2014 IEEE 34th International Conference on Distributed Computing Systems

Sim-Watchdog: Leveraging Temporal Similarity for Anomaly Detection in Dynamic Graphs

Guanhua Yan Stephan Eidenbenz

Los Alamos National Laboratory
{ghyan, eidenben}@lanl.gov

***Abstract*—Graphs are widely used to characterize relationships or information flows among entities in large networks or distributed systems. In this work, we propose a systematic framework that leverages temporal similarity inherent in dynamic graphs for anomaly detection. This framework relies on the Neyman-Pearson criterion to choose similarity measures with high discriminative power for online anomaly detection in dynamic graphs. We formulate the problem rigorously, and after establishing its inapproximibility result, we develop a greedy algorithm for similarity measure selection. We apply this framework to dynamic graphs generated from email communications among thousands of employees in a large research institution and demonstrate that it works effectively on a set of more than 100 candidate graph similarity measures.**

I. Introduction

Graphs are widely used to characterize relationships or information flows among entities in large networks or distributed systems, including but not limited to online social networks, Internet topology, world wide web, and email traffic. Due to ubiquity of graph-modeled data from such systems, finding anomalies in these data has become an important problem. In this study, we consider the general problem of anomaly detection on graph-modeled data in a dynamic environment, where graphs change and evolve constantly over time. Suppose that a network administrator wants to find whether an insider attack occurs in a computer network on a particular day, given the history of communications among the machines in this network. This real-world challenge can be cast into a problem of anomaly detection on temporal graphs by modeling daily computer communications as graphs, where vertices represent computer hosts and edges the communications among them.

Networks or distributed systems evolve according to certain laws or protocols, suggesting that their underlying graphs do not appear in a uniformly random fashion. For instance, communication traffic among computer hosts exhibits strong self-similar property [16], and there is thus long-range temporal dependence among communication traffic. Also, social networks [8, 18] and web graphs [7]) tend to grow in a scale-free manner. Hence, when we use a sequence of graphs to represent the dynamics of these systems, we expect that there are inherent similarities among these graphs, particularly those that are temporally close.

We propose to leverage such temporal similarity in dynamic graphs for the purpose of anomaly detection. Our intuition is that, given the observed similarity level among temporally close graphs, if there exists a graph with similar measurements that deviate significantly from their expected values relative to its temporal context, it is likely that anomalous events – sometimes malicious activities – occur in the system under study. Consider, for instance, a simple computer system with two computer hosts, and a potential adversary between them manipulating their communications. If on a certain day, the communications between the two hosts are drastically different from what have been observed on the previous day, it is likely that an adversary unaware of the inherent similarity is tampering with the communications. At the other extreme, if the communications between the two hosts are much more similar than expected, it is possible that the adversary is launching a replication attack.

There are, however, many measures that can be used to evaluate the similarity between two graphs. The challenge remains that given a certain network or distributed system, we are unsure of which similarity measures should be used to characterize its inherent similarity. In this work, we develop a systematic framework that automatically chooses those similarity measures useful for anomaly detection in dynamic graphs. Such a selection process is guided by knowledge of samples previously identified to be normal or anomalous. In a computer network, for instance, we can use data gleaned during those periods with known attacks or failures as anomalies and others as normal samples. With a set of selected similarity measures, we further build a classifier that detects anomalous graphs in an online fashion.

Our main contributions are summarized as follows. **(1)** We design and implement *Sim-Watchdog*, a systematic framework that leverages temporal similarity inherent in dynamic graphs representing real-world networks or distributed systems for anomaly detection. Applying the Neyman-Pearson criterion [25, 32], Sim-Watchdog chooses a subset of similarity measures that maximize the detection rate while ensuring that the false positive rate is below a certain threshold. Using the selected similarity measures, Sim-Watchdog is able to identify anomalous graphs in an online fashion with low computational overhead. **(2)** We formulate the problem of selecting similarity measures under the Neyman-Pearson

1063-6927/14 $31.00 © 2014 IEEE
DOI 10.1109/ICDCS.2014.24

criterion as a discrete optimization problem, and theoretically establish its inapproximibility result. **(3)** We apply Sim-Watchdog to email communication records among thousands of employees in a large research institution, and demonstrate its performances with similarity measures chosen from more than 100 candidates.

The remainder of the paper is organized as follows. Section II presents related work. In Section III, we formulate the problem of identifying anomalous samples from dynamic graphs. In Section IV, we introduce the Sim-Watchdog framework for anomaly detection from dynamic graphs, and discuss its implementation in Section V. We experimentally evaluate the performance of Sim-Watchdog in Section VI. Finally, we draw concluding remarks in Section VII.

II. Related Work

There have been a large body of previous works on detecting anomalous activities in networks and distributed systems (e.g., [26, 12, 2, 15, 30, 31, 29]), and a few surveys on this topic were given in [10, 24, 33]. Our work differs from these previous efforts as we only consider the problem of anomaly detection based on data collected from these systems that can be represented as dynamic graphs. As graphs are a generic model of representing relationships among entities, our proposed framework can be applied to anomaly detection in a diversity of other domains.

Our work differs from existing works on identifying anomalous subgraphs or patterns in static graphs [21, 9, 5, 1, 19], as static graphs cannot fully characterize the dynamic nature of network data. Anomaly detection for dynamic graphs has been investigated in a few previous efforts. In [3], Bilgin and Yener surveyed a few directions for anomaly detection in dynamic graphs, including time series analysis of graph data, anomaly detection using minimum description length, window-based approaches, and methods based on vertex/edge properties. Neil *et al.* applied scan statistics on dynamic graphs to identify anomalous subgraphs [20]. In [23], Park *et al.* proposed an adaptive weighting scheme to combine various graph features for anomaly detection in time-series of graphs. In the work by Papadimitriou *et al.* [22], they propose to use graph similarity to detect anomalous web structures. In their work, however, they do not address the issue of selecting effective similarity measures. After examining more than a hundred graph similarity measures, our study reveals that only a small number of them are needed to predict anomalous graph structures with high accuracy. Hence, it is important to find such similarity measures with high predictive power for practical use. To the best of our knowledge, our work is the first of its kind that provides a systematic framework for anomaly detection in dynamic graphs based on temporal similarity.

There have also been some efforts dedicated to graph classification. For instance, Ketkar *et al.* compared the performances of several graph classification algorithms in [14]. In [17], Li *et al.* suggest that graph classification can be performed on feature vectors constructed based on the global topological and label attributes from graphs. In [11], Fay *et al.* propose to discriminate graphs through their spectral projections. The crucial differences between graph classification and our work here are two fold. Firstly, in these graph classification tasks, the graphs are static and therefore, there are no temporal correlations among them. Secondly, the goal of anomaly detection from dynamic graphs is to identify those samples that originate from separate physical processes from the normal one. The anomalous graph samples, however, may result from diverse causes such as operational hiccups, human errors, and malicious attacks in the system under study. It is questionable to apply graph classification to anomaly detection simply by grouping anomalous graph samples together into individual classes, as although these samples are distinguishable from the normal ones, they may scatter at different locations in the entire sample space.

III. Problem Formulation

In this work, we consider anomaly detection on data collected from networks or distributed systems that can be abstracted as attributed directed graphs (ADGs), where nodes represent entities and edges the relationships or information flows among them. A node or edge is associated with various attributes capturing its characteristics. Consider a sequence of ADGs, $\{G_t(V_t, E_t)\}_{t=1,2,...,n}$, where $G_t(V_t, E_t)$ is the observed ADG at time step t. Without loss of generality, we use $\mathcal{G}_i^j$, where $0 < i \leq j$, to denote the sequence of ADGs seen from time steps i to j, i.e., $\mathcal{G}_i^j = \{G_t(V_t, E_t)\}_{t=i,...,j}$. Assuming that the graphs are collected over a stable set of entities (e.g., email addresses, IP addresses, etc.) we have that $V_t = V$ for any t.

The binary label information regarding each ADG G_t where $t = 1, 2, ..., n$ is given by $l(G_t) \in \{0, 1\}$. If $l(G_t) = 1$, it means that G_t is normal; otherwise, G_t is deemed anomalous. The label information of each ADG is obtained from previously reported incidents or expected anomalies (e.g., those collected during network maintenances or when synthetic attacks are injected). We also have a number of similarity measures S available. Each similarity measure $s \in S$ estimates the similarity between any two ADG instances G_i and G_j as $s(G_i, G_j) \in \mathbb{R}$. Note that when calculating the similarity $s(G_i, G_j)$, we can use any or any subset of attribute values associated with the nodes and/or the edges in G_i and G_j.

Our goal is to find a classifier C, which, given a newly observed ADG sequence, $G_{n+1}, G_{n+2}, ..., G_{n+m}$, returns whether each of them G_{n+i}, where $1 \leq i \leq m$, should be 0 (normal) or 1 (anomalous) based on previously observed ADGs in $\mathcal{G}_1^{n+i-1}$. To find classifier C, we need to address two key questions: *What similarity measures should be used to evaluate the similarity between two ADGs, and how should we train classifier C based on the chosen similar-*

ity measures? Next, we introduce *Sim-Watchdog*, a novel similarity-based anomaly detection framework, to tackle these issues.

IV. The Sim-Watchdog Framework

The architecture of Sim-Watchdog is illustrated in Figure 1. The Sim-Watchdog framework involves a few steps. First, for each candidate similarity measure in S, we train an individual classifier from the ADGs in the training dataset $\mathcal{G}_1^n$. Given the classification results of each individual classifier on the training data, we select similarity measures from S with high discriminative power and form an ensemble of classifiers. Using this ensemble of classifiers, we further perform anomaly detection on the newly observed sequence of ADGs in $\mathcal{G}_{n+1}^{n+m}$, and report whether each of them is anomalous or not.

Before delving into the details of individual components of Sim-Watchdog, we want to emphasize that for any anomaly detection system to be practically useful, it is crucial that the system should have low false positive rates. For example, a widespread complaint against anomaly detection-based intrusion detection systems is that they often produce too many false alarms [4]. Sim-Watchdog relies on the *Neyman-Pearson criterion* [25, 32] to select similarity measures. In short, the Neyman-Pearson criterion aims to maximize the detection rate while ensuring that the false positive rate should be below a certain threshold. Hence, the Neyman-Pearson criterion provides the flexibility in controlling the false positive rate of the anomaly detection system, which is beneficial to its practical deployment.

A. Training Individual Classifiers

In the Sim-Watchdog framework, we build an individual classifier for each candidate similarity measure in S. Before presenting how to train an individual classifier based on a similarity measure s, we introduce a few notations. Define $\mathcal{A}_{i,j}^o$, where $j \geq i > 0$ and $o \geq 0$, as follows:

$$\mathcal{A}_{i,j}^o = \{(G_a, G_b), \forall a, b : {\scriptstyle i \leq a, b \leq j,\ b=a+o,\ l(G_b)\cdot l(G_a)=1}\}. \tag{1}$$

That is to say, $\mathcal{A}_{i,j}^o$ contains all nondecreasingly ordered pairs of *normal* ADGs in sequence $\mathcal{G}_i^j$ of order o, where here order o means that the gap between the time steps of each pair of ADGs is exactly o.

Given any set $\mathcal{H}$ of pairs of graphs, we define the following:

$$\begin{aligned} \mu_s(\mathcal{H}) &= \frac{1}{|\mathcal{H}|} \sum_{\forall (G_a, G_b) \in \mathcal{H}} s(G_a, G_b) \\ \sigma_s(\mathcal{H}) &= \left(\frac{1}{|\mathcal{H}| - 1} \sum_{\forall (G_a, G_b) \in \mathcal{H}} (s(G_a, G_b) - \mu_s(\mathcal{H}))^2\right)^{1/2} \end{aligned}$$

Hence, $\mu_s(\mathcal{A}_{i,j}^o)$ and $\sigma_s(\mathcal{A}_{i,j}^o)$ are the estimated mean and standard deviation of the similarity values among all pairs of ADGs in $\mathcal{A}_{i,j}^o$ according to similarity measure s, respectively.

Let $C_{s,\theta}^o$ be a parameterized classifier based on similarity measure s at order o with parameter θ. Consider any ADG $G_t \in \mathcal{G}_1^n$ and any order o. If G_{t-o} is labeled as 0 (anomalous), then the individual classifier of order o always classifies the new ADG G_t as 1 (normal). Otherwise, if the following holds:

$$|s(G_t, G_{t-o}) - \mu_s(\mathcal{A}_{1,t-1}^o)| > \theta \times \sigma_s(\mathcal{A}_{1,t-1}^o), \tag{2}$$

then $C_{s,\theta}^o(G_t) = 0$; otherwise, $C_{s,\theta}^o(G_t) = 1$.

With all these notations, we now discuss how to train parameter θ for the individual classifier at order o according to similarity measure s. It is easy to see that with only few samples, their mean and standard deviation do not offer a good baseline for anomaly detection. For instance, the second sample is always classified as 0 unless it is the same as the first one. Hence, when we evaluate the detection rate and the false positive rate of a classifier, we ignore the classification results of the first α samples, where $0 \leq \alpha \ll n$, and we assume that the first α samples should contain at least two sample ADGs labeled as 1 (normal) for each order $o = 1, 2, ..., o_{max}$ where o_{max} is the maximum order we consider.

Let $d_{s,o}(\theta)$ and $w_{s,o}(\theta)$ denote the *detection rate* and the *false positive rate* of the individual classifier at order o with parameter θ trained based on similarity measure s, respectively. Assuming that $o < \alpha$, we thus have:

$$\begin{aligned} d_{s,o}(\theta) &= \frac{\sum_{t=\alpha+1}^{n} \delta(l(G_t)=0)\cdot\delta(l(G_{t-o})=1)\cdot\delta(C_{s,\theta}^o(G_t)=0)}{\sum_{t=\alpha+1}^{n} \delta(l(G_t)=0)\cdot\delta(l(G_{t-o})=1)} \\ w_{s,o}(\theta) &= \frac{\sum_{t=\alpha+1}^{n} \delta(l(G_t)=1)\cdot\delta(l(G_{t-o})=1)\cdot\delta(C_{s,\theta}^o(G_t)=0)}{\sum_{t=\alpha+1}^{n} \delta(l(G_t)=1)\cdot\delta(l(G_{t-o})=1)} \end{aligned} \tag{3}$$

Note that the delta function $\delta(x)$ returns 1 if x is true and 0 otherwise. Clearly, the choice of parameter θ affects both the false positive rate and the detection rate of the individual classifier trained on similarity measure s. Ideally, we would like to maximize the detection rate while minimizing the false positive rate. Optimizing both criteria, however, is typically difficult, and we thus enforce the Neyman-Pearson criterion here, which is to maximize the detection rate while ensuring that the false positive rate is no greater than a certain threshold ρ, where $0 \leq \rho \leq 1$. Hence, we want to find the solution to the following optimization problem:

$$\begin{aligned} \text{argmax}_\theta \quad & d_{s,o}(\theta) \\ \text{subject to:} \quad & w_{s,o}(\theta) \leq \rho \end{aligned} \tag{4}$$

To find the solution, we first establish the following theorem.

Theorem 1: Both $d_{s,o}(\theta)$ and $w_{s,o}(\theta)$ (weakly) monotonically decrease with θ.

Proof: First note that given the training sequence of ADGs $\mathcal{G}_1^n$, the classification result of classifier $C_{s,\theta}^o$ at any time

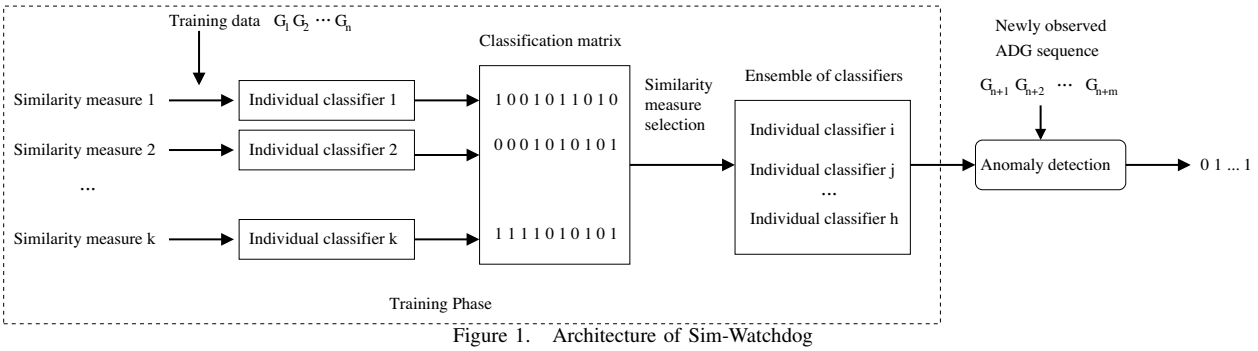


Figure 1. Architecture of Sim-Watchdog

step t varies with only θ. According to Eq. (2), increasing θ (weakly) monotonically decreases $\delta(C^o_{s,\theta}(G_t) = 0)$ at any time step t. Hence, increasing θ (weakly) monotonically decreases both $d_{s,o}(\theta)$ and $w_{s,o}(\theta)$ based on Eq. (3). □

Theorem 1 suggests that the solution to Eq. (4) occurs when $w_{s,o}(\theta) = \rho$ strictly holds, or the closest possible solution if such a condition does not hold in any solution. In practice, however, due to the discrete nature of training samples, there is a *range* of θ that achieves the same optimal detection performance. We use the following steps to find a solution to Eq. (4).

Step 1: We first use binary search to find the optimal or near optimal detection rate when the false positive rate is below the given threshold ρ as follows. We keep a plausible list L of θ values. Initialize θ, θ_{min}, and θ_{max} to be 1.0, 0.0, and $+\infty$, respectively. While $\theta_{max} - \theta_{min}$ is greater than a predefined threshold ϵ, calculate $w_{s,o}(\theta)$ and $d_{s,o}(\theta)$ according to Eq. (3) and proceed in the following cases:

- $w_{s,o}(\theta) > \rho$ (the false positive rate is higher than ρ under θ): according to Theorem 1, we should increase θ to find the boundary. Hence, we assign θ to θ_{min}. Moreover, if $\theta_{max} = +\infty$, we double the value of θ; otherwise, we let θ be the middle point between θ_{min} and θ_{max}.
- $w_{s,o}(\theta) \leq \rho$ (the false positive rate is no higher than ρ under θ): according to Theorem 1, we should decrease θ to find the boundary. Hence, we assign θ to θ_{max}, add tuple $(\theta, w_{s,o}(\theta), d_{s,o}(\theta))$ to list L, and then let θ be the middle point between θ_{min} and θ_{max}.

Step 2: We check all θ values on the plausible list L that have the best detection rate under the false positive rate constraint, and from them, find the one that has the *lowest* false positive rate. That is to say, find the tuple $(\theta', w', d') \in L$ that satisfies the following:

$$d' = d_{s,o}(\theta_{max}) \wedge \forall(\theta, w, d') \in L : w \geq w'. \quad (5)$$

Step 3: Using the performance at θ' as the target, we further search the range of θ values in the neighborhood of θ' that achieve the same performance as θ'. The algorithm returns the middle point in this range as the final solution to Eq. (4).

B. Similarity Measure Selection

Let $\theta^*_{s,o}$ be the parameter of the individual classifier trained on similarity measure s at order o. With a classifier of order o behaving as in Eq. (2), we let the individual classifier trained on similarity measure s work as follows:

$$C_s(G_t) = \prod_{o=1}^{o_{max}} C^o_{s,\theta^*_{s,o}}(G_t). \quad (6)$$

Hence, for any ADG G_t, the individual classifier trained for similarity measure s uses all possible normal samples observed within o_{max} orders to calculate the similarity values. If at *any* of these orders the corresponding classifier classifies the new instance as anomalous, then the overall classifier on this similarity measure classifies it as anomalous. The intuition behind such a stringent rule is to choose similarity measures that behave well at *all* orders.

Given the training ADG sequence $\mathcal{G}^n_1$, the classifier trained on each similarity measure $s \in S$ returns a sequence of classification results, $\widehat{C}_s = (C_s(G_t))_{t=1,2,...,n}$. Combining such classification results from all similarity measures in S leads to a classification matrix $\mathcal{M} = (\widehat{C}_{S_1}, ..., \widehat{C}_{S_{|S|}})^T$, where S_i denotes the ith similarity measure in S. The ith row in matrix $\mathcal{M}$ gives the classification results by the individual classifier trained on the ith similarity measure in S.

Next we discuss how to select similarity measures based on the classification matrix $\mathcal{M}$. This process, again, is driven by the Neyman-Pearson criterion. Consider any subset $S' \subseteq S$ of similarity measures. The ensemble of classifiers based on similarity measures in S', denoted by $C_{S'}$, works as follows:

$$C_{S'}(G_t) = \prod_{s \in S'} C_s(G_t). \quad (7)$$

Hence, *if and only if* individual classifiers trained for all similarity measures in S' agree that ADG G_t is normal, the ensemble of classifiers $C_{S'}$ treats it as normal. Then, $d(S')$ and $w(S')$, the detection rate and the false positive rate of classifier $C_{S'}$, are defined as follows, respectively:

$$d(S') = \frac{\sum_{t=\alpha+1}^{n} \delta(l(G_t) = 0) \times \delta(C_{S'}(G_t) = 0)}{\sum_{t=\alpha+1}^{n} \delta(l(G_t) = 0)}$$

$$w(S') = \frac{\sum_{t=\alpha+1}^{n} \delta(l(G_t) = 1) \times \delta(C_{S'}(G_t) = 0)}{\sum_{t=\alpha+1}^{n} \delta(l(G_t) = 1)}$$

According to the Neyman-Pearson criterion, we want to solve the following optimization problem, which we call the **Similarity Measure Selection (SMS)** problem:

$$\begin{array}{ll} \text{argmax}_{S'} & d(S') \\ \text{subject to:} & w(S') \leq \gamma \end{array} \tag{8}$$

where $0 \leq \gamma \leq 1$.

The inapproximability result of the SMS problem is established as follows:

Theorem 2: For any γ with $0 \leq \gamma \leq 1$ and any $\epsilon > 0$, SMS cannot be approximated in polynomial time within a factor of $1 - 1/e + \epsilon$ unless P = NP.

The proof of Theorem 2 is given in Appendix A. We may also be interested in another way of selecting similarity measures, which is the dual problem of SMS. Its inapproximibility result is provided in Appendix B. Next, we present a greedy algorithm that solves the SMS problem. Define η as follows:

$$\eta = \lfloor \gamma \sum_{t=\alpha+1}^{n} \delta(l(G_t) = 1) \rfloor \tag{9}$$

That is to say, η is equivalent to the maximum number of false positive samples allowed by the algorithm.

We also have the following definitions:

Definition 1: We say that two similarity measures s_1 and s_2 are isomorphic relative to a set of samples $T \subseteq \mathbb{N}_1^n$, i.e., $s_1 \equiv_T s_2$ if and only if for every $t \in T$, we have the following:

$$C_{s_1}(G_t) = C_{s_2}(G_t), \tag{10}$$

and a subset of similarity measures $S' \subseteq S$ is an isomorphic group relative to T if every two similarity measures in S' are isomorphic relative to T.

Definition 2: The utility of a similarity measure s relative to a set of samples $T \subseteq \mathbb{N}_1^n$, denoted as $u_T(s)$, is given by:

$$\begin{aligned} u_T(s) &= \frac{|g_T(s)|}{|c_T(s)|}, \text{ where:} \\ c_T(s) &= \{t \in T : l(G_t) = 1 \text{ and } C_s(G_t) = 0\}, \\ g_T(s) &= \{t \in T : l(G_t) = 0 \text{ and } C_s(G_t) = 0\}, \end{aligned} \tag{11}$$

and the utility of an isomorphic group S' relative to T is defined to be the utility of any similarity measure in it.

Hence, the utility of a similarity measure relative to T is the ratio of the number of anomalous samples in T successfully detected by this similarity measure to that of normal samples in T falsely classified by the similarity measure.

Algorithm 1 presents a solution to the SMS problem. The algorithm keeps track of the set of anomalous samples that have already been detected (D) as well as the set of normal samples that have been wrongly classified as anomalous (F). In each iteration (lines 3-22), the algorithm first finds the isomorphic groups of available similarity measures relative to the samples that are not in D or F. For each of these isomorphic groups, the algorithm checks whether adding it leads to a false positive rate higher than the predefined threshold, γ; if not, the algorithm picks the one with the highest utility and chooses the similarity measures in this group with the highest *relative discrepancies* at order 1, which will be explained shortly. Similar to Eq. (1), we define $\widehat{\mathcal{A}}_{i,j}^o$ as follows:

$$\widehat{\mathcal{A}}_{i,j}^o = \left\{(G_a, G_b), \forall a,b\text{: } i \leq a, b \leq j,\ b = a+o,\ l(G_b) \cdot l(G_a) = 0\right\}.$$

The relative discrepancy of similarity measure s at order o is defined as follows:

$$r_s = \frac{|\mu_s(\mathcal{A}_{i,j}^o) - \mu_s(\widehat{\mathcal{A}}_{i,j}^o)|}{\mu_s(\mathcal{A}_{i,j}^o)}, \tag{12}$$

where we recall that $\mu_s(\mathcal{H})$ gives the mean similarity measure over pairs of graphs in set $\mathcal{H}$. Intuitively, the relative discrepancy of a similarity measure s at order o shows the relative difference in the average similarity measured by s at order o between the set of all pairs of normal graphs and the set of pairs of graphs containing anomalous ones in the training dataset. For the purpose of anomaly detection, a higher relative discrepancy indicates better capability of separating anomalous samples from normal ones. Hence, when the isomorphic group with the highest utility contains a large number of similarity measures, we choose only a few of them with the highest relative discrepancies at order 1. We consider only order 1 because this is the order most often used in practice.

The algorithm terminates when an iteration cannot find any isomorphic group within the false positive rate threshold.

Clearly, Algorithm 1 bears a greedy nature, as in each iteration it always picks the isomorphic group of similarity measures with the highest utility. Consider the following classification matrix (with the first α columns removed) with nine samples and three similarity measures:

Label	1	1	1	0	0	0	0	0	0
Similarity Measure 1	0	1	1	0	0	0	1	1	1
Similarity Measure 2	1	0	1	1	1	0	0	1	1
Similarity Measure 3	1	0	0	0	0	1	1	0	0

Suppose that $\eta = 2$. The greedy algorithm first picks similarity measure 1 (utility = 3), and then similarity measure 2 (utility = 1). Hence, the eventual detection rate is $4/6 \approx 66.7\%$. It is noted that the optimal solution contains only similarity measures 2 and 3, with a detection rate of 100%. This suggests that the greedy algorithm cannot lead to a similar $(1-1/e)$ approximation ratio as achieved by the greedy solution to the maximum set covering problem [13].

C. Online Anomaly Detection

We have discussed how to choose similarity measures based on the Neyman-Pearson criterion from the training ADGs in $\mathcal{G}_1^n$. Let $S^* \subseteq S$ be the set of selected similarity measures. For a selected similarity measure $s \in S^*$, the

Algorithm 1 A greedy solution to the SMS problem

Require: Classification matrix $\mathcal{M}$, parameter γ, a set S of similarity measures, parameter k

Calculate η according to Eq. (9)
$D \leftarrow \emptyset$, $F \leftarrow \emptyset$, $T \leftarrow \mathbb{N}_{\alpha+1}^{n}$, $S' \leftarrow \emptyset$
while true do
  Find a set $\mathcal{Z}$ of isomorphic similarity measures in $S \setminus S'$ relative to T
  $u_{max} \leftarrow 0$, $Z_{max} \leftarrow \emptyset$
  for each isomorphic group $Z \in \mathcal{Z}$ **do**
    Calculate $g_T(Z)$, $c_T(Z)$, and $u_T(Z)$
    if $|c_T(Z) \cup F| \leq \eta$ and $u_T(Z) > u_{max}$ **then**
      $Z_{max} \leftarrow Z$
      $u_{max} \leftarrow u_T(Z)$
    end if
  end for
  if $Z_{max} \neq \emptyset$ **then**
    $Z_k^* \leftarrow$ the set of similarity measures in Z_{max} with the top-k relative discrepancy
    $D \leftarrow D \cup g_T(Z_{max})$
    $F \leftarrow F \cup c_T(Z_{max})$
    $T \leftarrow T \setminus (c_T(Z_{max}) \cup g_T(Z_{max}))$
    $S' \leftarrow S' \cup Z_k^*$
  else
    break
  end if
end while
return S'

corresponding classifier works as follows on a test ADG at time t (*i.e.*, $t > n$):

$$\widehat{C}_s(G_t) = C^{\widehat{o}}_{s,\theta^*_{s,\widehat{o}}}(G_t), \tag{13}$$

where $\widehat{o} = \min\{o : l(G_{t-o}) = 1\}$. That is to say, $\widehat{o}$ is the order of the last ADG that is labeled (or classified) as normal. Hence, the behavior of classifier $\widehat{C}_s(G_t)$ is different from that of $C_s(G_t)$ shown in Eq. (6). This is because during the training phase, we want to choose those similarity measures that perform well at all possible orders, but during the test phase, we expect that the similarity measurements at higher orders are less stable so we use only the classification result based on the most recent sample deemed as normal.

The classifier built on S^* behaves as follows:

$$C_{S^*}(G_t) = \prod_{s \in S^*} \widehat{C}_s(G_t) \text{ for any } t > n. \tag{14}$$

on any newly observed ADG. It is noted that calculation of $C_s(G_t)$ requires label information of previously observed ADGs, and the true label information is only known for ADGs in the training sequence. To circumvent this issue, we define the label information of an ADG G_t when $t > n$ as the classification result of the anomaly detector in Eq. (14).

Consider time step t, where $t > n$, and any selected similarity measure $s \in S^*$. Suppose that the last ADG labeled as 1 (normal) for similarity measure s is $t-o$. Then, for similarity measure s, the ensemble of classifiers uses the detection result of the classifier of order o from similarity measure s (see Eq. (2)). This requires the knowledge of both $\mu_s(\mathcal{A}^o_{1,t-1})$ and $\sigma_s(\mathcal{A}^o_{1,t-1})$, the mean and estimated standard deviation of similarity measure s of order o. Next, we discuss how to update the $\mu_s(\mathcal{A}^o_{1,t})$ and $\sigma_s(\mathcal{A}^o_{1,t})$ in an online fashion once the label of ADG G_t is decided by the ensemble of classifiers as in Eq. (14). Suppose that the maximum order that we keep for each selected similarity measure is o_{max}. Hence, we expect that a sequence of ADGs of length more than o_{max} should be observed very rarely, and if this indeed occurs, we should examine the system more closely to reveal the cause and accordingly, we may need to retrain the ensemble of classifiers.

We let ξ_o keep the count of samples for each order $o = 1, ..., o_{max}$ that gives $|\mathcal{A}^o_{1,t}|$. After the training phase, ξ_o is initialized to be $|\mathcal{A}^o_{1,n}|$. For brevity, define the following: $\mu^o_{s,t} = \mu_s(\mathcal{A}^o_{1,t})$, $\sigma^o_{s,t} = \sigma_s(\mathcal{A}^o_{1,t})$, and $s_{t,o} = s(G_t, G_{t-o})$. Algorithm 2 presents the algorithm to update $\mu^o_{s,t}$ and $\sigma^o_{s,t}$ in an online fashion. Given a newly labeled sample at time step t, the algorithm first checks whether it is labeled as 0 (anomalous). If so, it keeps the old values of $\mu^o_{s,t}$ and $\sigma^o_{s,t}$. Otherwise, it checks every order $o = 1, 2, ..., o_{max}$ to see whether the sample o time steps backward is also normal. Provided that both the current sample and the sample o time steps backward are normal, the algorithm updates the estimated mean and standard deviation based on recurrence relation [6].

Algorithm 2 Update $\mu^o_{s,t}$ and $\sigma^o_{s,t}$ at time step t for $s \in S^*$ and $o = 1, ..., o_{max}$

Require: $\mu^o_{s,t-1}$ and $\sigma^o_{s,t-1}$ for each $s \in S^*$ and each $o = 1, ..., o_{max}$, and ξ_o for each $o = 1, ..., o_{max}$

if $l(G_t) = 0$ **then**
  for $o = 1, ..., o_{max}$ **do**
    $\mu^o_{s,t} \leftarrow \mu^o_{s,t-1}$
    $\sigma^o_{s,t} \leftarrow \sigma^o_{s,t-1}$
  end for
else
  Calculate $r(t)$, the index of last sample labeled as 1
  for $o = 1, ..., o_{max}$ **do**
    if $l(G_{t-o}) = 0$ **then**
      $\mu^o_{s,t} \leftarrow \mu^o_{s,t-1}$
      $\sigma^o_{s,t} \leftarrow \sigma^o_{s,t-1}$
    else
      $\xi_o \leftarrow \xi_o + 1$
      $\mu^o_{s,t} \leftarrow \mu^o_{s,t-1} + \frac{s_{t,t-o} - \mu^o_{s,t-1}}{\xi_o}$
      if $\xi_o > 1$ **then**
        $\sigma^o_{s,t} \leftarrow \sqrt{\frac{(\xi_o - 2)(\sigma^o_{s,t-1})^2 + (s_{t,t-o} - \mu^o_{s,t})(s_{t,t-o} - \mu^o_{s,t-1})}{\xi_o - 1}}$
      else
        $\sigma^o_{s,t} \leftarrow 0$
      end if
    end if
  end for
end if

It is noted that Algorithm 2 incurs only light computational overhead. It requires storage of only those ADG samples labeled as 1 (normal) within the past o_{max} time

steps as well as their indices, the current $\mu^o_{s,t}$ and $\sigma^o_{s,t}$ for each $s \in S^*$ and $o = 1, 2, ..., o_{max}$, and the counter ξ_o for each $o = 1, 2, ..., o_{max}$.

V. Implementation

We implement Sim-Watchdog using both C++ and Python. Currently, we have considered three types of similarity measures: *ranking-based*, *distribution-based*, and *aggregation-based* similarity measures.

A. Ranking-based Similarity Measures

Given two ADGs G_1 and G_2, a ranking-based similarity measure first calculates the centrality measure of each node from a graph-theoretic perspective. Such centrality measures include: *indegree centrality* (the indegree of each node), *outdegree centrality* (the outdegree of each node), *inweight centrality* (the sum of weights on all incoming edges to each node), *outweight centrality* (the sum of weights on all outgoing edges from each node), *betweenness centrality* (the number of pairs of nodes of which the shortest paths pass each node), and *pagerank* (the algorithm used by Google to rank webpages).

For each centrality measure, we rank the nodes in an ADG in a decreasing order, assuming that a node with a higher centrality measure is deemed to be more important. If multiple nodes have exactly the same centrality measure, we *randomly* order them. Let the ranking of nodes in ADG G according to centrality measure x be $\mathcal{R}_{x,G}$, where $\mathcal{R}_{x,G}[i]$ gives the *rank* of the ith node in ADG G. Consider two ADGs G_1 and G_2. Given the same set of vertices V shared by G_1 and G_2, Sim-Watchdog implements the following similarity measures:

Overlapping ratio: The overlapping ratio of the top k nodes between G_1 and G_2 is given by:

$$\frac{\sum_{i=1}^{|V|} \delta(\mathcal{R}_{x,G_1}[i] \leq k \wedge \mathcal{R}_{x,G_2}[i]) \leq k)}{k} \tag{15}$$

Biased overlapping ratio [28]: Let the step size be h and the weight per step reduces at a rate exponential in p where $0 < p < 1$. The top-k biased overlapping ratio of nodes between G_1 and G_2 is given by:

$$\sum_{i=1}^{k} p^k \frac{\sum_{j=1}^{|V|} \delta(\mathcal{R}_{x,G_1}[j] \leq kh \wedge \mathcal{R}_{x,G_2}[j]) \leq kh)}{kh} + 1 - \sum_{i=1}^{k} p^k.$$

B. Distribution-based Similarity Measures

In an ADG, we also look at the distribution of the following quantities: *indegree distribution* (the distribution of the indegrees of the nodes), *outdegree distribution* (the distribution of the outdegrees of the nodes), *inweight distribution* (the distribution of the total weight on the incoming edges), and *outweight distribution* (the distribution of the total weight on the outgoing edges). Given any distribution $p(x)$ with $x \in \mathbb{R}^+$ for ADG G, we quantize it into a vector of discrete values $\mathcal{Q}_x$ of bin size b, where:

$$\mathcal{Q}_G[i] = \int_{ib}^{(i+1)b} p(x)dx, \text{where } i \in \mathbb{N}. \tag{16}$$

The distance measures between two distributions $\mathcal{Q}_{G_1}$ and $\mathcal{Q}_{G_2}$ from ADG G_1 and G_2, respectively, include following:

Euclidean distance: It is simply the Euclidean distance between vectors $\mathcal{Q}_{G_1}$ and $\mathcal{Q}_{G_2}$:

$$\sqrt{\sum_{i=0}^{+\infty} (\mathcal{Q}_{G_1}[i] - \mathcal{Q}_{G_2}[i])^2}. \tag{17}$$

JS (Jensen-Shannon) distance: Define vector $\mathcal{Q}'$ as follows: $\mathcal{Q}'[i] = (\mathcal{Q}_{G_1}[i] + \mathcal{Q}_{G_2}[i])/2$ for any $i \in \mathbb{N}$. The JS distance between vectors $\mathcal{Q}_{G_1}$ and $\mathcal{Q}_{G_2}$ is given by:

$$\frac{1}{2} \sum_{i \in \mathbb{N}: \mathcal{Q}'[i] > 0} \big(\mathcal{Q}_{G_1}[i] log \frac{\mathcal{Q}_{G_1}[i]}{\mathcal{Q}'[i]} + \mathcal{Q}_{G_2}[i] log \frac{\mathcal{Q}_{G_2}[i]}{\mathcal{Q}'[i]}\big). \tag{18}$$

Hellinger distance: The Hellinger distance between vectors $\mathcal{Q}_{G_1}$ and $\mathcal{Q}_{G_2}$ is:

$$\sqrt{\frac{1}{2} \sum_{i \in \mathbb{N}} \big(\sqrt{\mathcal{Q}_{G_1}[i]} - \sqrt{\mathcal{Q}_{G_2}[i]}\big)^2}. \tag{19}$$

C. Aggregation-based Similarity Measures

In an ADG an attribute can be associated with a node or an edge. Hence, for the same attribute type x, we can construct a vector $\mathcal{V}_{x,G}$ from ADG G. Aggregation-based similarity measures calculate the distance between such attribute vectors from two ADGs. If $\mathcal{V}_{x,G_1}$ and $\mathcal{V}_{x,G_2}$ contain numerical values, their distance can be evaluated as follows:

l_p **norm:** The l_p-norm distance between $\mathcal{V}_{x,G_1}$ and $\mathcal{V}_{x,G_2}$ is:

$$\big(\sum_i (\mathcal{V}_{x,G_1}[i] - \mathcal{V}_{x,G_2}[i])^p\big)^{\frac{1}{p}} \tag{20}$$

Gaussian kernel. Using the Gaussian kernel, the distance between $\mathcal{V}_{x,G_1}$ and $\mathcal{V}_{x,G_2}$ is given by:

$$\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\sum_i (\mathcal{V}_{x,G_1}[i] - \mathcal{V}_{x,G_2}[i])^2}{2\sigma^2}}. \tag{21}$$

Numerical attribute types include *edge weight*, *node weight*, *clustering coefficient* of each node, and the *reciprocity* of each node (the number of bidirectional relationships between this node and other nodes), etc.

On the other hand, if $\mathcal{V}_{x,G_1}$ and $\mathcal{V}_{x,G_2}$ contain set values, their distance can be evaluated as follows:

Normalized weighted intersection: We define the normalized weighted intersection of $\mathcal{V}_{x,G_1}$ and $\mathcal{V}_{x,G_2}$ as follows:

$$\frac{|\mathcal{V}_{x,G_1} \cap \mathcal{V}_{x,G_2}|}{\max\{|\mathcal{V}_{x,G_1}|, |\mathcal{V}_{x,G_2}|\}}. \tag{22}$$

Jaccard similarity: The Jaccard similarity of $\mathcal{V}_{x,G_1}$ and $\mathcal{V}_{x,G_2}$ is given by:

$$\frac{|\mathcal{V}_{x,G_1} \cap \mathcal{V}_{x,G_2}|}{|\mathcal{V}_{x,G_1} \cup \mathcal{V}_{x,G_2}|}. \tag{23}$$

Dice similarity: The Dice similarity of $\mathcal{V}_{x,G_1}$ and $\mathcal{V}_{x,G_2}$ is given by:

$$\frac{2|\mathcal{V}_{x,G_1} \cap \mathcal{V}_{x,G_2}|}{|\mathcal{V}_{x,G_1}| + |\mathcal{V}_{x,G_2}|}. \quad (24)$$

Examples of set attribute types include the set of neighbors that a node can reach within k hops (*reachable nodes*).

VI. Evaluation of Sim-Watchdog

As we are not aware of any public dynamic graph dataset with labeled anomalous samples, we introduce how to generate the evaluation dataset from an email dataset collected from a large research institution. We further use this email dataset to evaluate the performance of Sim-Watchdog.

A. Description of Email Dataset

Our email dataset contains email communication records in a large research institution with around ten thousand employees within one year. In total, it comprises 62,946,439 emails, including not only internal emails between the employees but also inbound and outbound emails that the employees communicated with the outside people.

Figure 2(1) show the number of internal emails, outbound external emails, and inbound external emails on each given day in the dataset. There is a strong periodic pattern in each of these curves. Typically, the workdays from Monday to Thursday witness a high volume of emails, and we see that both the numbers of internal and external emails drop on Friday due to the alternative work schedules which allow employees to choose working only on every other Fridays. In early June there was prominent disruption in the aforementioned periodic patterns in email traffic, as the number of internal emails dropped to a significantly low level even on weekdays. This period actually coincided with the ten days of mandatory evacuation due to a wild fire that occurred nearby. There was also a huge spike in the number of internal emails in mid August. This spike resulted from a glitch of the internal system which repeatedly sent email notifications to many employees on that day.

For this study, we generate temporal graphs from the email dataset on a weekly basis. We consider only those internal emails sent among confirmed employees of the institution. As the spike in mid August seen in Figure 2(1) was caused by emails sent automatically from an internal system email account, these emails are not reflected in the temporal graphs generated as described, but such anomalies are easy to identify. In each graph, a vertex represents a confirmed employee's email address, and an edge is created from vertices A to B if A has sent at least one email to B in the corresponding week. For the entire year, we have a sequence of 52 graphs, each generated from email communications in a full week, and Figure 2(2) summarizes the basic properties of these graphs.

From Figure 2(2), there are noticeable dips for the numbers of emails in some weeks. Close examination reveals that these weeks contain either national holidays, or days of main disruptions (e.g., fire evacuation and snow days), as shown as follows: week 2 (*Martin Luther King, Jr. Day*), week 7 (*President's Day*), week 21 (*Memorial Day*), week 25 (*Fire evacuation*), week 26 (*Independence Day*), week 35 (*Labor Day*), week 40 (*Columbus Day*), week 44 (*Veterans Day*), week 46 (*Thanksgiving Day*), week 48 (*snow days*), and week 51 (*Christmas*). We thus treat the graphs generated from these weeks as anomalous, and the others normal. Hence, the fraction of anomalous samples is approximately 21% (≈11 / 52).

B. Experimental Setup

In all our experiments, we set parameter α (see Section IV-A) to be 6, meaning that the first six samples are not used for similarity measure selection. We use the first 24 samples for training and the remaining ones for testing purpose. We also let parameter o_{max} be 2, and γ (see Eq. (8)) be 0.05. We choose parameter ρ (see Eq. (4)) in the range of $[0.01, 0.02, ..., 0.15]$.

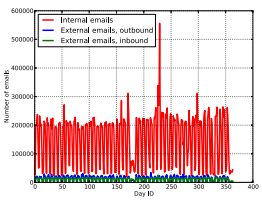
Ranking-Based Similarity Measures: The rankings are generated from six centrality measures, including indegree centrality, outdegree centrality, inweight centrality, outweight centrality, betweenness, and pagerank. For each centrality measure, we consider both the overlapping ratio and biased overlapping ratio methods to compare rankings. When the overlapping ratio approach is used, we consider the top k nodes, where k is chosen between 100 and 500. When the biased overlapping ratio approach is used, the step size is set to be 10, parameter p be 0.5, and k between 10 and 50. In total, we have 24 rank-based similarity measures.

Distribution-Based Similarity Measures: We consider the distribution of indegree, outdegree, inweight, and outweight in each ADG. When calculating the distribution, we choose the bin size from 1, 10, and 100. The distance between two distributions is evaluated in three ways: Euclidean distance (EU), JS distance (JS), and Hellinger distance (HL). We have 36 distribution-based similarity measures in total.

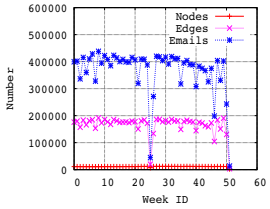
Aggregation-Based Similarity Measures: We consider four aggregate attributes: edge weight (number of messages on each edge), node weight (number of messages sent from each user), clustering coefficient, and reciprocity. For each one of them, we use the Gaussian kernel (where parameter σ varies among 200, 400, 800, and 1600) and l_p (where p is 1 or 2) to evaluate the distance between two attribute vectors. We also consider the set of reachable nodes as the attribute, and use normalized weighted intersection, Jaccard similarity, and Dice similarity to evaluate the distance between two sets. In total, we have 48 aggregation-based similarity measures.

C. Experimental Results

Evaluation of Sim-Watchdog: Figure 3 shows the performance of Sim-Watchdog when it is fed all three types of similarity measures, which amount to 108 individual ones

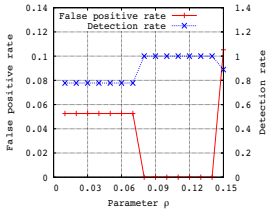


(1) Temporal volumes

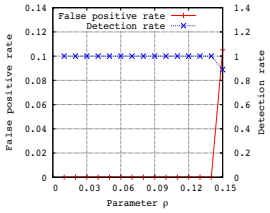


(2) Basic graph properties

Figure 2. Email dataset description

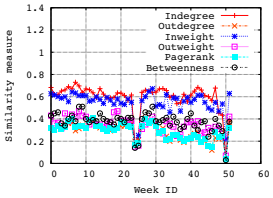


(1) k = 1

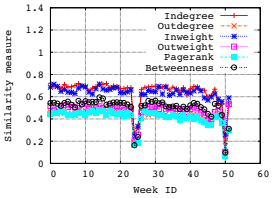


(2) k = 3

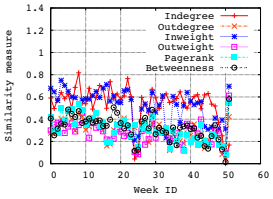
Figure 3. Performance of Sim-Watchdog



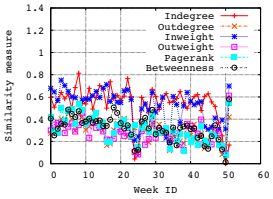
(1) Top 100 overlapping



(2) Top 500 overlapping

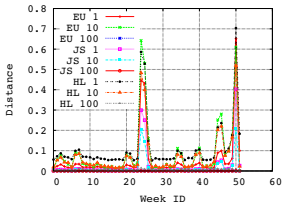


(3) Biased overlapping ($k = 10$)

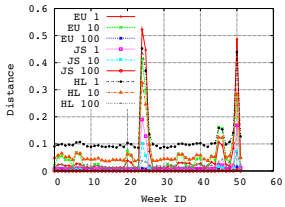


(4) Biased overlapping ($k = 50$)

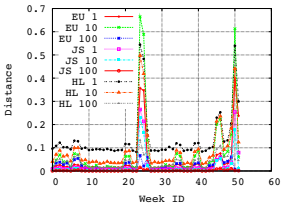
Figure 4. Rank-based similarity measures (for biased overlapping, the step size h is 10)



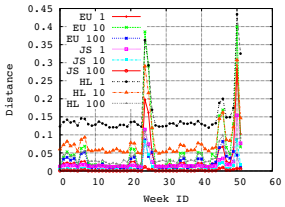
(1) Indegree



(2) Outdegree

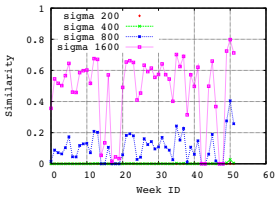


(3) Inweight

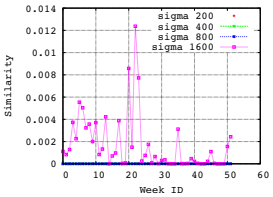


(4) Outweight

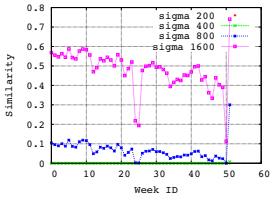
Figure 5. Distribution-based similarity measures



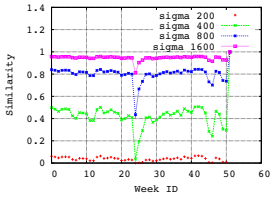
(1) Edge weight, gaussian



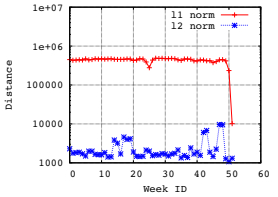
(2) Node weight, gaussian



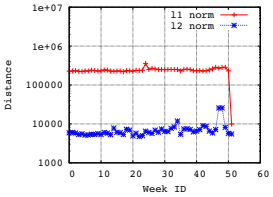
(3) Clustering coefficient, gaussian



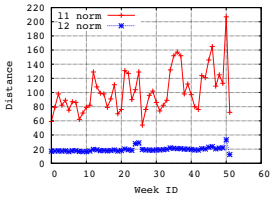
(4) Reciprocity, gaussian



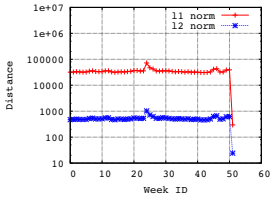
(5) Edge weight, l_p-norm



(6) Node weight, l_p-norm



(7) Clustering coefficient, l_p-norm



(8) Reciprocity, l_p-norm

Figure 6. Aggregation-based similarity measures

ρ range	k = 1	k = 3
$[0.01, 0.07]$	Inweight, JS, bin=100	Inweight, JS, bin=100 Inweight, JS, bin=10 Inweight, EU, bin=100
$[0.08, 0.14]$	Inweight, JS, bin=100	Inweight, JS, bin=100 Inweight, JS, bin=10 Outdegree, JS, bin=10
$[0.15, 0.15]$	Indegree, HL, bin=10	Indegree, HL, bin=10

Table I
SELECTED SIMILARITY MEASURES

in total. Recall that Sim-Watchdog selects the similarity measures with the top k relative discrepancy within the same isomorphic groups. We notice that the performance of Sim-Watchdog varies under three ranges of ρ values. The selected similarity measures are presented in Table I. All the chosen similarity measures are of type *distribution-based*.

It is noted from Figure 3 that parameter ρ affects the performance of Sim-Watchdog. Recall that ρ controls the false positive rate of the classifiers trained for *individual* similarity measures, and the threshold on the false positive rate of the *ensemble* of classifiers, γ, is fixed at 0.05 in our experiments. If ρ is very small, individual classifiers are trained to have low false positive rates. Hence, to satisfy the false positive rate constraint γ, the ensemble of classifiers thus have more choices from these individual ones. On the other hand, if ρ is very large, individual classifiers are trained to allow high false positive rates, which provides the ensemble of classifiers fewer choices that meet the false positive rate constraint γ. This explains that for $k = 3$, fewer similarity measures are selected when $\rho = 0.15$ than the other ranges of ρ values. It is also observed that for the middle range (i.e., $0.08 \leq \rho \leq 0.14$), Sim-Watchdog is able to detect all anomalies without leading to any false alarms, suggesting that when parameters are properly set, Sim-Watchdog can perform with high accuracy.

Similarity measures between consecutive graphs: After observing the performance of Sim-Watchdog depicted in Figure 3, we now show how the similarity between consecutive graphs evaluated by different types of similarity measures evolves over time to gain deeper insights into the similarity measures selected by Sim-Watchdog. The rank-based similarity measures among consecutive graphs extracted from the email dataset are depicted in Figure 4. One interesting observation is that the indegree and the inweight centrality measures are higher than the other types of similarity measures, regardless of the scheme used to compare the rankings. Moreover, for the top k overlapping ratio scheme, when we increase k, the similarity curve becomes smoother. In contrast, for the top k biased overlapping ratio sheme, increasing k from 10 to 50 leads to little change, because the overlapping ratio among a large number of top-ranked nodes is weighted with only a small value (exponential in the number of steps included). The distribution-based similarity measures between consecutive graphs extracted from the email dataset are shown in Figure 5. We observe that the spikes in the curves coincide with the dips seen in Figure 2(2). This suggests that distribution-based similarity measures could perform well in anomaly detection for the email dataset. Figure 6 shows the aggregation-based similarity measures of two consecutive graphs extracted from the email dataset. Clearly, these measures do not contain strong signals that we can rely on to predict the anomalies shown in Figure 2(2). All these observations confirm the choices on similarity measures made by Sim-Watchdog.

Phase Transition: One may wonder why there is a noticeable phase transition in Table I for different ρ ranges. Recall that there are 24 samples in the training part, among which the first six are ignored and the remaining ones have two labeled as 0 (anomalous). From Eq. (3), we know that calculation of the false positive rate at a given order o depends on the cases where *both* the current sample and the past one of order o are labeled as 1 (normal). We have 14 such cases, among which, if only one were to be falsely classified as 0, the false positive rate would be around 7.14%, and if two of them falsely classified as 0, the false positive rate would be around 14.28%. This explains why the phase transition occurs when ρ increases from 0.07 to 0.08, or from 0.14 to 0.15.

VII. CONCLUSIONS

The ubiquity of graphs in modeling relationships or information flows among entities in networks and distributed systems motivates us to explore methods for anomaly detection in dynamic graphs. In this work, we propose a systematic framework called Sim-Watchdog, which leverages temporal similarity inherent in graph-modeled network data for anomaly detection. We apply Sim-Watchdog to dynamic graphs extracted from email communication records in a large research institution and demonstrate its effectiveness for supervised anomaly detection.

This work focuses on anomaly detection on dynamic graphs abstracted from data collected from networks or distributed systems. Albeit graphs are commonly used to characterize the structures or information flows in these systems, they are not a panacea for detecting all possible anomalous activities. For instance, it may be difficult to rely on only graph-based anomaly detection schemes to detect protocol misuse at individual nodes. Hence, the proposed method in this work, while offering a new perspective into anomaly detection in networks or distributed systems, is not intended to replace, but rather to complement existing approaches in this domain.

The similarity measures currently implemented in Sim-Watchdog are by no means exhaustive. Actually, it is an active research field to explore graph kernels [17]. Despite the variety of ways of comparing graphs, it is unclear which one, or subset, of them performs best, and the answer surely depends on the specific system that produces the dynamic

graphs. Hence, these recent advances are orthogonal to the rationale behind Sim-Watchdog, which is to offer a generic framework for selecting similarity measures most qualified for anomaly detection. These new graph similarity measures can be easily incorporated into the Sim-Watchdog framework, which remains as our future work.

In this study, we use dynamic graphs extracted from an email dataset for performance evaluation. The design of Sim-Watchdog renders it applicable to any dynamic graph dataset containing labeled training samples. If Sim-Watchdog cannot find any candidate similarity measures that meet the false positive rate requirements in the Neyman-Pearson criterion, we should further examine the labeled samples to look for other similarity measures with better discriminative power.

REFERENCES

[1] L. Akoglu, M. McGlohon, and C. Faloutsos. Oddball: spotting anomalies in weighted graphs. In *PAKDD'10*.

[2] P. Barford and D. Plonka. Characteristics of network traffic flow anomalies. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*. ACM, 2001.

[3] C. C. Bilgin and B. Yener. Dynamic network evolution: Models, clustering, anomaly detection. *IEEE Networks*, 2006.

[4] H. Cavusoglu, B. Mishra, and S. Raghunathan. The value of intrusion detection systems in information technology security architecture. *Information Systems Research*, 16(1), 2005.

[5] D. Chakrabarti. Autopart: parameter-free graph partitioning and outlier detection. In *PKDD'04*.

[6] T. F. Chan, G. H. Golub, and R. J. LeVeque. Algorithms for computing the sample variance: Analysis and recommendations. *The American Statistician*, 37(3), 1983.

[7] M. E. Crovella and A. Bestavros. Self-similarity in world wide web traffic: evidence and possible causes. *IEEE/ACM Trans. Netw.*, 5(6):835–846, December 1997.

[8] H. Ebel, L.-I. Mielsch, and S. Bornholdt. Scale-free topology of e-mail networks. *arXiv preprint cond-mat/0201476*, 2002.

[9] W. Eberle and L. Holder. Discovering structural anomalies in graph-based data. In *Proceedings of the Seventh IEEE International Conference on Data Mining Workshops*, 2007.

[10] J. M. Estevez-Tapiador, P. Garcia-Teodoro, and J. E. Diaz-Verdejo. Anomaly detection methods in wired networks: a survey and taxonomy. *Computer Communications*, 2004.

[11] D. Fay, H. Haddadi, S. Uhlig, L. Kilmartin, A. W. Moore, J. Kunegis, and M. Iliofotou. Discriminating graphs through spectral projections. *Computer Networks*, 55(15), 2011.

[12] Y. Gu, A. McCallum, and D. Towsley. Detecting anomalies in network traffic using maximum entropy estimation. In *Proceedings of ACM IMC'05*, pages 32–32, 2005.

[13] D. S. Hochbaum. Approximation algorithms for NP-hard problems. chapter Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems. PWS Publishing Co., 1997.

[14] N. S. Ketkar, L. B. Holder, and D. J. Cook. Empirical comparison of graph classification algorithms. In *IEEE Symposium on Computational Intelligence and Data Mining*, 2009.

[15] A. Lakhina, M. Crovella, and C. Diot. Diagnosing network-wide traffic anomalies. In *ACM SIGCOMM Computer Communication Review*, 2004.

[16] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Trans. Netw.*, 2(1):1–15, February 1994.

[17] G. Li, M. Semerci, B. Yener, and M. J. Zaki. Graph classification via topological and label attributes. In *9th Workshop on Mining and Learning with Graphs*, 2011.

[18] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *IMC'07*.

[19] H.D.K. Moonesinghe and P.-N. Tan. Outrank: a graph-based outlier detection framework using random walk. *International Journal on Artificial Intelligence Tools*, 17(01):19–36, 2008.

[20] J. Neil, C. Storlie, C. Hash, A. Brugh, and M. Fisk. Scan statistics for the online detection of locally anomalous subgraphs. *Technometrics*, 2013.

[21] C. C. Noble and D. J. Cook. Graph-based anomaly detection. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003.

[22] P. Papadimitriou, A. Dasdan, and H. Garcia-Molina. Web graph similarity for anomaly detection. *Journal of Internet Services and Applications*, 1(1):19–30, 2010.

[23] Y. Park, C. E. Priebe, and A. Youssef. Anomaly detection in time series of graphs using fusion of graph invariants. *http://arxiv.org/abs/1210.8429*.

[24] A. Patcha and J.-M. Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 51(12):3448–3470, 2007.

[25] C. Scott and R. Nowak. A neyman-pearson approach to statistical learning. *IEEE Transactions on Information Theory*, 51(11), 2005.

[26] M. Thottan and C. Ji. Anomaly detection in IP networks. *Signal Processing, IEEE Transactions on*, 51(8), 2003.

[27] V. V. Vazirani. *Approximation algorithms*. Springer, 2004.

[28] W. Webber, A. Moffat, and J. Zobel. A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst.*, 28(4), 2010.

[29] G. Yan. Peri-watchdog: Hunting for hidden botnets in the periphery of online social networks. *Computer Networks*, 57(2):540–555, 2013.

[30] G. Yan, S. Eidenbenz, and E. Galli. SMS-watchdog: Profiling social behaviors of sms users for anomaly detection. In *Recent Advances in Intrusion Detection*. Springer, 2009.

[31] G. Yan, S. Eidenbenz, and B. Sun. Mobi-watchdog: You can steal, but you can't run! In *Proceedings of the second ACM conference on Wireless network security*, pages 139–150. ACM, 2009.

[32] Q. Yan and R. S. Blum. Distributed signal detection under the neyman-pearson criterion. *Information Theory, IEEE Transactions on*, 47(4):1368–1377, 2001.

[33] Y. Zhang, N. Meratnia, and P. Havinga. Outlier detection techniques for wireless sensor networks: A survey. *Communications Surveys & Tutorials, IEEE*, 12(2):159–170, 2010.

APPENDIX A: PROOF OF THEOREM 2

Proof: First note that the classification results of the first α ADGs are ignored in computation of $d(S')$ and $w(S')$. Hence, the original SMS problem is equivalent to the one in which we remove the first α columns of classification matrix $\mathcal{M}$ and let α be 0. Hence, we assume that α is 0 in our proof.

Consider an instance I_{MSC} of the Maximum Set Covering Problem (MSC): given a universe $\mathcal{U} = \{\mathcal{U}_1, \mathcal{U}_2, ..., \mathcal{U}_{|\mathcal{U}|}\}$, a family $\mathcal{F}$ of subsets of $\mathcal{U}$, any integer k, and any integer x, select k subsets such that their union contains as many elements as possible in $\mathcal{U}$.

Given I_{MSC}, we construct the corresponding instance I_{SMS} of the SMS problem as follows. The training ADG sequence contains $|\mathcal{U}| + z$ ADGs in it, where z will be explained shortly. For the first $|\mathcal{U}|$ ADGs, their labels are always 0 (anomalous), and for the remaining ones their labels are 1 (normal). There are $|\mathcal{F}|$ similarity measures in S. For the individual classifier trained on the ith similarity measure, it classifies the jth ADG, where $1 \leq j \leq |\mathcal{U}|$, as 0 (anomalous) if element $\mathcal{U}_j$ belongs to the ith subset in family $\mathcal{F}$, and 1 (normal) otherwise. For any γ with $0 \leq \gamma \leq 1$, we have:

Case 1: $k \geq \gamma|\mathcal{F}|$. We have $z = \lceil k/\gamma \rceil$. Note that $\lfloor x \rfloor$ and $\lceil x \rceil$ return the floor and ceil functions of real number x, respectively. The individual classifier trained on the ith similarity measure, where $1 \leq i \leq |\mathcal{F}|$, classifies the ith ADG among the last z ADGs as 0 (anomalous) and all the others as 1 (normal).

Case 2: $k < \gamma|\mathcal{F}|$. We have $z = \lfloor \frac{|\mathcal{F}|-k}{1-\gamma} \rfloor$. The individual classifier trained on the ith similarity measure, where $1 \leq i \leq |\mathcal{F}|$, classifies the ith ADG among the last z ADGs as 0 (anomalous), the last $z - |\mathcal{F}|$ ADGs also as 0, and all the others as 1 (normal).

For the SMS problem, we want to find a subset of similarity measures such that their detection rate is as high as possible under the constraint that the false positive rate is at most γ.

Let OPT_{MSC} and OPT_{SMS} be the optimal solutions of I_{MSC} and I_{SMS}, respectively. It is known that MSC cannot be approximated in polynomial time within a factor of factor of $1 - 1/e + \epsilon$ for any $\epsilon > 0$ unless P = NP [13]. To prove that SMS cannot be approximated in polynomial time within a factor of $1 - 1/e + \epsilon$ for any ϵ unless P = NP, we show the following:

[1] $\mathbf{OPT_{MSC} \geq x \Rightarrow OPT_{SMS} \geq x/|\mathcal{U}|}$: Consider the set S' of similarity measures that correspond to the chosen subsets in OPT_{MSC}, and we have $|S'| = k$. We now show that using the similarity measures in S' leads to a false positive rate of at most γ. For Case 1 ($k \geq \gamma|\mathcal{F}|$), the false positive rate is $|S'|/z$; therefore, we have: $|S'|/z \leq k/z \leq \gamma$. For Case 2 ($k < \gamma|\mathcal{F}|$), the false positive rate is $(|S'| + z - |\mathcal{F}|)/z$; therefore, we have:

$$(|S'|+z-|\mathcal{F}|)/z \leq 1+(k-|\mathcal{F}|)/z \leq 1+(k-|\mathcal{F}|)\frac{1-\gamma}{|\mathcal{F}|-k} = \gamma.$$

Next, we show that using the similarity measures in S' leads to a detection rate no less than $x/|\mathcal{U}|$. As $OPT_{MSC} \geq x$, the solution covers at least x elements in $\mathcal{U}$. Given how we construct I_{SMS}, using the similarity measures in S' can correctly detect x out of $|\mathcal{U}|$ ADGs labeled as 0. Hence, we must have: $OPT_{SMS} \geq x/|\mathcal{U}|$.

[2] $\mathbf{OPT_{MSC} < (1 - 1/e + \epsilon)x \Rightarrow OPT_{SMS} < (1 - 1/e + \epsilon)x/|\mathcal{U}|}$: We prove $OPT_{SMS} \geq (1 - 1/e + \epsilon)x/|\mathcal{U}| \Rightarrow OPT_{MSC} \geq (1-1/e+\epsilon)x$ instead. Consider the set F' of subsets in family $\mathcal{F}$ that correspond to the similarity measures selected in OPT_{SMS}. As $OPT_{SMS} \geq (1 - 1/e + \epsilon)x/|\mathcal{U}|$, at least $(1 - 1/e + \epsilon)x$ ADGs labeled as 0 have been detected successfully as anomalous by OPT_{SMS}. Hence, the union of subsets in F' have at least $(1 - 1/e + \epsilon)x$ elements.

Let y be the number of similarity measures chosen in OPT_{SMS}. For Case 1 ($k \geq \gamma|\mathcal{F}|$), the false positive rate is y/z, which is no greater than γ; therefore, we have: $y \leq \gamma z$. As y is an integer, $y \leq \lfloor \gamma z \rfloor = \lfloor \gamma \lceil k/\gamma \rceil \rfloor$. If $\gamma = 1$, $y \leq k$ immediately follows; otherwise ($\gamma < 0$), $y \leq \lfloor \gamma(k/\gamma + 1) \rfloor = k$. For Case 2 ($k < \gamma|\mathcal{F}|$), the false positive rate is $(y + z - |\mathcal{F}|)/z$, which is no greater than γ; therefore, we have: $y \leq |\mathcal{F}| - (1 - \gamma)z$. As y is an integer, $y \leq \lfloor |\mathcal{F}| - (1 - \gamma)z \rfloor = \lfloor |\mathcal{F}| - (1 - \gamma)\lfloor \frac{|\mathcal{F}|-k}{1-\gamma} \rfloor \rfloor$. Hence, if $\gamma = 0$, $y \leq k$ immediately follows; otherwise ($\gamma > 0$), $y \leq \lfloor |\mathcal{F}| - (1 - \gamma)(\frac{|\mathcal{F}|-k}{1-\gamma} - 1) \rfloor = \lfloor k + (1 - \gamma)) \rfloor = k$. As for both cases no more than k similarity measures are chosen and selecting the subsets corresponding to these similarity measures in OPT_{SMS} leads to at least $(1-1/e+\epsilon)x$ elements covered in F', OPT_{MSC} must contain at least $(1 - 1/e + \epsilon)x$ elements. □

APPENDIX B: THE DUAL SMS (SMS^{-1}) PROBLEM

Using the same notations as Eq. (8), the **Dual Similarity Measure Selection (SMS^{-1})** problem is defined as follows:

$$\begin{aligned} \text{argmin}_{S'} \quad & w(S') \\ \text{subject to:} \quad & d(S') \geq \gamma' \end{aligned} \tag{25}$$

where $0 \leq \gamma' \leq 1$.

In the following theorem, we establish the inapproximability result of the SMS^{-1} problem:

Theorem 3: For any γ' with $0 \leq \gamma' \leq 1$ and a constant b, SMS^{-1} cannot be approximated in polynomial time within a factor of $b \log n$, where n is the number of training samples, unless $NP \subseteq ZTIME(n^{loglogn})$[1].

Proof: First note that the cardinality set cover (CSC) problem cannot be approximated in polynomial time within a factor of $b \log \tilde{n}$, where $\tilde{n}$ is the size of the universal set of the set cover instance, unless $NP \subseteq ZTIME(\tilde{n}^{\log \log \tilde{n}})$ [27]. Next, we prove that there is a gap preserving reduction from CSC to SMS^{-1}.

Consider an instance I_{CSC} of the CSC problem: given a universe $\mathcal{U} = \{\mathcal{U}_1, \mathcal{U}_2, ..., \mathcal{U}_{|\mathcal{U}|}\}$, a family $\mathcal{F}$ of subsets of $\mathcal{U}$, and an integer k, find whether there is a set covering of size at most k. A cover or set covering is defined to be a subfamily $\mathcal{C} \subseteq \mathcal{F}$ of sets whose union is $\mathcal{U}$. Given I_{CSC}, we construct the corresponding instance $I_{SMS^{-1}}$ of the SMS^{-1} problem as follows. There are $|\mathcal{F}|$ similarity measures in S. The training ADG sequence contains $|\mathcal{F}| + \lfloor |\mathcal{U}|/\gamma' \rfloor$ ADGs in it. (1) For the first $|\mathcal{F}|$ ADGs, their true labels are all 1 (normal), and the ith similarity measure where $1 \leq i \leq |\mathcal{F}|$ classifies only the ith ADG among these first $|\mathcal{F}|$ ADGs as 0 (anomalous) and all the other ADGs as 1 (normal). (2) For the next $\mathcal{U}$ ADGs, their true labels are all 0 (anomalous), and the individual classifier trained for the ith similarity measure, where $1 \leq i \leq |\mathcal{F}|$, classifies the jth ADG among these $\mathcal{U}$ ADGs as 0 (anomalous) if the ith subset contains element $\mathcal{U}_j$, and 1 (normal) otherwise. (3) For the last $\lfloor |\mathcal{U}|/\gamma' \rfloor - |\mathcal{U}|$ ADGs, their true labels are all 0 but all similarity measures classify them as 1 (normal).

[1]$\mathbf{OPT_{CSC} \leq x \Rightarrow OPT_{SMS^{-1}} \leq x/|\mathcal{F}|}$:

Given an optimal solution to CSC, we choose the similarity measures corresponding to the chosen subsets in $\mathcal{F}$. As $OPT_{CSC} \leq x$, at most x similarity measures are chosen. Note that in $I_{SMS^{-1}}$, only the first $|\mathcal{F}|$ ADGs have true labels as 1 (normal). Hence, the false positive rate is at most $x/|\mathcal{F}|$. On the other hand, for the middle $|\mathcal{U}|$ ADGs (whose true labels are 0), using the selected similarity measures must be able to detect all them due to set covering. As there are in total $\lfloor \mathcal{U}/\gamma' \rfloor$ ADGs with true labels as 0, the fraction of these ADGs that are detected as 0 by the selected similarity measures is $\mathcal{U}/\lfloor \mathcal{U}/\gamma' \rfloor \geq \gamma'$. Hence, using the selected similarity measures leads to $d(S') \geq \gamma'$. Hence, we must have: $OPT_{SMS^{-1}} \leq x/|\mathcal{F}|$.

[2]$\mathbf{OPT_{CSC} > bx \log\log|\mathcal{U}| \Rightarrow OPT_{SMS^{-1}} > bx \log\log|\mathcal{U}|/|\mathcal{F}|}$:

We prove that $OPT_{SMS^{-1}} \leq bx \log\log|\mathcal{U}|/|\mathcal{F}| \Rightarrow OPT_{CSC} \leq bx \log\log|\mathcal{U}|$ instead. Consider S', the set of selected similarity measures in the optimal solution to SMS^{-1}. As $d(S') \geq \gamma'$ and the number of ADGs with true labels as 0 is $\lfloor |\mathcal{U}|/\gamma' \rfloor$, using similarity measures in S' can detect at least $\lceil \gamma' \times \lfloor |\mathcal{U}|/\gamma' \rfloor \rceil = |\mathcal{U}|$ ADGs labeled as 0 (anomalous). Given how we construct $I_{SMS^{-1}}$, the subsets corresponding to the selected similarity measures must be a set cover. As $OPT_{SMS^{-1}} \leq bx \log\log|\mathcal{U}|/|\mathcal{F}|$ and there are $|\mathcal{F}|$ ADGs labeled as 1, the number of selected similarity measures is no greater than $bx \log\log|\mathcal{U}|$. Hence, we must have: $OPT_{CSC} \leq bx \log\log|\mathcal{U}|$. □

[1]Note that $ZTIME(T(n))$ contains every language for which there exists an expected time $O(T(n))$ zero-error probabilistic Turing Machine.