Localizing anomalous changes in time-evolving graphs

Kumar Sricharan
Palo Alto Research Center
sricharan.kumar@parc.com

Kamalika Das
UARC, NASA Ames Research Center
kamalika.das@nasa.gov

ABSTRACT

Given a time-evolving sequence of undirected, weighted graphs, we address the problem of localizing anomalous changes in graph structure over time. In this paper, we use the term 'localization' to refer to the problem of identifying abnormal changes in node relationships (edges) that cause anomalous changes in graph structure. While there already exist several methods that can detect whether a graph transition is anomalous or not, these methods are not well suited for localizing the edges which are responsible for a transition being marked as an anomaly. This is a limitation in applications such as insider threat detection, where identifying the anomalous graph transitions is not sufficient, but rather, identifying the anomalous node relationships and associated nodes is key. To this end, we propose a novel, fast method based on commute time distance [23] called **CAD** (Commute-time based Anomaly detection in Dynamic graphs) that detects node relationships responsible for abnormal changes in graph structure. In particular, CAD localizes anomalous edges by tracking a measure that combines information regarding changes in graph structure (in terms of commute time distance) as well as changes in edge weights. For large, sparse graphs, CAD returns a list of these anomalous edges and associated nodes in $O(n \log n)$ time per graph instance in the sequence, where n is the number of nodes. We analyze the performance of CAD on several synthetic and real-world data sets such as the Enron email network, the DBLP co-authorship network and a worldwide precipitation network data. Based on experiments conducted, we conclude that CAD consistently and efficiently identifies anomalous changes in relationships between nodes over time.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database applications—*Data Mining*

SIGMOD '14 Snowbird, Utah USA

Keywords

graph analysis, temporal graphs, anomaly localization, anomaly detection, commute time distance, random walks

1. INTRODUCTION

Mining for anomalies in data sets is an important problem [7], with diverse areas of application including security, finance, health care and several other industries. While this problem has been studied extensively for unstructured data sets, it is only recently that methods have been developed for detecting outliers for structured forms of data such as graph data. Graph structured data has become ubiquitous as a powerful representation framework for relational data such as social interactions, communication networks, web graphs, citation networks, gene interaction networks etc. Detecting anomalies in graphs is necessary to address several problems such as network intrusion, spam filtering, gene network analysis, and churn analysis in social networks.

Detection of anomalous nodes in *static* graphs has been well studied [10, 17, 2, 14]. Recently, several papers have addressed the problem of anomaly detection in *dynamic* graphs. These include methods based on temporal clustering [20], node-based features extraction [1], matrix decomposition [21], graph distance based time series analysis methods [18], and spectral methods [12]. All of these methods for dynamic graphs, however, are designed for event detection, *i.e.* to detect when anomalous changes in graph structure occur, and do not address the problem of localization, *i.e.* identifying changing node relationships that contribute to such anomalous changes in graph structure. In a subset of these methods ([1, 12]), it is possible to identify a set of anomalous nodes by computing their contribution to the overall anomaly score at each temporal transition. While this set of nodes that contribute to the anomaly score may include nodes that exhibit anomalous changes in relationships, they also include other non-anomalous nodes that are either affected by such structural changes or exhibit benign changes in relationships, resulting in a large number of false alarms. This point is further elaborated in Section 3.4.

In this paper, we are interested in identifying anomalous changes in node relationships that are responsible for abnormal changes in graph structure, and the corresponding anomalous nodes associated with these edges. The motivation for our work comes from the problem of identifying malicious individuals in an organization by detecting abnormal or unexpected changes in their social interactions (email, instant messenger etc.) with other employees in the organization [16]. Graph based insider threat detection has

been previously studied, but only for static graphs [10].

To this end, we propose a novel method based on commute time distance [23] called **CAD** (Commute-time based Anomaly detection in Dynamic graphs) to localize anomalous changes in graph structure. The commute time distance between any pair of nodes is the average time needed to traverse the distance between these two nodes via random walks [23]. CAD localizes anomalous edges by tracking a measure that combines information regarding changes in graph structure (in terms of commute time distance) as well as change in edge weights. The anomalous edge information is subsequently summarized to identify the anomalous nodes associated with these edges. For large, sparse graphs with n nodes, CAD has a run-time of $O(n \log n)$ per graph instance in the sequence.

Finally, we note that while our motivation comes from the specific problem of insider threat detection, CAD can be applied to several other problems where detecting edges and associated nodes responsible for changes in structure is of interest. Examples include scientific collaboration networks where CAD can be used to identify changes in collaboration patterns, and in networks of geographic locations where CAD can be used to identify abnormal climate patterns over time. In this paper, we verify using ground truth information or external anecdotal evidence that CAD consistently finds anomalous changes in node relationships over time for several synthetic and real-world data sets including the Enron data, the DBLP data and world-wide precipitation data.

The rest of this paper is organized as follows. In Section 2, the problem framework is described. In Section 3, CAD is discussed as a solution to the problem. Section 4 describes the results obtained when CAD is applied to synthetic and real-world data sets. Conclusions are given in Section 5.

2. PROBLEM FRAMEWORK

Let G_t, $t = 1, \ldots, T$ be a temporal sequence of graphs. Each G_t is a weighted undirected graph with a fixed vertex (node) set $V = \{v_1, .., v_n\}$. Define the edge set $E = \{e_{(1,1)}, .., e_{(n,n)}\}$ of size n^2, with $e_{(i,j)}$ denoting the edge between nodes v_i and v_j. For each graph G_t, denote the corresponding symmetric adjacency matrix by $A_t \in \mathbb{R}^{n \times n}$, *i.e.* the edge weight of edge $e_{(i,j)}$ in graph G_t is given by $A_t(i, j)$.

Note that our definition of the edge set E includes the set of all n^2 edges in a complete graph. This does not imply that the graphs G_t, $t = 1, \ldots, T$ are complete. Rather, the absence of an edge between nodes i and j in graph G_t is reflected by $A_t(i, j) = 0$. Let the average number of edges with non-zero edge weight in the graphs G_t, $t = 1, \ldots, T$ per time instance be given by m. In our experiments in Section 4, we find that the large, real world graphs we analyze are sparse graphs [3], where $m = O(n) \ll n^2$.

2.1 Problem Statement

For each transition between graph instances G_t and G_{t+1}, our goal is to detect the set of anomalous edges $E_t \subseteq E$ whose change in weights are responsible for structural differences between G_t and G_{t+1}. We are interested in the following three different cases involving changes in edge weight. They can be categorized as:

i. Case 1: high magnitude change (increase or decrease) in edge weight from time t to $t + 1$.

ii. Case 2: new edges that bring distant nodes closer.

iii. Case 3: decrease in edge weight (or deletion of edges) between central or bridge nodes in the graph that push proximal nodes far apart.

We illustrate the reason for choosing these three specific cases in Section 2.2 using a toy example. We note that there exist other possible cases involving changes in edge weights such as small changes in edge weights including addition or deletion of edges between tightly-coupled node pairs (node pairs $i, j \in V$ which are well-connected via neighboring nodes even if the edge between i, j is deleted) which are not significant enough for the overall graph structure to change substantially. Such cases are not of interest in this paper. A node $v_i \in V$ is deemed to be anomalous with respect to the transition from t to $t + 1$ if $e_{(i,j)} \in E_t$ for some $j \in \{1, .., n\}$. Denote the set of anomalous nodes with respect to the transition from t to $t + 1$ as V_t. Note that the anomaly sets E_t and V_t can be an empty sets in the absence of any anomalous structural differences in the transition from time t to $t + 1$.

2.2 Illustrative example

Let us consider a dynamic undirected weighted graph of 17 nodes that are loosely partitioned into two sets: blue(b) and red(r). The nodes are labeled $b_1, .., b_8, r_1.., r_9$. In Figure 1, two instances of this graph at time slices t and $t + 1$ are shown. In time slice t, the edge weight (indicating say, the number of communications) is indicated by the width of the black lines. In time slice $t+1$, black lines are used to indicate edges which remain unchanged in the transition from t to $t+1$, while green lines indicate modified edges. Bold green lines indicate edges whose weights have increased, while dotted green lines indicate edges whose weights have decreased.

Change in edge weights: We enumerate the set of changes in edge weights from time slice t to $t + 1$ with respect to Figure 1. We stress that the 5 different scenarios (S1:S5) considered here are only example instances of the cases described in Section 2.1

- *S1: New edge between* b_1, r_1 (refers to Case 2)
- *S2: Small decrease in edge weight between* r_7, r_8 (refers to Case 3)
- *S3: Large increase in edge weight between* b_4, b_5 (refers to Case 1)
- *S4: Small decrease in edge weight between* b_1, b_3
- *S5: Small increase in edge weight between* b_2, b_7

Real-world scenarios: Anomalies in dynamic graphs mean different things in different application contexts. For example, in an insider threat detection application, anomalies are people who might have malicious intentions that get reflected in some of their social interactions with colleagues. In an organizational event detection setting, the anomalous communication patterns among top corporate, finance, and legal executives are indicative of a major event involving the organization. On the other hand, for a scientific collaboration network, anomalies are simply interesting observations indicating change in research interests, unexpected collaborations, and so on. In the interest of space, we map our toy example to two such real-world dynamic network settings.

Insider threat detection setting: In an insider threat detection setting, these nodes can be thought of as 17 employees in an organization, who are loosely partitioned into two sets of users based on their communication patterns, per-

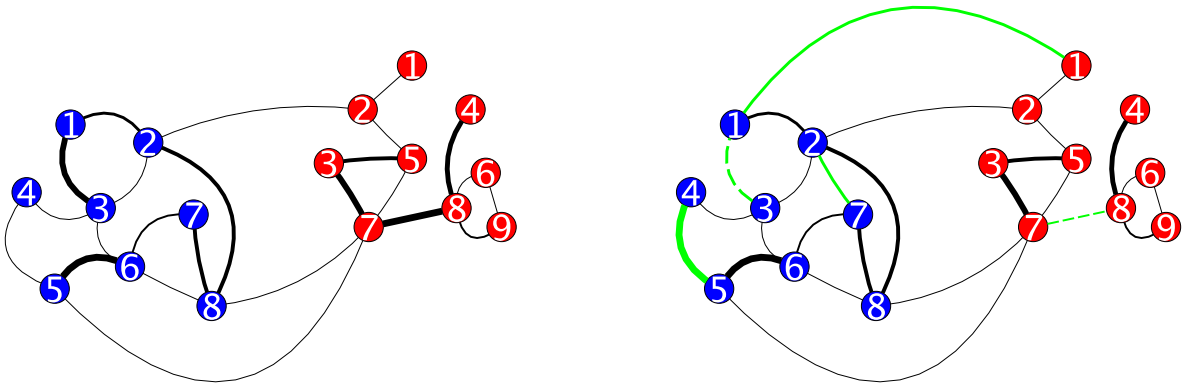


Figure 1: Time slices t (left) and $t+1$ (right) of the dynamic graph in the toy example with 17 nodes labeled $b_1, .., b_8, r_1.., r_9$. In time slice t, the edge weight is indicated by the width of the black lines. In time slice $t+1$, black lines are used to indicate edges which remained unchanged, while green lines indicate modified edges. Bold green lines indicate new edges, while dotted green lines indicate existing edges whose weights have decreased. The two sets of node colors, blue and red, indicate two loosely connected sets of nodes with limited interactions among them in time slice t.

haps indicative of different groups within the organization. In this case, S1 is suspicious because it involves contact between two employees who are fairly unrelated in terms of their job roles and can be indicative of collusion. S2 can reveal employee actions responsible for creating inter-group factions with possibly malicious motivation. Likewise, S3 is suspicious because the large increase in communication can be indicative of an organized collusion effort. On the other hand, S4 and S5 reflect the dynamic nature of the graph and are benign from the perspective of threat detection.

Scientific collaboration network setting: In a author collaboration network setting, these nodes can be thought of as 17 different authors, who are loosely partitioned into two sets of authors based on their research areas. In this case, S1 is interesting because it can indicate collaboration between two authors who are fairly unrelated in terms of their research areas. Likewise, S3 is interesting because it suggests much stronger and probably a more expanded research collaboration, even possibly the presence of a new funding mechanism. The nodes connected to these edges can be considered the seeds for discovering a new micro-community within the research area. The weakening of strong ties within a community, as described in S2, can probably indicate the splitting of a big research area into distinct groups.

In summary, the first three edge weight changes would imply significant change in relationship between nodes in the graph, either due to a increase (decrease) in the amount of communication between two nodes that are weakly (strongly) connected, or due to sharp change in volume of such communication. Such changes are therefore examples of anomalous changes. The last two changes reflect the dynamic nature of the graph and should be treated as benign from the perspective of threat detection. Based on this setup, the nodes $r_1, r_7, r_8, b_1, b_4, b_5$ connected through the anomalous edges should be considered as anomalous nodes.

In the subsequent sections, we solve the problem defined in Section 2.1 by casting it as an optimization problem.

2.3 Graph distance metrics

For any subset of edges $S \subset E$, let $\bar{d}_S(G, H)$ be a generic notion of distance between any two graphs G, H that captures structural differences (of the form highlighted in Section 2.1 (listed as Cases (i), (ii) and (iii))) that arise due to abnormal changes in the edges in the complimentary set $E - S$ when transitioning between G and H. This notion is formalized via the following definition:

DEFINITION 2.1. *Given a distance function $\bar{d}_S(G, H)$ between any two graphs G, H that captures structural differences due to changes in edges in $E - S$ and a dissimilarity threshold δ, two graphs G and H are considered similar with respect to the edge set $E - S$ at level δ if $\bar{d}_S(G, H) < \delta$, and considered dissimilar if $\bar{d}_S(G, H) \geq \delta$.*

Given a graph G_t at time instance t, and its instantiation G_{t+1} at time instance $t+1$, a significant structural change is then said to occur between graphs G_t and G_{t+1} with respect to the edges in $E-S$ when one or more edge weights in $E-S$ change in magnitude such that $\bar{d}_S(G_t, G_{t+1}) \geq \delta$.

If $\bar{d}_S(G_t, G_{t+1}) < \delta$ for some subset S, then by Definition 2.1, $E - S$ excludes the anomalous set E_t with respect to $\bar{d}_S(.,.)$ at level δ, *i.e.*, $E_t \subseteq S$. On the other hand, if $\bar{d}_S(G_t, G_{t+1}) > \delta$, then $S - E_t \neq \phi$. The quantity $\bar{d}_S(G_t, G_{t+1})$ can therefore be minimized over S in order to identify the anomalous set E_t with respect to $\bar{d}_S(.,.)$ at level δ during the transition from t to $t+1$. Based on this, we define the problem of structural change identification in temporal graphs using an optimization framework.

2.4 Optimization framework

We define the set of anomalous edges E_t with respect to $\bar{d}_S(.,.)$ at level δ as solution to the optimization problem:

$$E_t := \arg\min_S |S| \quad \text{subject to} \quad \bar{d}_S(G_t, G_{t+1}) < \delta. \tag{1}$$

The goal of this optimization problem is to identify anomalous edges as the minimal set of edges, which when left unchanged during the transition from t to $t+1$, would result in the graphs $G_t(S)$ and $G_{t+1}(S)$ being structurally similar at level δ with respect to $\bar{d}_S(.,.)$, *i.e.* $\bar{d}_S(G_t, G_{t+1}) < \delta$.

2.4.1 Polynomial-time solution

Observe that the above formulation defined in (1) is a combinatorial optimization problem that is intractable in practice for any reasonably large sized graphs. The solution to this problem, however, can be reduced to polynomial time if $\bar{d}_S(.,.)$ has the following structure: for any $S \subseteq E$,

$$\bar{d}_S(G_t, G_{t+1}) = \sum_{e \in E-S} \Delta E_t(e), \quad (2)$$

where $\Delta E_t(e)$ is a non-negative functional of the graphs G_t and G_{t+1} independent of the set S. If the distance metric satisfies the structure in (2), the optimization problem in (1) is equivalent to simply sorting the set of values

$$\{\Delta E_t(e); \forall e \in E\}$$

and choosing E_t to be the smallest set of edges S (in terms of the cardinality $|S|$ of S) such that $\sum_{e \in E-S} \Delta E_t(e) < \delta$, which can be solved in polynomial time.

2.4.2 Existing distance measures

Several distance functions between graphs have been proposed previously, including (i) Maximum Common Subgraph [19], (ii) Graph edit distance [11], (iii) Modality distance [6] and (iv) Spectral distance [13]. However, none of these distance measures satisfy (2), which means it would require exhaustive search on the candidate edges for solving the combinatorial optimization problem in (1). To address this, we propose a new distance measure that satisfies (2).

2.5 Proposed distance metric

Let $d_t(i,j)$ represent some suitable notion of distance between two nodes v_i and v_j with respect to graph G_t. Then, a possible measure for capturing the structural distance between the graphs G_t and G_{t+1} due to any changes corresponding to scenarios (i), (ii) or (iii) would be: For any subset $S \subseteq E$, define

$$\bar{d}_S^{(1)}(G_t, G_{t+1}) = \sum_{e(i,j) \in E-S} |d_t(i,j) - d_{t+1}(i,j)|.$$

However, this distance measure $\bar{d}_S^{(1)}(G_t, G_{t+1})$ is a poor choice for identifying anomalous edges for the following reason. The use of $\bar{d}_S^{(1)}(G_t, G_{t+1})$ in (1) would result in E_t being the set of edges $e_{(i,j)}$ with large values for $|d_t(i,j) - d_{t+1}(i,j)|$. However, structural changes induced by an anomalous edge $e(i,j)$, not only causes the distance between nodes v_i and v_j to change sharply (*i.e.*, $|d_t(i,j) - d_{t+1}(i,j)| \gg 0$), but also causes the distance between nodes connected to node v_i and nodes connected to node v_j to change significantly. This is illustrated in detail in Section 3.4 (please see discussion on COM method) via the illustrative example described earlier. As a result, employing $\bar{d}_S^{(1)}(G_t, G_{t+1})$ in (1) would result in E_t consisting of several benign or non-anomalous edges.

To address this issue with $\bar{d}_S^{(1)}(G_t, G_{t+1})$, we propose the following modified distance function. For any $S \subseteq E$, define

$$\bar{d}_S^{(0)}(G_t, G_{t+1}) = \sum_{e \in E-S} \Delta E_t(e),$$

where $\Delta E_t(e)$ for $e = e_{i,j}$is given by

$$\Delta E_t(e_{i,j}) = |A_{t+1}(i,j) - A_t(i,j)| \times |d_{t+1}(i,j) - d_t(i,j)|.$$

For anomalous edges corresponding to large changes in magnitude (Case 1), $|A_{t+1}(i,j) - A_t(i,j)|$ will be large, which will result in $|c_{t+1}(i,j) - c_t(i,j)|$ being large and as a result, $\Delta E_t(e_{i,j})$ will be large. For edges corresponding to Case 2 involving new edges that bring distant nodes together, $|d_{t+1}(i,j) - d_t(i,j)|$ will be large while $A_{t+1}(i,j) - A_t(i,j)$ will be non-zero and positive and as a result $\Delta E_t(e_{i,j})$ will be large. Finally, in Case 3, once again $|d_{t+1}(i,j) - d_t(i,j)|$ will be large while $A_{t+1}(i,j) - A_t(i,j)$ will be non-zero and negative, and once again $\Delta E_t(e_{i,j})$ will be large. Thus, edges corresponding to all the three different cases will be marked with a high anomaly score.

For non-anomalous edges $e(i,j)$ involving small magnitude changes $|A_{t+1}(i,j) - A_t(i,j)|$ between node-pairs i,j that are tightly coupled, $|d_{t+1}(i,j) - d_t(i,j)|$ will also be small due to the coupling and therefore the product $\Delta E_t(e_{i,j})$ will also be small. The other scenario involving non-anomalous edges is as follows: let $e(i,j)$ be an anomalous edge corresponding to Cases 2 or 3. In this case, for some neighboring node of i (say n_i) and j (say n_j), the distance $|d_{t+1}(n_i, n_j) - d_t(n_i, n_j)|$ will be large because $|d_{t+1}(i,j) - d_t(i,j)|$ will be large. But because the edge between n_i and n_j is non-anomalous, and therefore does not belong to Cases 2 or 3, it follows that $|A_{t+1}(i,j) - A_t(i,j)|$ will be 0, and as a result $\Delta E_t(e_{i,j})$ will be 0. In either case, the score $\Delta E_t(e_{(i,j)})$ will be small for non-anomalous edges.

As a result, employing $\bar{d}_S^{(0)}(G_t, G_{t+1})$ in (1) would result in E_t primarily consisting of edges with abnormal changes corresponding to Cases (i), (ii) and (iii) that are of interest in this paper. Furthermore, observe that condition (2) is trivially satisfied by $\bar{d}_S^{(0)}(G_t, G_{t+1})$. Because of these desirable properties, we use $\bar{d}_S^{(0)}(G_t, G_{t+1})$ in the CAD algorithm (described next) for localizing anomalous edges and nodes.

3. COMMUTE TIME BASED ANOMALY DETECTION

In this section, we review the commute time distance measure, and subsequently describe the CAD algorithm based on the distance metric $\bar{d}_S^{(0)}(G_t, G_{t+1})$ proposed in Section 2.5.

3.1 Commute time distance

Given an arbitrary adjacency matrix A, a random walk on the corresponding graph is a sequence of nodes described by a finite Markov chain which is time-reversible [11]. The commute time is the expected number of steps that a random walk starting at i will take to reach j once and go back to i for the first time [14]. The commute time can be computed from the Moore-Penrose pseudoinverse of the graph Laplacian matrix. Denote the degree matrix corresponding to the weighted adjacency matrix A by D. Further denote $L = D - A$ and L^+ as the graph Laplacian matrix and its pseudoinverse respectively. The commute time is:

$$c(i,j) = V_G(l_{ii}^+ + l_{jj}^+ - 2l_{ij}^+), \quad (3)$$

where $V_G = \sum_{i=1}^n D(i,i)$ is the volume of the graph, and l_{ij}^+ is the (i,j) element of L^+. The commute time $c(i,j)$ has been shown to be a valid distance metric between any pair of nodes i,j. In particular, the commute time between the nodes on a graph is the Euclidean distance in the space spanned by the eigenvectors of the graph Laplacian L corresponding to A. Commute time distance has been used previously to capture structure in graphs in clustering [23] and for anomaly detection in static graphs [14].

Denote the commute time distance with respect to each of the adjacency matrices $A_t, t = \{1, .., T\}$ by $c_t(.,.)$. In this paper, we choose

$$d_t(.,.) := c_t(.,.).$$

We note that there exist several other ways to determine distances between nodes in a graph, including shortest path, alternative distance measures based on random walks and others [8, 9]. While any of these distance measures can be used as an alternative to commute time distance, we chose to use commute time distance for the following two reasons. (i) Robustness - the fact that commute time is averaged over all paths (and not just the shortest path) makes it more robust to data perturbations, and (ii) scalability - there have been recent breakthroughs [15] in approximately computing the commute time with high accuracy using methods that scale to very large graphs, which we discuss next.

Computation of commute time distance for a single graph instance having n nodes using (3) is $O(n^3)$. Clearly, this is prohibitively expensive to use on large graphs with several thousands of nodes. In order to address this issue, Khoa and Chawla [15] have proposed an approximate commute time embedding method to compute the commute time with high accuracy in $O(kn)$ time, where k (referred to as k_{RP} in [15]) is the dimension of the approximate embedding, using near-linear time solvers for diagonally dominant systems of equations. Furthermore, they show that for the choice of $k = O(\log n/\epsilon^2)$, their method computes commute time distances accurately within error of the order of ϵ^2 (please refer to equation (5.2) in [15]). For this choice of $k = O(\log n/\epsilon^2)$, the run-time complexity of [15] is $O(n \log n)$.

3.2 CAD algorithm

The CAD algorithm is formally defined next. First, the commute time distances $c_t(i, j)$ are calculated for every pair of nodes $v_i, v_j \in V$ and every time instance $t = 1, .., T$ with embedding dimension k using [15]. Subsequently, the scores $\Delta E_t(.)$ are given by

$$\Delta E_t(e_{i,j}) = |A_{t+1}(i, j) - A_t(i, j)| \times |c_{t+1}(i, j) - c_t(i, j)|.$$

are computed for each $t = 1, \ldots, T$ and $i, j \in \{1, \ldots, n\}$. Anomalous edges E_t at each transition t to $t+1$ are detected using the scores $\Delta E_t(.)$ as described in Section 2.4.1, by simply sorting the set of values

$$\{\Delta E_t(e); \forall e \in E\}$$

and choosing E_t to be the smallest set of edges S (in terms of the cardinality $|S|$ of S) such that $\sum_{e \in E-S} \Delta E_t(e) < \delta$ for a threshold δ. Once the anomalous edges have been determined, the nodes V_t associated with these edges are declared as anomalous nodes. The pseudocode for the resulting CAD algorithm is formally stated in Algorithm 1.

CAD has two input parameters: approximation parameter k and threshold δ. The process for choosing k and δ in practice is described in Section 4.1.1.

3.3 Runtime of CAD

The computation of the commute time distance per graph instance is given by $O(n \log n)$ [15]. The number of non-zero entries in $\{\Delta E_t(e); \forall e \in E\}$ will be of the same order as the number of non-zero entries in $A_t, A_t + 1$, *i.e.* $O(m)$. This implies that the runtime for sorting the set of values $\{\Delta E_t(e); \forall e \in E\}$ and subsequently determining the smallest set of edges S such that $\sum_{e \in E-S} \Delta E_t(e) < \delta$ will require runtime of order $O(m \log m)$. The total runtime of CAD is clearly dominated by these two operations and is therefore given by $O(n \log n + m \log m)$.

For all large, sparse, real world graphs $m = O(n) \ll n^2$ [3]. Therefore, the overall running time for CAD is given by $O(n \log n + n \log n) = O(n \log n)$.

3.4 Comparison with related methods

We now describe the existing methods [20, 1, 21, 18, 12] in the context of the illustrative example and contrast them with CAD. Of these, all but [1] are designed only to detect whether a graph transition is anomalous or not, and do not address the problem of localizing anomalous edges and nodes.

The activity vector method (henceforth referred to as ACT) proposed by Ide and Kashima [12], is based on eigenvectors of adjacency matrices. ACT identifies anomalies by comparing the activity vectors (first eigenvector of adjacency matrices) at successive graph instances. Akoglu and Faloutsos' [1] method (henceforth referred to as AFM) extend this work by first constructing dependency matrices based on correlations between local node features, and then applying ACT to these derived dependency matrices. As mentioned in the introduction section, AFM [1] identifies anomalous nodes by determining which nodes contribute the most to the anomaly score during that transition. The same strategy can be used to find anomalous nodes using ACT as well.

The drawback with using ACT to identify anomalous nodes is that all nodes which are affected by changes in structure contribute to the anomaly score and will be classified as anomalies, resulting in false alarms. On the other hand, AFM uses *local* features (average edge weight, degree etc.) based on the egonets (1-step neighborhoods) of each node, which do not necessarily differentiate between significant changes in graph structure with benign changes.

For example, consider the change in edge weight between nodes r_7 and r_8. When using AFM, the local features for r_7 and r_8 only change marginally, more or less to the same extent as the change in local features of nodes b_1, b_3 (due to decrease in edge weight between b_1 and b_3). As a result, when using AFM, several non-anomalous nodes (for *e.g.*, b_1, b_3) will be detected along with anomalous nodes (for *e.g.*, r_7, r_8), resulting in a large number of false alarms. Also, as a consequence of the change in edge weight between nodes r_7 and r_8, nodes r_4, r_6, r_8, r_9 become a loosely connected component with respect to the rest of the graph. As a result, when using ACT, all the nodes r_4, r_6, r_8, r_9 will contribute to the anomaly score during the transition from t to $t+1$ and will be classified as anomalies, once again resulting in false alarms. This is verified experimentally in the AFM paper [1] (in Section III.D and Section IV).

We note that the edges associated with nodes r_4, r_6, r_9 *remain the same*, and that the edge responsible for this change in graph structure is associated only with node r_8. In other words node r_8 is responsible for the significant structural change in the graph during this transition. A similar argument holds true for node r_7. The objective of CAD is to only identify anomalous nodes such as r_8, as opposed to identifying non-anomalous nodes that exhibit superficial changes in relationships (such as b_1, b_3) or are affected by change in structure (such as r_4, r_6, r_9).

Algorithm 1 CAD

Input: Vertex set V, edge set E, adjacency matrix sequence $A_t, t = 1, .., T$, threshold δ, embedding dimension k
Output: Anomalous edge sets E_t, node sets V_t
for t in {1, .., T-1} **do**
    Compute commute time distance $c_t(i, j)$ for every pair of nodes $v_i, v_j \in V$ with embedding dimension k using [15]
end for
for t in {1, .., T-1} **do**
    Compute $\Delta E_t(e_{(i,j)}) = |A_{t+1}(i, j) - A_t(i, j)| \times |c_{t+1}(i, j) - c_t(i, j)|$ for every pair of nodes $v_i, v_j \in V$
    Choose anomaly edge set E_t as the set of edges S with smallest cardinality $|S|$ such that $\sum_{e \in E-S} \Delta E_t(e) < \delta$
    for Edge $e_{(i,j)}$ in E_t **do**
        $v_i \cup v_j \rightarrow V_t$
    end for
    Output anomalous edges E_t, anomalous nodes V_t
end for

There are two other related methods to CAD: (i) ADJ, where $\Delta E_t(e_{(i,j)})$ is set to be $|A_{t+1}(i, j) - A_t(i, j)|$, and (ii) COM, where $\Delta E_t(e_{(i,j)})$ is set to be $|c_{t+1}(i, j) - c_t(i, j)|$. ADJ therefore only looks at difference in adjacency matrices, and COM only looks at differences in commute times. Observe that both ADJ and COM satisfy condition (2).

However, both of them will result in detection of benign edges as anomalies. This can be seen in the case of the toy example as follows. In the case of ADJ, the change in relationship between nodes b_2 and b_7 is benign, but will be classified as anomalous to the same extent as the edge between b_1, r_1 due to similar changes in edge weight. On the other hand, in the case of COM, while $\Delta E_t(r_7, r_8)$ will be large, $\Delta E_t(r_i, r_j)$ will also be large for all pairs of $i, j : i \in \{1, 2, 3, 5, 7\}, j \in \{4, 6, 8, 9\}$ because of the decreased edge strength between r_7 and r_8.

Unlike ADJ and COM, $\Delta E_t(b_2, b_7)$ will be smaller than $\Delta E_t(b_1, r_1)$ in the case of CAD because $|c_{t+1}(b_2, b_7) - c_t(b_2, b_7)|$ will be smaller relative to $|c_{t+1}(b_1, r_1) - c_t(b_1, r_i)|$ due to the tight coupling between b_2, b_7. Also, $\Delta E_t(r_i, r_j)$ will be small for all pairs of $i, j : i \in \{1, 2, 3, 5, 7\}, j \in \{4, 6, 8, 9\}$ except $i = 7, j = 8$ because $|A_{t+1}(i, j) - A_t(i, j)| = 0$ in these cases. CAD therefore should perform favorably compared to ADJ and COM. This is corroborated in Section 4.1 via experimental results.

3.5 Quantitative performance of CAD on toy example

Using the toy example in Figure 1, we describe how CAD determines anomalous nodes during the transition from time slice t to $t + 1$. Because of the small number of nodes ($n = 17$), we determine the exact commute time distance using (3), and subsequently calculate the anomaly scores $\Delta E_t(.)$.

Recall that the commute time distance $c_t(i, j)$ is equal [14] to the Euclidean distance in an embedded space given by the eigenvectors [4] of the Laplacian matrix L_t. Ignoring the trivial first eigenvector of all 1's, we plot the second (Fiedler vector) and third eigenvectors corresponding to the Laplacians of A_t and A_{t+1} in Figure 2. From equation (5) in [14], the commute time distance is approximately equal to the Euclidean distance in the 2-dimensional embedding in Figure 2. This implies that pairs of nodes with smaller commute time distances should be closer to each other in Figure 2, and pairs of nodes with larger commute time distances should be farther apart. In Figure 2(a), in agreement with our intuition, all the red nodes are close together, all the blue nodes are close together, and the blue and red nodes are reasonably well separated. In Figure 2(b), we observe that the nodes r_4, r_6, r_8, r_9 become distant from the rest of the nodes - this is in agreement with the structure change in the graph due to the weakening of the edge between r_7 and r_8. Also, nodes r_1 and b_1 are a lot closer in 2(b) compared to 2(a), due to the new edge between r_1 and b_1. Finally, nodes b_4 and b_5 are a lot closer due to the strengthening of the edge between them.

It follows from these observations that the anomaly scores $\Delta E_t(.)$ corresponding to (b_1, r_1) (b_4, b_5) and (r_7, r_8) should be large, and the rest of the anomaly scores should be small. This is indeed the case, as seen from Table 1 where the anomaly scores $\Delta E_t(.)$ for all the edges with non-zero scores are listed. For some suitable threshold δ, edges between nodes (b_1, r_1) (b_4, b_5) and (r_7, r_8) and corresponding nodes $b_1, b_4, b_5, r_1, r_7, r_8$ will be identified as anomalies.

Edge	b_1, r_1	b_4, b_5	r_7, r_8
$\Delta E_i(.)$	10.6	9.56	8.99
Edge	b_1, b_3	b_2, b_7	Rest
$\Delta E_i(.)$	0.1	0.22	0

Table 1: Table listing the values of $\Delta E_t(.)$ for edges in the illustrative example.

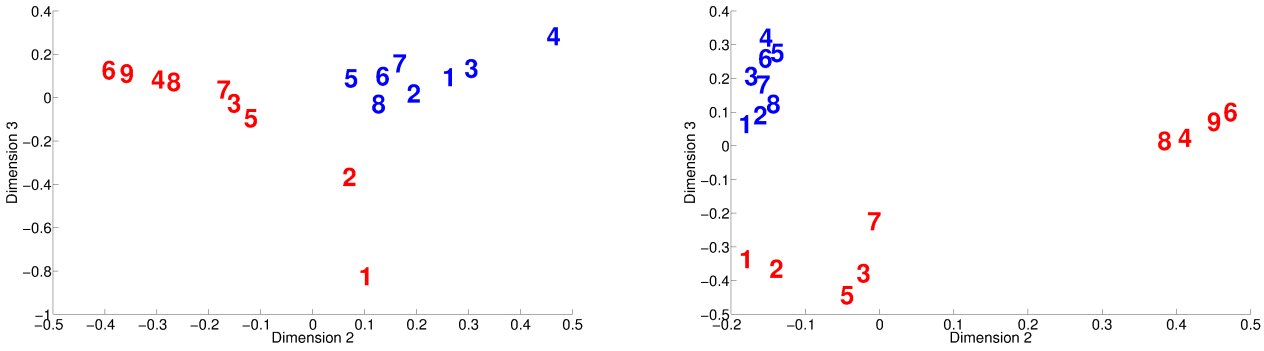
3.5.1 Comparison with ACT

Since the ACT method by Ide *et al.* [12] is closest to CAD in terms of being an eigenvector based anomaly detection technique, we use this method as a baseline for comparing our results. We point out that ACT takes graph sequences as inputs, computes the leading eigenvector (called activity vector) $a_t \in \mathbb{R}^n$ of the adjacency matrices A_t, and returns a decision as to whether the t to $t+1$ graph transition is anomalous based on an anomaly measure computed using the activity vectors. Although the anomaly detection method of Akoglu and Faloutsos (AFM) [1] is an extension of ACT, we do not use this method as a baseline. This is because, AFM, unlike ACT and CAD, does not base its analysis on the graph adjacency matrix, but on dependency matrices constructed from *local* features based on egonets of nodes in the graph. Also, depending on which local features are being used, AFM can return different sets of anomalies, making comparison of results difficult.

We apply the ACT algorithm to this data set, and com-

Node	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9
ΔN_t	10.5	0.30	0.20	9.56	9.56	0.00	0.30	0.00	10.29	0.00	0.00	0.00	0.00	0.00	8.99	8.99	0.00

Table 2: Table listing values of $\Delta N_t(.)$ for nodes in the illustrative example.



(a) Time t (b) Time $t+1$

Figure 2: Illustrative example: 2 dimensional Laplacian eigenmap embeddings corresponding to graph instances at time t, $t+1$. The colored numbers indicate the node locations for $r_1, .., r_9$ (labeled in red) and $b_1, .., b_8$ (labeled in blue).

pare the results of CAD and ACT in Figure 3. Using the procedure described in [1], ACT can be used to identify anomalous nodes with respect to the transition by comparing the summary r_t of the activity vector from $t-w+1$-th (w being the window size) to the t^{th} transition to the activity vector a_{t+1} of the $(t+1)^{th}$ instance. The anomaly scores being reported for ACT for each node v_i are given by the absolute difference $|a_{t+1}(i) - r_t(i)|$, with $w = 1$. For the purposes of comparing with ACT, we define anomaly scores with respect to CAD for nodes as follows: The anomaly score for each node $v_i \in V$ for a transition from t to $t+1$ is given by

$$\Delta N_t(i) \quad = \quad \sum_{j \in \{1,\ldots,n\}} \Delta E_t(e_{(i,j)}).$$

Anomaly scores $\Delta N_t(.)$ for all nodes are listed in Table 2.

For fair comparison, we normalize the anomaly scores from CAD and ACT by the respective maximum anomaly score. From Figure 3, it is clear that the anomaly scores $\Delta N_t(i)$ generated by CAD are significantly higher for the responsible nodes b_1, b_4, b_5, r_1, r_7, r_8 compared to the rest of the nodes. On the other hand, while the anomaly scores generated using ACT for the 4 nodes b_4, b_5, r_7, r_8 is higher compared to the rest, the scores for b_1, r_1 are small. Furthermore, the difference between the scores corresponding to these 4 nodes b_4, b_5, r_7, r_8 and the rest of the nodes is not as large as is the case with CAD. This comparison reiterates the observation in Section 3.4 that CAD only identifies responsible nodes, whereas ACT can also potentially identify nodes which are affected by the change in graph structure.

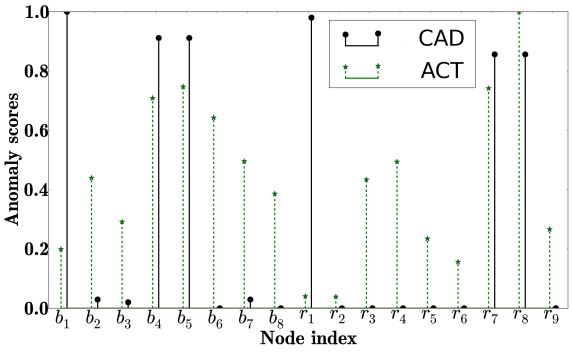


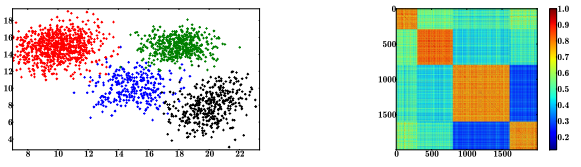
Figure 3: Normalized anomaly scores for CAD and ACT on toy data set. CAD identifies only responsible nodes, whereas ACT also assigns significant scores to nodes affected by change in graph structure.

4. EXPERIMENTS

CAD is applied to several synthetic and real world data sets in order to localize anomalous edges and nodes responsible for significant changes in graph structure over time. Since the real data sets only provide anecdotal evidence of anomalies, we create a synthetic data set from Gaussian mixtures to quantitatively analyze the performance of CAD, and to compare against ACT. The three real world data sets on which we run CAD are (i) the Enron email network data[1], (ii) the DBLP co-authorship network data[2], and (iii) a world-wide precipitation network data[3]. The results on these data sets are presented as a proxy for our results on a proprietary organizational email network data, collected from a corporation for experimental purposes for insider threat detection.

We compare the performance of CAD against several different algorithms. In addition to ACT, we also compare the performance of two variants of ACT - the ADJ and COM methods discussed in Section 3.4. Recall that ADJ

[1] `http://www.isi.edu/~adibi/Enron/Enron.htm`
[2] `http://dblp.org/xml/`
[3] `thttp://www.cdc.noaa.gov/data/gridded/data.ncep.reanalysis.html`



(a) (b)

Figure 4: Random realizations from 4-component Gaussian mixture, and the corresponding adjacency matrix.

only looks at difference in adjacency matrices, and COM only looks at differences in commute times. Finally, given the commonplace nature of node centrality measures [5], we also evaluate the performance of centrality for the purpose of determining anomalous nodes. In particular, let $cc_t(i)$ denote the closeness centrality measure of nodes v_i with respect to graphs G_t. The anomaly scores for the closeness centrality (CLC) method for each node v_i are given by $|cc_{t+1}(i) - cc_t(i)|$. The measures $|cc_2(i) - cc_1(i)|$ are then thresholded to identify anomalous nodes.

Experimental setup: CAD, as well as all algorithms compared as baselines, have been implemented in python and run on a 64-bit 2.327 GHz dual quad core Dell Precision T7500 desktop running Red Hat Enterprise Linux version 6.0 having 32 GB of physical memory.

4.1 Quantitative analysis: synthetic data

We perform experiments to evaluate the following characteristics of CAD: (i) robustness to parameter selection, (ii) accuracy, and (iii)scalability using synthetic data generated as follows. We draw 2000 random samples from a 2-dimensional Gaussian mixture distribution with 4 components. Figure 4a shows this data set with data from each component represented by a different color. For any pair of points i, j in this sample, we compute the Euclidean distance $d(i, j)$ between them. We then construct an adjacency matrix P where $P(i, j) = \exp(-d(i, j))$. The adjacency matrix corresponding to the set of realizations in Figure 4a is shown in Figure 4b. The graph corresponding to this adjacency matrix constitutes nodes belonging to 4 different clusters, with strong intracluster edges and weaker inter-cluster edges. We perturb this adjacency matrix P by adding a small amount of random noise to the data, and compute the adjacency matrix Q in an identical manner to P. We also construct a random matrix $R \in \mathbb{R}^{2000\times2000}$, where each entry $R(i, j)$ in this random matrix is given by

$$R(i,j) = \begin{cases} 0 & \text{with probability } p = 0.95 \\ \text{u(i,j)} & \text{with probability } p = 0.05, \end{cases}$$

where $u(i, j)$ is a random number drawn uniformly between 0 and 1. We then consider a dynamic graph sequence with two temporal instances $\{A_t, t = 1, 2\}$, with $A_1 = P$ and $A_2 = Q + (R + R')/2$.

In the transition from A_1 to A_2, we keep track of all edges for which $R(i, j) \neq 0$ such that i, j belong to different clusters. We consider these edges and the associated nodes to be anomalous because these edges establish ties between nodes belonging to different clusters, thereby contributing to anomalous change in graph structure.

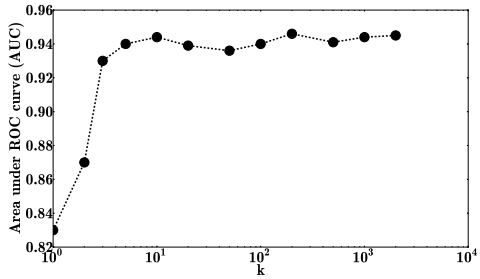


Figure 5: Variation of AUC against k. From the figure, it is clear that the performance of CAD is invariant to the choice of k for values of $k > 10$.

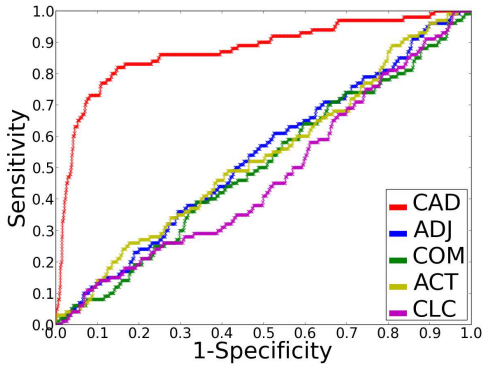


Figure 6: ROC curves comparing the 4 methods: CAD, ACT, COM and ADJ. The performance of COM and ADJ are close to the random baseline, and the performance of ACT is marginally better. CAD outperforms the other methods by a significant margin.

4.1.1 Parameter selection

There are 2 user-given parameters for CAD: δ and k. Clearly, the anomaly scores $\Delta E_t(.)$ returned by CAD are independent of the choice of δ and do not affect the ROC performance of CAD. The approximation parameter k for commute time distance computation, however, affects the accuracy of the anomaly scores. For the synthetic data, we run CAD with various different choices of k, compute the area under the ROC curve (AUC) as before, and plot the variation of AUC against k in Figure 5. It is clear from the figure that the performance of CAD is invariant to the choice of k for values of $k > 10$, for the synthetic data.

4.1.2 Accuracy

We generate 100 sets of realizations of this synthetic data. For each instance, we run CAD (with $k = 50$) and determine the anomaly scores $\Delta E_1(.)$ for all pairs of nodes. We then determine the anomalous node and edge sets E_1 and V_1 for a range of different values δ using Algorithm 1. The accuracy of CAD can be evaluated either by contrasting E_1 against the ground truth set of anomalous edges, or by contrasting V_1 against the ground truth set of anomalous nodes. Qualitatively, the performance in terms of either edges or nodes should be fairly similar and therefore either can be chosen for the sake of evaluation. Because we intend to compare the performance of CAD against ACT, we chose to evaluate

accuracy in terms of nodes instead of edges.

In order to compare the performance of the five methods - CAD, COM, ADJ, ACT and CLC, ROC curves are generated for each of these methods by sweeping over different values of the threshold δ and comparing the resulting anomalous node sets against ground truth. The ROC curves for these five algorithms averaged over the 100 sets of realizations of synthetic data is shown in Figure 6. From this figure, we note that performance of ADJ, COM, ACT and CLC is close to the random baseline. This is in sharp contrast to the performance of CAD, which is significantly better. The area under the ROC curves for CAD, ADJ, COM, ACT and CLC are respectively given by 0.88, 0.53, 0.51, 0.53 and 0.49. It is interesting to note that suitably combining the adjacency matrix information in ADJ and commute time information in COM together in CAD results in significant increase in anomaly detection performance.

4.1.3 Scalability

We generate symmetric random graphs of varying sizes to study the scalability of CAD and the different alternatives. Based on our observation in Section 4.1.1, we select with $k = 10$ wherever relevant. Since most real-world graphs are sparse with $m = O(n)$ [3], we fix the sparsity level at $1/n$ for this study. The results indicate that CAD is able to process synthetically generated random graphs having $n = O(10^7)$ nodes and $m = O(10^7)$ edges in 5 minutes on average over 10 trials. Since COM relies on the similar computations, the running time of COM is comparable with CAD. ACT only has to compute the largest eigen vector and has a significantly low computation time of close to 1 minute for graphs having $n = O(10^7)$ nodes and $m = O(10^7)$. Since ADJ computes the differences in sparse adjacency matrices, the running time of ADJ is the lowest at an average of 10 seconds. For the different graph size, the CLC method takes approximately one third the time taken by CAD. However, the running time of CLC is highest affected by the graph sparsity, and even for modestly higher sparsity such as $m = 10n$, it performs worse than all the alternatives. In summary, CAD can comfortably operate on very large graphs with runtimes comparable to its alternatives, while providing the highest accuracy for detecting anomalous changes.

4.2 Qualitative analysis: real data

We run CAD on 3 real-world data sets and verify the identified anomalies based on anecdotal evidence.

Choice of approximation parameter k: During analysis on real-world data sets, similar to our observations with the synthetic data, we observed that we get a very consistent set of anomalies using CAD for any choice of $k > 10$. Based on these observations and also those presented in [15], we conclude that the results are largely robust to the choice of k and we use $k = 50$ in all our experiments.

Threshold (δ) selection: We automate the selection of the threshold δ as follows. Let $l > 0$ be the average number of anomalies the user is interested in per graph instance. Then, δ is chosen such that the total number of anomalous nodes given by $\sum_{t=1}^{T-1} |V_t| = l(T-1)$. This selection of the threshold δ ensures that l anomalous nodes are selected on average for each transition t to $t+1$. By selecting a single threshold for all $t = 1, .., T-1$ rather than selecting the top l anomalies for each transition t to $t+1$, we ensure that no anomalies are reported for transitions when there is no anomalous activity, and similarly ensure that more than l anomalies are selected for transitions where anomalies are present. We note that while this procedure for threshold selection needs to be done offline, the procedure can be suitably modified in an online setting by aggregating scores up to the current graph instance and updating the threshold.

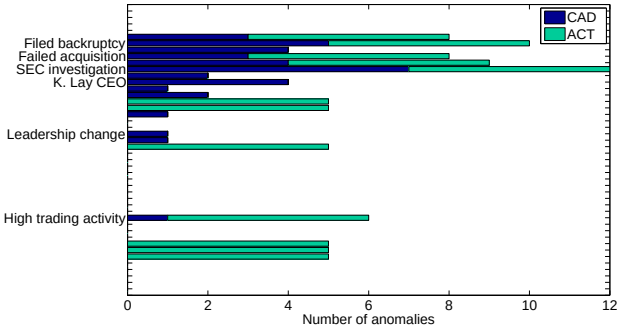


Figure 7: Bar plot of anomalous graph instances detected by CAD (blue) and ACT(green) against the Enron scandal timeline.

ACT results: We determine anomalies using ACT in the following manner. We use ACT to determine which of the transitions are anomalous. For each such anomalous transition, we declare the top 5 nodes with the highest, non-zero anomaly scores $|a_{t+1}(i) - r_t(i)|$ to be anomalous. This is identical to the presentation of results in [1] (Figure 7).

4.2.1 Enron email network data

The Enron email network data is a sparse graph constructed based on emails exchanged among the 151 employees of Enron Corporation between the months of December 1998 and November 2002. This graph data set containing 151 nodes and approximately 300 edges for the most dense graph instances, has been used in the past for evaluating anomaly detection algorithms [22]. We aggregate the data on a monthly basis leading to 48 monthly graph instances represented as 151×151 symmetric adjacency matrices, where the edge weights in each matrix indicate the number of times emails are exchanged between any two employees in the corresponding month. We run CAD and ACT on this data, with threshold δ corresponding to $l = 5$ for CAD and $w = 3$ for ACT. For ACT, we report the top 5 anomalies. Since the Enron data set only has 151 nodes, we did not need the approximation in commute distance calculation. The algorithm finished processing each graph instance in a few seconds even using the exact calculation given by (3) for commute distance. With the hypothesis that emailing pattern within the organization was influenced by the scandal, we verify whether the anomalous edges and nodes detected during the different times overlap with the list of key players and correspond to the scandal timeline.

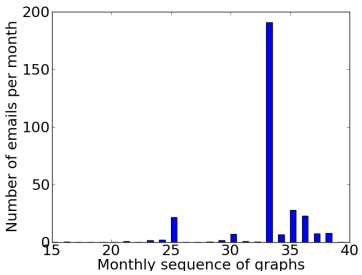
Figure 7 shows a timeline of the Enron scandal and the anomalous graph transitions found by CAD (in blue) and ACT(in green) as a stacked bar plot. Presence of bars indicate that those graph transitions have been marked anomalous by the respective algorithms due to the presence of anomalous relationship changes. The length of the bars indicate the number of identified anomalous nodes, dependent on the choice of δ. It can be seen that both ACT and CAD

find graph transitions corresponding to important scandal events to be anomalous. The first 23 months in the data set from Dec 1998 to Sept 2000, is a period of calm in the organizational email network, with ACT reporting 4 and CAD reporting 1 graph transition as anomalous. The common anomalous graph transition (transition number 12, Oct 1999 - Nov 1999) is when one of the traders, Chris Germany[4] starts interacting with a lot of other traders within the company, probably indicating a sudden increase in trading activity within the organization. The portion of data corresponding to Mar 2002 to Nov 2002 is also a calm period corresponding to the end of the scandal. The period of turmoil for Enron was from Feb 2001 to Feb 2002 (graph instances 27-39), with Jeff Skilling being hired as CEO in Feb 2001, followed by questionable reporting of earnings in the next quarter, leading to unstable stock prices. CAD has identified 10 transitions to be anomalous during the 12 month of the scandal, whereas ACT has identified 6, all of which overlap with CAD.

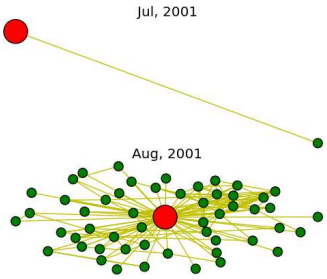
We found that Kenneth Lay is involved in the highest number of anomalous edges in the anomalous edge set E_{32}, which corresponds to the transition from Jul to Aug 2001 (transition between graph instances 32 and 33). This coincides with the time when Kenneth Lay was reappointed as CEO of Enron and the scandal was starting to get press attention. Figure 8a shows the histogram of email communication to and from Kenneth Lay over the entire 44 month period. The histogram shows a huge spike in his volume of emails in the 33rd month. Figure 8b is the subgraph for Kenneth Lay (the red node in the center) for months 32 and 33. It is observed that he started interacting with many people belonging to different job roles within the company during graph instance 33. Therefore, it is justifiable that Kenneth Lay is picked up as an anomaly during that transition. For the top 5 anomalies we have reported for ACT, Kenneth Lay has never been identified as an anomaly in any transition. During the same transition (32 - 33), the top anomaly found by ACT is James Steffes, VP of government affairs. However, on careful examination of his connections during this transition, it is observed that there is more than 50% overlap in his connections. Also, the new connections are similar to the existing ones in terms of job roles. Steffes was sending few emails to these people in July 2001, whereas in Aug the volume of emails multiplied by many folds. The biggest change in his network is therefore in terms of volume. This change is therefore not as important in terms of anomalous behavior causing structural change in the network, as the Kenneth Lay example.

Additionally, CAD has successfully found Rosalie Fleming (assistant to Kenneth Lay) as an anomaly just before Jeff Skilling took over as CEO from Kenneth Lay during Dec 2000. David Delainey, CEO of Enron Energy, has also been tagged as an anomaly by CAD during the Oct 2001-Nov 2001 transition. On further investigation, we found that this was due to his involvement in the Dynergy acquisition that was being planned in Oct 2001. During the period of Nov 2001 to Feb 2002, most anomalous edges found in E_t belong to Enron employees having job roles of legal specialists, president and vice presidents, and traders. This is expected, since it was during this time that the acquisition failed and Enron declared bankruptcy. In summary, CAD not only identifies the transition periods that are most tumultuous in the Enron scandal timeline, but, unlike ACT, also localizes the key players of the scandal with high level of accuracy.

the anomalies identified by CAD, unlike ACT, correlate strongly with the timeline of the Enron scandal and the key players involved in it.



(a) Histogram of emails sent/received by Kenneth Lay during the 44 months.



(b) Subgraph showing K. Lay during July and Aug 2001.

Figure 8: Email pattern of ex-CEO Kenneth Lay of Enron.

4.2.2 *DBLP data*

The DBLP data set is a collaboration network from the scientific community in which each node represents an author and the edge weight between two authors indicates how many papers they have coauthored every year. We filter the data to consider 6574 authors who published at least two papers every year from 2005 to 2010. The sparse graphs constructed for each of the time instances had approximately 30K edges. We ran CAD on this data set (with threshold δ corresponding to $l = 20$) in order to discover authors who established collaborations in new research areas. On average, CAD processed each graph instance in 40 seconds for $k = 50$. We uncovered several interesting anomalies, but for sake of brevity, we report three specific examples:

(i) During the 2005-06 transition, author Atanas Rountev was involved in the most number of anomalous edges returned in E_t. Among these, the edge with the largest score $\Delta E_t(.)$ was between Atanas Rountev and P. Sadayappan. On further investigation it turns out that Atanas Rountev's research focus till 2005 was software engineering, but in 2006 he started collaborating with P. Sadayappan, whose focus is high performance computing, and published multiple papers in several high performance computing conferences.

(ii) During the same 2005-06 transition, the anomalous edge set E_t returned edges connecting Salvatore Orlando with the authors Francesco Bonchi and Fosca Giannotti. On further examination, we found that prior to 2006, Salvatore Orlando researched performance aspects of databases. In

[4] `http://www.ids.cs.columbia.edu/sites/default/files/hierarchyv3.pdf`

2006, he started collaborating with Francesco Bonchi and Fosca Giannotti, and started publishing in core database conferences. We note that anomaly scores $\Delta E_t(.)$ in this case are lower compared to the scores for Atanas Rountev, which agrees with the fact that the switch from performance aspects of databases to core databases is less severe compared to the switch from software engineering to high performance computing.

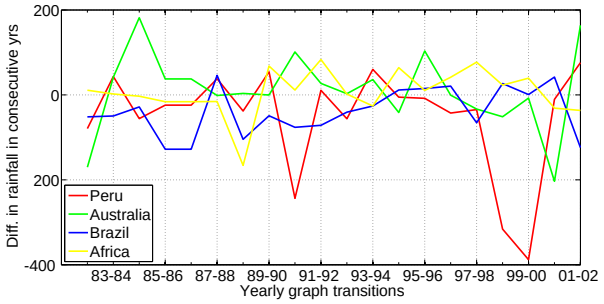


Figure 10: Difference in consecutive years' rainfall in January from 1982-02.

(iii) The third example concerns the severed relationship between Oliver Brdiczka and Max Mühlhäuser during the transition 2008-09. It can be explained by the fact that Oliver and Max published several papers in 2008 as colleagues at Technische Universität Darmstadt. However, in 2009, Oliver switched to publishing in human computer interaction conferences which coincided with his move to the Palo Alto Research Center.

4.2.3 *World-wide precipitation data*

The world-wide precipitation (rainfall) data consists of monthly averages of observations between 1982 and 2002 recorded at 0.5° resolution on the entire earth's grid. Precipitation recordings are only available on land surface and there are 67,420 such data locations for every month. Precipitation has a seasonal and spatial variation that is reasonably stable over time. Abnormal weather patterns can therefore be uncovered by detecting abnormal variations in precipitation data. In order to determine such abnormal variations, we construct 10-nearest neighbor graphs for each month over the 67,420 locations, where the edge strength between a node i and one of its 10 neighbors j is given by $\exp\left(-\|p_i - p_j\|^2/2\sigma^2\right)$ with p_i and p_j corresponding to precipitation levels recorded at i and j for that month and σ is the kernel bandwidth. In order to account for seasonal trends, we at a time restrict our attention to the 21 graphs corresponding to one particular month of the year, and apply CAD to each of the 12 sequences of 21 graphs each. To run CAD with $k = 50$ on each graph instance (with threshold δ corresponding to $l = 30$), the average running time was bounded by 2 minutes.

We discovered several interesting anomalies upon running CAD on this data set. In the interest of space, we present one set of interesting anomalies discovered by CAD for the month of January during the transition from 1994 to 1995. The top anomalous edges reported by CAD during the transition from January, 1994 to January, 1995 were those between southern Africa and eastern equatorial Africa, Brazil and eastern equatorial Africa, Peru and the Amazon basin, Peru and the southeast Asian island of Malaysia, and between Australian inland and equatorial Africa. Figure 9 shows these location pairs connected by these anomalous edges (in blue) using red squares and yellow circles. Analyzing the data it can be observed that locations in southern Africa and Brazil received much higher rainfall during that year compared to the past year, whereas Peru and Australia received less rain than the year before. During the same transition, rainfall in equatorial Africa as well as the Amazon basin did not change. Therefore, the edge strength (similarity value computed using the exponential function described above) between these location pairs changed compared to the previous year. On verification using external evidence, we found that the La Niña[5] weather pattern occurred in 1995, which is characterized by wetter winters in southern Africa and Brazil, very high rainfall in southeast Asian islands, and drought-like conditions in Peru, Chile, and eastern equatorial Africa. Therefore, we can easily verify that the top anomalies found by CAD except Australia can be explained by the La Niña phenomenon. During the same time, the Australian plains suffered one of the worst droughts since 1900[6] and therefore, experienced much lower rainfall than normal, which is why it became closer to drier regions like the African plains. These evidences corroborate the fact that the anomalous relationships identified by CAD are indeed valid anomalies.

The *difference* in average precipitation experienced by southern Africa, Brazil, Peru and Australia between successive years from 1981 to 2012 during the month of January is shown in Figure 10. From this figure, we note that locations in southern Africa and Brazil were found to have received higher rainfall during 1995 as compared to 1994, whereas Peru and Australia received less rain than usual. However, observe that increase and decrease in rainfall in these locations, as shown in Figure 10, is very subtle relative to some of the other variations seen in the figure. It is therefore unlikely that these anomalies would have been detected using time series analysis on the average precipitation data. The reason CAD was able to detect this anomalous weather pattern was because it resulted in simultaneous, albeit subtle changes in precipitation across several locations, which in turn resulted in significant changes in the structure of the nearest neighbor graphs. This ability of CAD in picking up subtle but simultaneous changes in precipitation across several locations therefore makes CAD useful for identifying teleconnections[7], *i.e.* climate anomalies related to each other at long distances.

5. CONCLUSION

A novel method called CAD is proposed for localizing abnormal changes in nodes relationships (edges) that are responsible for anomalous change in structure in weighted dynamic graphs. Existing anomaly detection methods for dynamic graphs such as ACT are primarily designed only to detect whether a graph structure change is anomalous or not, and are not well suited for identifying the responsible edges and nodes. This makes CAD, to the best of our knowledge, the first of its kind for localizing anomalies in

[5] https://en.wikipedia.org/wiki/El_Nino
[6] http://en.wikipedia.org/wiki/2000s_Australian_drought
[7] http://en.wikipedia.org/wiki/Teleconnection

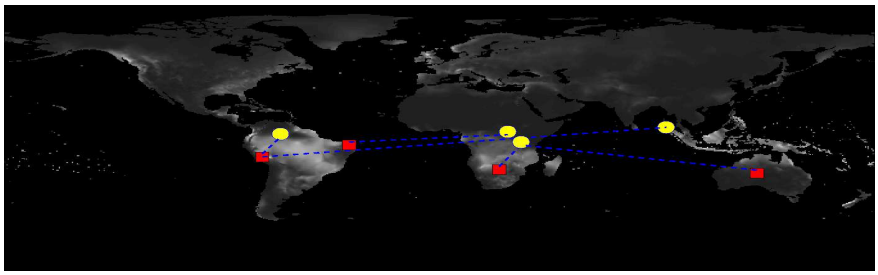


Figure 9: Heat map of rainfall for January 1995. Red squares and yellow circles are nodes associated with anomalous edges (indicated by blue dotted lines) found by CAD.

time-evolving graphs. CAD tracks changes in edge strength and structure (via commute time distance) in order to determine these anomalies. CAD has an $O(n \log n)$ run-time complexity per graph instance for large, sparse graphs, making it scalable. Our experimental studies on synthetic and large real world datasets showed that CAD consistently and efficiently localizes anomalous edges and associated nodes responsible for anomalous changes in graph structure.

6. ACKNOWLEDGEMENT

This research has been funded by the DARPA/ADAMS program under contract W911NF-11-C-0216. The authors would like to thank Prof. Stephen Boyd at Stanford University for initial discussions on this work and Prof. Sanjay Chawla and Nguyen Lu Dang Khoa at University of Sydney for providing code for the spielman-teng solver. Finally, the authors would like to thank the reviewers for their very detailed reviews and comments.

7. REFERENCES

[1] L. Akoglu and C. Faloutsos. Event detection in time series of mobile communication graphs. In *Army Science Conference*, 2010.

[2] L. Akoglu, M. McGlohon, and C. Faloutsos. Oddball: Spotting anomalies in weighted graphs. *Adv. in Know. Discovery and Data Mining*, pages 410–421, 2010.

[3] J. Batson, D. Spielman, N. Srivastava, and S. Teng. Spectral sparsification of graphs: theory and algorithms. *Comm. of the ACM*, 56(8):87–94, 2013.

[4] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in NIPS*, 14:585–591, 2001.

[5] S. P. Borgatti. Centrality and network flow. *Social Networks*, 27(1):55–71, 2005.

[6] H. Bunke, P. Dickinson, A. Humm, C. Irniger, and M. Kraetzl. Computer network monitoring and abnormal event detection using graph matching and multidimensional scaling. In *Adv. in Data Mining Applns in Medicine, Web Mining, Marketing, Image and Signal Mining*, pages 576–590. 2006.

[7] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58, 2009.

[8] P. Chebotarev and E. Shamis. On proximity measures for graph vertices. *arXiv preprint math/0602073*, 2006.

[9] J. Chen and I. Safro. A measure of the local connectivity between graph vertices. In *Proc. of Int. Conf. on Computational Science*, pages 196–205, 2011.

[10] W. Eberle and L. Holder. Discovering structural anomalies in graph-based data. In *ICDM Workshops*, pages 393–398, 2007.

[11] X. Gao, B. Xiao, D. Tao, and X. Li. A survey of graph edit distance. *Pattern Analysis and Applications.*, 13(1):113–129, 2010.

[12] T. Ide and H. Kashima. Eigenspace-based anomaly detection in computer systems. In *ACM SIGKDD*, pages 440–449, 2004.

[13] I. Jovanović and Z. Stanić. Spectral distances of graphs. *Linear Algebra and its Applications*, 436(5):1425–1435, 2012.

[14] N. Khoa and S. Chawla. Robust outlier detection using commute time and eigenspace embedding. In *Adv. in Knowledge Discovery and Data Mining*, pages 422–434. 2010.

[15] N. Khoa and S. Chawla. Large scale spectral clustering using resistance distance and spielman-teng solvers. In *Discovery Science*, pages 7–21, 2012.

[16] M. Maybury, P. Chase, B. Cheikes, D. Brackney, S. Matzner, B. Wood, T. Longstaff, T. Hetherington, C. Sibley, J. Marin, L. Spitzner, J. Copeland, S. Lewandowski, and J. Haile. Analysis and detection of malicious insiders. In *Conf. on Intell. Anal.*, 2005.

[17] C. Noble and D. Cook. Graph-based anomaly detection. In *ACM SIGKDD*, pages 631–636, 2003.

[18] B. Pincombe. Anomaly detection in time series of graphs using arma processes. *ASOR Bulletin*, 24, 2005.

[19] P. Shoubridge, M. Kraetzl, W. Wallis, and H. Bunke. Detection of abnormal change in a time series of graphs. *Journal of Interconnection Networks*, 3:85–101, 2002.

[20] J. Sun, C. Faloutsos, S. Papadimitriou, and P. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *ACM SIGKDD*, pages 687–696, 2007.

[21] J. Sun, Y. Xie, H. Zhang, and C. Faloutsos. Less is more: Compact matrix decomposition for large sparse graphs. Technical report, CS Dept., CMU, 2007.

[22] B. Thompson and T. Eliassi-rad. Dapa-v10: Discovery and analysis of patterns and anomalies in volatile time-evolving networks. In *Workshop on Information in Networks*, 2009.

[23] L. Yen, D. Vanvyve, F. Wouters, F. Fouss, M. Verleysen, and M. Saerens. Clustering using a random walk based distance measure. In *Proc. of ESANN*, pages 317–324, 2005.