note that $\|\Delta A_\omega\|_1 = 1$, since $\Delta A_\omega$ is a column-normalized stochastic matrix, and $p_\omega^{\prime\prime}$ is a PageRank vector.

**Proof of Lemma 6 (Upper bound of $\|p_\omega^{\prime}\|_1$).**

For brevity, $p_\omega \leftarrow p_\omega(t - \Delta t), p_\omega \leftarrow p_\omega(t), p_\omega \leftarrow p_\omega(t + \Delta t), \Delta p_\omega \leftarrow p_\omega - p_\omega - p_\omega, A \leftarrow \hat{A}_\omega + \Delta A_\omega, \Delta A_\omega \leftarrow \Delta A_\omega$. In addition, we omit c by substituting $A \leftarrow cA$ and $\Delta A \leftarrow c\Delta A$ during this proof. By Lemma 1, $\Delta p_\omega$ is:

$$\Delta p_\omega = \sum_{k=0}^{\infty} (A + \Delta A_1 + \Delta A_2)^k (\Delta A_3 p_1)$$

$\Delta p_\omega$ can be viewed as an updated ScoreS with the original adjacency matrix $Y_1 = (A + \Delta A_1)$, the update $\Delta A_2$, and the starting vector ($\Delta A_3 p_1$) from an original vector $p_{temp} = \sum_{k=0}^{\infty} Y_1^k (\Delta A_2 p_1)$.

Then, by Lemma 1, $\Delta p_\omega$ is presented as follows:

$$\Delta p_\omega = p_{temp} + \sum_{k=0}^{\infty} (Y_1 + \Delta A_3)^k \Delta A_3$$

Thus $\Delta p_\omega - \Delta p_\omega$ becomes as follows:

$$\|\Delta p_\omega - \Delta p_\omega\|_1 = \sum_{k=0}^{\infty} \|Y_1^k (\Delta A_3 p_1)\|_1 + \sum_{k=0}^{\infty} \|Y_1^k (\Delta A_3 p_1)\|_1 + \|Y_1^k (\Delta A_3 p_1)\|_1$$

$$\|p_\omega^{\prime}\|_1 = \|p_\omega^{\prime\prime} - p_\omega^{\prime}\|_1 \leq \frac{c}{1-c} \|\Delta A_\omega\|_1$$

Note that $A + \Delta A_1$ and $A + \Delta A_1 + \Delta A_2$ are column-normalized stochastic matrices. Then $\|\Delta p_\omega - \Delta p_\omega\|_1$ is bounded as follow:

$$\|\Delta p_\omega - \Delta p_\omega\|_1 \leq \frac{c}{1-c} \|\Delta A_2 - \Delta A_\omega\|_1 + (\frac{c}{1-c})^2 (\|\Delta A_2\|_1^2 + \|\Delta A_2\|_2^2)$$

Then, recovering $c$ from all terms, $\|p_\omega^{\prime}\|_1$ is bounded as follows:

$$\|p_\omega^{\prime}\|_1 = \frac{\|\Delta p_\omega - \Delta p_\omega\|_1}{\Delta t^2}$$

$$\leq \frac{c}{1-c} \|\Delta A_2 - \Delta A_\omega\|_1 + (\frac{c}{1-c})^2 (\|\Delta A_2\|_1^2 + \|\Delta A_2\|_2^2)$$

**Proof of Theorem 1.** (Upper bounds of $\|p_\omega^{\prime}\|_1$ and $\|p_\omega^{\prime\prime}\|_1$ with ANOMALYs) Use Lemma 3 and 5.

**Proof of Lemma 7 (Upper bounds of $\|p_\omega^{\prime}\|_1$ and $\|p_\omega^{\prime\prime}\|_1$ with ANOMALYs).** For brevity, denote $p_\omega^{\prime\prime} \leftarrow p_\omega(t)$ and $p_\omega^{\prime\prime} \leftarrow p_\omega(t + \Delta t)$. By Lemma 2,
\[ \|p^n_w - p^\Delta_w\|_1 \text{ is presented as follows:} \]
\[ p^n_w - p^\Delta_w = \sum_{k=0}^{\infty} c^k (\tilde{A}^T_w + \Delta A_w)^k c A A_w p^n_w \]
\[ + (1 - c) \sum_{k=0}^{\infty} c^k (\tilde{A}^T_w + \Delta A_w)^k A b_w' \]
\[ \|p^n_w - p^\Delta_w\|_1 \leq \frac{c}{1 - c} \|\Delta A_w\| + \|A b_w\| \]
\[ \|p^\Delta_w\|_1 = \|p^n_w\|_1 = 1 \text{ since } \tilde{A}^T_w + \Delta A_w \text{ is a column-normalized stochastic matrix and } p^n_w \text{ is a PageRank vector.} \]

**Proof of Lemma 8 (Upper bound of }\|p^\Delta_w\|_1).**
For brevity, denote \( p_0 \leftarrow p_w(t - \Delta t), \) \( p_1 \leftarrow p_w(t), \) \( p_2 \leftarrow p_w(t + \Delta t), \) \( \Delta p^0 \leftarrow p_1 - p_0, \) \( \Delta p^\Delta \leftarrow p_2 - p_1, \) \( A \leftarrow \tilde{A}^T_w, \) \( \Delta A_1 \leftarrow \Delta A_{w_1}, \) \( \Delta A_2 \leftarrow \Delta A_{w_2}, \) \( \Delta A_{w_1} \leftarrow \Delta b_{w_1} \) and \( \Delta b_2 \leftarrow \Delta b_{w_2}. \) In addition, we omit the \( c \) term by substituting \( A \leftarrow c A, \) \( \Delta A \leftarrow c \Delta A \) and \( \Delta b \leftarrow (1 - c) \Delta b \) during this proof. By Lemma 2, \( \Delta p^0 \) and \( \Delta p^\Delta \) are presented as follows:
\[ \Delta p^0 = \sum_{k=0}^{\infty} (A + \Delta A_1)^k A b_1 + \sum_{k=0}^{\infty} (A + \Delta A_2)^k A b_2 \]
\[ \Delta p^\Delta = \sum_{k=0}^{\infty} (A + \Delta A_1 + \Delta A_2)^k A b_1 + \sum_{k=0}^{\infty} (A + \Delta A_1 + \Delta A_2)^k A b_2 \]
Subtracting the first term of \( \Delta p^\Delta \) from the first term of \( \Delta p^0 \) is equal to \( p^\Delta_w \) as shown in Lemma 6. Then \( \Delta p^\Delta = \Delta p^\Delta. \)

By substituting \( A + \Delta A_1 \) with \( Y_2 \), the last two terms in the above equation are presented as follows:
\[ \sum_{k=0}^{\infty} (Y_2 + \Delta A_2)^k A b_2 \]
\[ = \sum_{k=0}^{\infty} \sum_{i=0}^{\infty} (Y_2 + \Delta A_2)^k A b_2 \]
\[ = \sum_{k=0}^{\infty} \sum_{i=0}^{\infty} (Y_2 + \Delta A_2)^k A b_2 \]
\[ + \frac{c}{1 - c} \|\Delta A_2\| \|A b_2\| \]

In the first equation, we treat \( \sum_{k=0}^{\infty} (Y_2 + \Delta A_2)^k A b_2 \) as an updated PageRank with the update \( \Delta A_2 \) from an original PageRank \( \sum_{k=0}^{\infty} Y_2^k A b_2 \), then apply Lemma 1. Note that both \( \|\sum_{k=0}^{\infty} Y_2^k\|_1 \) and \( \|\sum_{k=0}^{\infty} (Y_2 + \Delta A_2)^k\|_1 \) have value \( \frac{1}{1 - c} \) since the original expressions with \( c \) terms are as follows:
\[ \sum_{k=0}^{\infty} Y_2^k = \sum_{k=0}^{\infty} c^k (\tilde{A}^T_w + \Delta A_{w_1})^k \]
\[ \sum_{k=0}^{\infty} (Y_2 + \Delta A_2)^k = \sum_{k=0}^{\infty} c^k (\tilde{A}^T_w + \Delta A_{w_1} + \Delta A_{w_2})^k \]
\( (\tilde{A}^T_w + \Delta A_{w_1}) \) and \( (\tilde{A}^T_w + \Delta A_{w_1} + \Delta A_{w_2}) \) are column-normalized stochastic matrices. Then, recovering \( c \) from all terms, \( \|p^\Delta_w\|_1 \) is bounded as follow:
\[ \|\Delta p^\Delta - \Delta p^\Delta\|_1 \]
\[ \leq \frac{\|p^n_w\|_1}{\Delta t^2} \]
\[ + \frac{1}{\Delta t} (\|A b_2 - A b_1\|_1 + \frac{c}{1 - c} \|A b_2\|_1) \]

**Proof of Theorem 2. (Upper bounds of }\|p^\Delta_w\|_1 \text{ and } \|p^\Delta_w\|_1 \text{ with } ANOMALY\text{W) } Use Lemma 4, 6 and 8.**