# On Signatures for Communication Graphs

Graham Cormode #[1], Flip Korn #[2], S. Muthukrishnan *[3], Yihua Wu *[4]

#*AT&T Labs—Research*
*Florham Park, NJ 07932, U.S.A.*
[1]graham@research.att.com
[2]flip@research.att.com

**Department of Computer Science, Rutgers University*
*110 Frelinghuysen Road, Piscataway, NJ 08854, U.S.A.*
[3]muthu@cs.rutgers.edu
[4]yihwu@cs.rutgers.edu

*Abstract*— **Communications between individuals can be represented by (weighted, multi-) graphs. Many applications operate on communication graphs associated with telephone calls, emails, Instant Messages (IM), blogs, web forums, e-business relationships and so on. These applications include identifying repetitive fraudsters, message board aliases, multiusage of IP addresses, etc. Tracking electronic identities in communication networks can be achieved if we have a reliable "signature" for nodes and activities. While many examples of ad hoc signatures can be proposed for particular tasks, what is needed is a systematic study of the principles behind the usage of signatures for any task.**

**We develop a formal framework for the use of signatures in communication graphs and identify three fundamental properties that are natural to signature schemes: persistence, uniqueness and robustness. We argue for the importance of these properties by showing how they impact a set of applications. We then explore several signature schemes — previously defined and new — in our framework and evaluate them on real data in terms of these properties. This provides insights into suitable signature schemes for desired applications. Finally, as case studies, we focus on two concrete applications in enterprise network traffic. We apply signature schemes to these problems and demonstrate their effectiveness.**

## I. Introduction

In the everyday world, instances of interaction or communication between individuals are everywhere. For example, individuals speak to each other via telephone; IP traffic is passed between hosts; authors write documents together; and so on. There are other examples in which individuals interact with other entities, such as when users pose search queries; post and comment on messages on bulletin boards or blog sites; or when stock traders transact stocks, bonds and other goods. In an indirect sense, users "communicate" with each other via the common objects. In all cases the communication between individuals can be repeated at different times (such as in the case of telephone calls) and weighted (say, the quantity of stock bought, or the duration of a call).

Given this abundance of communication between individuals, many applications rely on analyzing the patterns behind the communications, for example:

- *Anti-Aliasing:* is an individual behind multiple presences in the communication network? This happens e.g. when an individual has multiple connection points (home, office, hotspot) to the Internet.
- *Security:* has some individual's 'identity' been taken over by someone else? This happens when a person is given access to another's laptop, or when a cellphone is stolen and used by someone else. Is a new user who arrives at a particular time really the reappearance of an individual who has been observed earlier?
- *Analysis of Data Anonymization:* can we identify nodes from an anonymized graph given outside information about known communication patterns per individual? This happens in author identification of double-blind submissions.

Each of the questions above and others of this nature that rely on communication patterns can naturally be solved by designing suitable *signatures* for the individuals. Informally, signatures capture the distinctive or discriminatory communication behavior of an individual (telephone user, IP address, trader or user of a search service, etc). While the concept of signatures is self-evident, formalizing and applying signatures to a specific task is really an art. Typically, in any particular task, "signatures" are defined based on intuition and experimentally validated against a labeled set. This approach has been instantiated successfully for certain specific communication settings and applications. It was done for telephone networks in [5], [10] where the authors defined a *community of interest* to be the top-k numbers called by a given telephone number. With appropriate age weighting and a suitable $k$, this was argued to be highly discriminatory for detecting repetitive debtors. A second example is when a signature formed from bibliographic citations is used to identify authors of double-blind submissions [11]. There are many other examples in areas including Security [18], [6], [4], [5], Collaborative Filtering [25], Computer Networking [22], [27], [16], [21], and Social Network Analysis [8], [1].

In this paper, we adopt the signature-based approach to analyzing the patterns of communication exhibited by individuals. However, we focus on the process of how signatures are developed and applied. In particular, we propose a framework in which we first agree on a set of properties of signatures that are natural and, when faced with an application, determine which of these properties of signatures are needed, and then, seek out examples of signatures already known or design new ones which will have those properties. Hence, the process will focus on abstract properties that are needed, and "shopping" for signatures with those properties.

We develop a framework for the formal use of signatures for tasks that involve analyzing communication patterns, for very general notions of communication. We model the communications between entities using a suitably weighted graph, and define a signature of a node abstractly in terms of the graph. Our contributions are:

- We identify basic properties of signatures such as persistence, uniqueness and robustness, and for several tasks that involve analyzing communication patterns, study which of these properties are needed. For example, a task such as finding multiple presences of the same individuals in a time window does not need signatures to be persistent; likewise, analyzing the changing behavior of a single individual over time may not need signatures to be discriminating.
- We consider specific signatures — some previously known, others new — for communication graphs and study what basic properties they have, using extensive experiments with real data. This helps identify which signatures are suitable for each of the tasks.
- We complement our conceptual results with a detailed experimental study on two concrete applications in enterprise network traffic. We adopt the framework for respective tasks. Our results show that signatures based on the combination of a few application-desired properties is quite effective.

In what follows, we first introduce our framework for analyzing signatures in Section II, the properties we desire, and an analysis of their values for a variety of applications. In Section III, we describe various signature schemes based on expected features of communication graphs, and their characteristics. We evaluate signatures empirically, first studying the general characteristics in Section IV, and then for particular applications in Section V. We lastly discuss scalability issues in Section VI, then survey related work and give concluding remarks.

## II. Framework

Here we describe our framework for designing and evaluating topological signatures for communication graphs. We define the domain of our signatures, and three general properties for evaluating them. Finally, we discuss how these properties relate to specific applications of signatures.

### A. Individuals and Labels

A communication graph is defined by the observed patterns of communications between nodes representing individual users. However, we observe only the *labels* of these nodes rather than the actual identities of the *individuals* who are communicating. For example, we may see traffic on a network between pairs of IP addresses, or calls between pairs of telephone numbers. These may be unique to individuals, but not necessarily: an IP address may be dynamically reassigned to another user, a cell phone may be loaned to a friend, etc.[1] What we can do is to analyze the observed communication between nodes in the graph, and infer the behavior of individuals. We need the assumption that the hidden mapping of individuals to node labels in the graph is for the most part consistent over time: if the mapping of every label is randomly reassigned at every time step, then the task of building good signatures becomes appreciably harder, especially if only basic information about the communications is available. Indeed, many of the applications we discuss here concern finding examples where the mapping from users to labels is slightly perturbed. In subsequent sections, we concentrate on building signatures based on the observable labels, while understanding that our purpose is to use the signatures to identify the behavior of individuals.[2]

### B. Signature Space

Let $G_t = \langle V, E_t \rangle$ be a communication graph that has been aggregated over some time interval at $t$.[3] The graph may be revealed as a sequence of directed edges $(v, u)$, and then aggregated, or may arrive as a set of aggregated edges. An edge $(v, u) \in E_t$ represents communication exchanges from node $v$ to $u$ in $G_t$, and the weight of edge $(v, u)$, denoted $C[v, u]$, reflects the "volume" (e.g., frequency) of this communication. For each node $v \in V$, we denote by $I(v)$ and $O(v)$ the set of $v$'s in-neighbors and out-neighbors during the time interval, respectively. That is, $I(v) = \{u | (u, v) \in E_t\}$ and $O(v) = \{u | (v, u) \in E_t\}$. In many common cases the nodes are partitioned into two distinct classes, such as clients and servers, and so the induced graph is bipartite. A bipartite communication graph $G_t = \langle V_1 + V_2, E_t \subseteq V_1 \times V_2 \rangle$, with nodes partitioned into disjoint sets $V_1$ and $V_2$, has directed edges $(v, u) \in E_t$ with $v \in V_1$ and $u \in V_2$.

To define our signatures, we make use of a *relevancy function*, $w$, so that $w_{vu}$ indicates the relevance of $u$ to $v$. Initially, assume $w$ is given; we later discuss choices of $w$.

*Definition 1:* **(Graph Signature)** We define a communication graph signature $\sigma_t(v)$ for node $v \in V$ at time $t$ as a subset of $V$ with top-$k$ associated weights[4], that is,
$\sigma_t(v) := \{(u, w_{vu}) | u \neq v \in V, w_{vu} \geq w_v^{(|V|-k)}, w_{vu} \in \Re^+\}$,
where $k < |V|$; $w_v^{(i)}$ is the $i$th order statistic of $\{w_{vu} | u \in V\}$, that is, $w_v^{(1)} \leq w_v^{(2)} \leq \ldots \leq w_v^{(i)} \leq \ldots \leq w_v^{(|V|)}$.

When the graph is bipartite, we may restrict the signature for nodes in $V_1$ to consist only of nodes in $V_2$, especially if the size of the sets is unbalanced, i.e. $|V_1| \ll |V_2|$. Otherwise,

[1] However, we consider a group of people sharing a node, e.g., a family with a shared Internet connection, or even a computer program with a particular communication pattern, to represent an "individual" in our setting if the group membership is consistent over time.

[2] We use the terms "individuals" and "users" interchangeably; likewise, "labels" and "nodes".

[3] In practice, $V = V_t$ may vary between windows, but only by a small amount.

[4] The *top* weights follow naturally since $w$ quantifies node relevance, and thus filters out noise while pruning storage space.

the treatment of bipartite graphs is the same as that for general graphs.

We deliberately restrict the scope of the signature space to include only graph features. Although some prior work on related questions has used features which do not fit into this setting, such as the maker of the cellphone or the age of the blog user associated with a node, and those based on interarrival distributions [13], this definition is sufficiently broad to capture a large class of possible signature schemes. In many common settings only communication "flows" are revealed, in the form of graph edges aggregated over multiple occurrences and summarized as total volumes, such as Call Detail Records in telephony [5] and NetFlow for summarizing IP traffic at a router [20]. In addition, this definition conforms with prior work in [5]. Thus, this restriction allows us to thoroughly explore signature schemes in a well-defined, useful space. Moreover, this definition lends to more human comprehensible signatures, and simple descriptions of causes for differences.

The above definitions leave room for many alternatives. Designing a good signature requires much insight and care. We discuss how to select an appropriate set of nodes with associated weights ("signature scheme") in Section III. Next we introduce some general properties that are desirable for any signature, and discuss how they apply to a variety of problems.

### *C. Signature Properties*

The traditional function of a signature is to authenticate an individual's identity via handwritten depictions of his or her name. In our context, signatures are based on profiling interactions specific to the individual. As with the handwritten case, a useful communication signature should satisfy the following properties:

*Definition 2:* **(General Properties)**

- *Persistence*: an individual's signature should be fairly stable across time, that is, not differ much when comparing similarities at consecutive time intervals. Slowly evolving signatures may be acceptable but abruptly changing signatures are not; otherwise, it will not give a reliable way to identify the individual.
- *Uniqueness*: one individual's signature should not "match" another's (defined below). That is, if two signatures match, then they should belong to the same individual.
- *Robustness*: the ability to identify an individual from a signature should not be sensitive to small perturbations. Any noise introduced in the process of providing a signature should not interfere with its effectiveness.

To measure these properties and so be able to compare different signature schemes, we need a way to match identities based on signatures. A natural approach involves defining distance functions $\mathrm{Dist}(\sigma_1, \sigma_2)$ between two signatures $\sigma_1$ and $\sigma_2$. Then we can more precisely define and measure persistence in terms of the distance between a node's signature at two different time steps; uniqueness in terms of the distance between a given node's signature and that of another node in the graph; and robustness as the distance between a node's signature with and without small perturbations. That is, for a fixed $v$ we measure the three graph properties, given some node $u \neq v$, as follows (w.l.o.g., fix $0 \leq \mathrm{Dist}(\cdot,\cdot) \leq 1$):

- Persistence: $1 - \mathrm{Dist}(\sigma_t(v), \sigma_{t+1}(v))$
- Uniqueness: $\mathrm{Dist}(\sigma_t(v), \sigma_t(u))$
- Robustness: $1 - \mathrm{Dist}(\sigma_t(v), \hat{\sigma}_t(v))$, where $\hat{\sigma}_t(v)$ has been slightly perturbed from $\sigma_t(v)$.

These definitions can accommodate different choices for Dist and $\hat{\sigma}_t(v)$. We can now compare different signature schemes with respect to persistence, uniqueness and robustness using distance measures. These are defined so that a larger value in each case indicates greater presence of these properties, up to 1 (perfect).

| Applications | Persistence | Uniqueness | Robustness |
|---|---|---|---|
| Multiusage Detection | Low | High | High |
| Label Masquerading | High | High | Medium |
| Anomaly Detection | High | Low | High |

TABLE I

DIFFERENT APPLICATIONS AND THEIR REQUIREMENTS

Because of the hidden mapping from individuals to labels, some trivial signature schemes do not suffice. We could assign each node $v$ the signature $\sigma(v) = \{(v, 1)\}$: the signature is the node label. This is insufficient for persistence and uniqueness, since the signature relates only to the node, and not the individual: if the user changes, the signature of the node remains the same and so it fails.

### *D. Applying Signatures*

We now specify some example tasks that involve analyzing communication patterns, and discuss which properties of a signature (listed above) are needed to solve it. Table I summarizes these observations.

**Multiusage Detection.** Multiusage occurs when a single individual exhibits similar behavior via multiple node labels during the same time period; detecting such multiusage has also been called "Anti-Aliasing" [23]. This could be the result of malicious behavior such as in link spam where websites attempt to manipulate search engine rankings through aggressive interlinking to simulate popular content, or benign behavior such as a single individual communicating from multiple distinct node labels. The key signature property needed is uniqueness, since the assumption is that if nodes have distinct users then they have dissimilar signatures. To detect multiusage, we compute $\mathrm{Dist}(\sigma_t(v), \sigma_t(u))$ for node pairs within the $t$th time window, and look for high degrees of pairwise similarity.

**Label Masquerading.** Label masquerading occurs when one user switches all his or her communication from one node to originate from another. An example of this is the repetitive debtors problem [10], where a consumer switches accounts with no intention of paying for his or her usage. This has implications for data anonymization as a user who is effectively unable to masquerade is susceptible to anonymity intrusion.

| Characteristics | Properties |
|---|---|
| Engagement | persistence, robustness |
| Novelty | uniqueness |
| Locality | uniqueness |
| Transitivity | persistence, robustness |

TABLE II

COMMUNICATION GRAPH CHARACTERISTICS AND PROPERTIES

The key signature properties required here are persistence and uniqueness. On the assumption that such masquerades are relatively rare within the whole graph, to find instances we seek node pairs where there is very little or no similarity within one time window of interest, but very similar behavior in subsequent windows. Formally, the detection process involves computing the persistence values $1-\mathrm{Dist}(\sigma_t(v), \sigma_{t+1}(u))$, for each $v$, and uniqueness values of a fixed $v$ $\mathrm{Dist}(\sigma_t(v), \sigma_t(u))$, for each $u \neq v$. A masquerader who switches from $v$ to $u$ is likely to be detected when corresponding persistence and uniqueness values are both high.

**Anomaly Detection.** We define an anomaly as an abrupt and discernible change in the behavior of a fixed label $v$ observed in consecutive time windows. This change could be the result of malicious behavior such as fraud, or could be due to benign factors such as one individual going on vacation (resulting in a change in communication patterns). The key signature property that will be useful for detecting anomalies is persistence. Robustness is also needed, as we expect some noise and variation over time. Uniqueness is not as important here: we can tolerate some nodes have similar signatures, since we only compare signatures of the same node over time. A simple algorithm to detect anomalies from signatures is to compute value given by the above definition of persistence, $1-\mathrm{Dist}(\sigma_t(v), \sigma_{t+1}(v))$, for each $v$, and reporting those $v$ with unusually small values. Consequently, signatures that exhibit higher persistence over a longer term will be more effective at detecting anomalies.

## III. EXAMPLE SIGNATURE SCHEMES

The framework in Section II leaves a lot of scope for different signature schemes that satisfy the desired properties. In this section, we study different features of communication graphs that help us build useful signatures.

- *Engagement/Communication strength*: the edge weights in communication graphs indicate the amount of interaction between each pair. So a heavier edge should make the participating pair of nodes "closer" to each other, and hence more likely to figure in each other's signatures. Basing signatures on these larger weights should make the signatures robust to small perturbations. Further, we can assume that high interaction in one time period predicts high interaction in future time periods, and so will improve persistence.
- *"Novelty" of neighbors*: typically communication graphs exhibit a "power-law"-like distribution of node degrees, so a few nodes have very high degree, but the majority have smaller (constant) degree. A node with high in-degree in a graph may be a poor member of a signature, since it is not very discriminating. For example, a directory assistance number in the phone graph or a search engine in the web traffic graph may be used by many people, and hence be poor in distinguishing between them. So nodes with lower in-degree are more "specific", and may be preferable for uniqueness.
- *Locality*: because of the degree distribution, communication graphs are far from complete, and instead some nodes are much closer (in terms of graph hop distance) than others. For a given node, choosing nearby nodes may be more relevant than those that are far away, leading to increased distinguishability and hence uniqueness. In addition, a signature may be more human interpretable if it relates to nodes in the immediate neighborhood than seemingly arbitrary nodes scattered across the whole graph.
- *Transitivity/Path Diversity*: communication graphs, although not dense, are also far from being skeletal trees; between pairs of nodes there are typically many paths. We assume that the more connecting paths, the "closer" these two nodes are (even if they are not directly connected). That is, a signature is likely to be more persistent and robust if it relates node pairs with multiple connecting paths.

Table II summarizes the links between graph characteristics and our desired signature properties.

We now describe a variety of signature schemes. Most are quite simple to state, and based on extensions of prior work. We emphasize that our concern is not the novelty or otherwise of these signatures, but rather the evaluation within our framework, and the extensive experimental comparison which follows.

### A. One-hop Neighbors Based Approaches

We first consider signature schemes that only pick from the immediate (one-hop) neighbors in the graph. For each neighbor $j$ of $i \in V$, we compute a relevance measure $w_{ij}$, indicating the computed importance of $j$ to $i$. Following Definition 1, we retain the $k$ nodes $j$ with the largest values of $w_{ij}$. For bipartite graphs, for each $i \in V_1$, we retain the $k$ nodes $j$ among $V_2$ with the largest values of $w_{ij}$. Ties may be broken arbitrarily, and if there are fewer than $k$ nodes with non-zero values of $w_{ij}$, we retain only this subset.

*Definition 3:* The *Top Talkers (TT)* scheme sets $w_{ij} = C[i,j]/\sum_{(i,v)\in E_t} C[i,v]$. That is, the signature of $i$ consists of the (at most) $k$ nodes adjacent to $i$ with the highest incoming edge weights $w_{ij}$ from $i$.

This might correspond to the most called telephone numbers, or the most visited web sites, for $i$. The definition only takes into account Communication Strength, and is implicit in the "Communities of Interest" work, for detecting fraudulent activity [5]. A feature of that work was that it additionally created a signature from the combination of multiple time-steps by using an exponential decay function applied to older data. It is straightforward to apply these definitions over a set of modified edge weights $C'[i,j]$, which reflect an appropriate exponential decay or other combination of historical data. Hence, we treat such time decay as orthogonal to our main line of inquiry, and do not consider it explicitly any further.

*Definition 4:* The *Unexpected Talkers (UT)* scheme sets $w_{ij} = C[i,j]/|I(j)|$. Thus the signature for $i$ consists of the (at most) $k$ nodes $j$ with the largest incoming edge weights from $i$, scaled by the number of $j$'s incoming edges.

By factoring in "Novelty" of neighbors, this definition downweights nodes which might be universally popular and dominate signatures, leading to false matches and hence low uniqueness. The prevalence of such nodes will depend on characteristics of the setting inducing the communication graph. For example, there are relatively few nodes of this kind in the telephone call graph: although people may regularly call directory assistance, they will typically call friends and family more often, hence such nodes are unlikely to dominate their signature. However, in the web traffic graph, one can observe sites which attract a lot of incoming traffic, from many different users, such as search, web mail, and video sites. Having such nodes in a signature is unlikely to provide a good signature. One could remove such nodes altogether. However, there can be many such nodes, and the list evolves over time as new nodes attract interest. Secondly, there is still some information in the set, to create some signature even if these are the only destinations a node $i$ communicates with. Other functions of $|I(j)|$ and $C[i,j]$ are possible (e.g., $C[i,j]\log(|V|/|I(j)|)$, by analogy with the TF-IDF measure). In our detailed experiments, we did not see much variation in results for different scaling functions.

### B. Multi-hop Neighbors Based Approach

The one-hop approach is highly appropriate for certain graphs, e.g. the telephone call graph. But there are other communication graph settings where no one-hop signature can do well. Consider the (bipartite) communication graph induced by customers hiring movies. It is unlikely that any customer will rent the same title in two subsequent time periods. Thus, no matter how one-hop neighbors are weighted, signatures will have poor persistence. But we can at least hope for somewhat better signatures if we look beyond the immediate neighborhood.

For a multi-hop signature based approach to be successful, we need to be able to find nodes and weights outside the immediate (one-hop) neighborhood of node $i$ that nevertheless accurately represent $i$. So, even if $i$ communicates with completely different sets of nodes in each time period, our hypothesis is that there is sufficient information in the broader link structure of the graph so that we will find a set of nodes and weights for $i$ that are similar in both time periods (persistence) while being different to those found for other nodes (uniqueness). Clearly, the validity of this will depend on the nature of the communication graph. We propose an example signature scheme, and validate it experimentally on a variety of graphs.

| Scheme | Characteristics | Properties |
|---|---|---|
| TT | locality, engagement | uniqueness, robustness |
| UT | novelty, locality | uniqueness |
| RWR | transitivity, engagement | persitence, robustness |
| RWR$^h$ | locality, transitivity | persistence, uniqueness, robustness |

TABLE III

PROPERTIES USED BY SIGNATURE SCHEMES

*Definition 5:* The *Random Walk with Resets (RWR)* signature scheme is defined as follows: starting from node $i$, we define $\vec{w}_i = [w_{ij}]_{|V|\times 1}$ as the steady-state probability vector, where $w_{ij}$ is the probability that a random walk from $i$ occupies node $j \in V$. Each step in the random walk either selects an edge to follow with probability proportional to the edge weight or, with probability $c$, returns to node $i$.
As before, we take the $k$ largest $w_{ij}$s in $\vec{w}_i$ to define the signature for $i$. Although this is the stationary distribution of a random walk, it is can be computed exactly. The definition of $w_{ij}$ is equivalent to the personalized PageRank [9] with an input set of preferences equal to the single node $i$ and can be computed as follows.

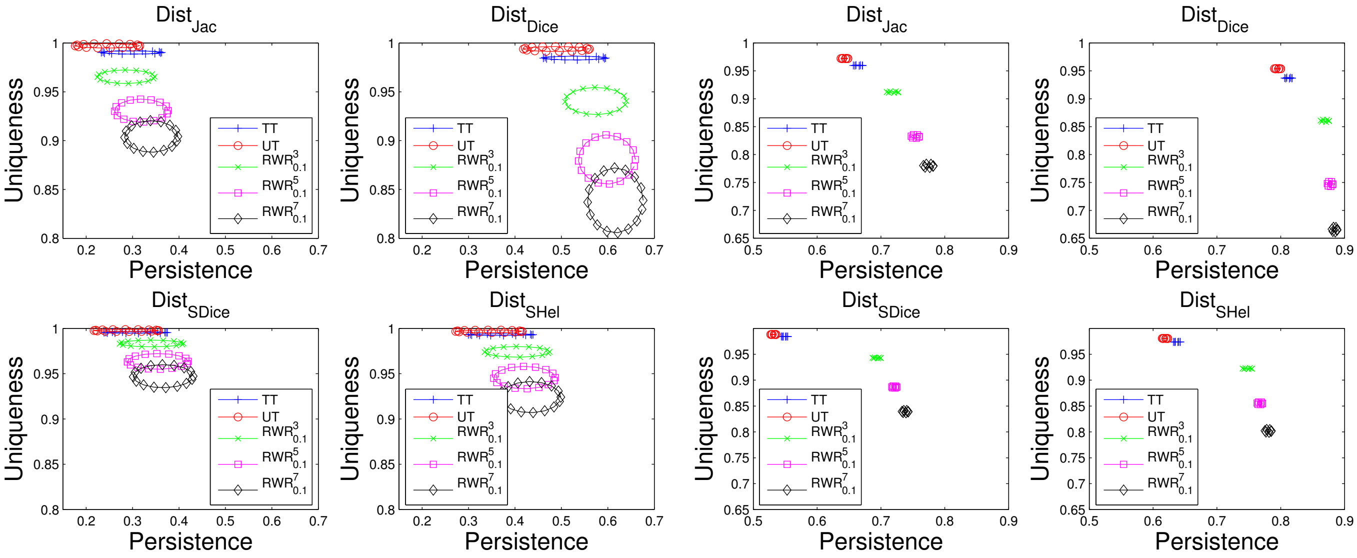
**Computation of RWR.** Recall that $C$ is the adjacency matrix of the graph $G_t$ from which we compute the transition matrix $P$. Here $P(i,j) = C[i,j]/\sum_{j=1}^{|V|} C[i,j]$ denotes the probability of taking edge $(i,j)$ from node $i$. Let $\vec{s}_i$ be the start-node vector with 1 in position $i$ and 0 elsewhere. Then the steady-state probability vector $\vec{r}_i$ can be solved by using the iterative approach $\vec{r}_i^{\,l} = (1-c)P\vec{r}_i^{\,l-1} + c\vec{s}_i$, where $\vec{r}_i$ is initialized to $\vec{s}_i$ and $c$ is the probability of resetting. This quickly converges [2], in time $O(|E|)$ per iteration.

**Computation of RWR$_c^h$.** RWR$_c^h$ modifies the above scheme by restricting the random walk to nodes at most $h$ hops from $i$. To compute RWR$_c^h$, we take the iterative algorithm defined above, and proceed for only $h$ iterations. When $c = 0$ and $h = 1$, RWR$^h$ is identical to the Top Talkers scheme. By increasing $h$, we tradeoff between the local (TT) scheme and the global (RWR) scheme.

Table III summarizes the schemes in terms of communication graph characteristics exploited and the resulting signature properties from Section II-C that are captured. Based on our analysis of application requirements, we reason that RWR will perform well at anomaly detection; RWR$^h$ will succeed at label masquerading, and TT will be good for multiusage detection.

## IV. EVALUATIONS OF SIGNATURE PROPERTIES

In this section, we evaluate the quality of signature schemes on various data sets with respect to persistence, uniqueness and robustness. In particular, we focus on two real data sets: flow data from an enterprise network; and database query logs. All experiments were performed on a dual 2.8GHz desktop machine with 2GB RAM. From each graph, we select signatures for each individual using the TT, UT and RWR schemes outlined in Section III.



(a) Network flow data

(b) User query logs

Fig. 1. Signature persistence and uniqueness on two real data sets.

### A. Data Sets

**Enterprise network data.** We collected six weeks' worth of flow records from a large enterprise network. LAN switches and a Network Interface Card were configured to monitor all traffic from more than 300 local hosts including desktop machines, laptops and some servers; these hosts are the focal point of our analysis. We captured all outgoing flows from the local hosts to external hosts in the network. No communications between local hosts are visible on the monitored links. In this study we used TCP traffic only, and removed weekend data from our data set for purpose of a more consistent per-day traffic mix. The total six week collection yielded more than 1.2 GB of network flow records, and contains about 400K distinct IPs. The flows were aggregated over regular time windows to form communication graphs. We used an interval of five days to present results; the results were similar with other window sizes. The weight of a directed edge was measured as the total number of TCP sessions during the time interval. In all experiments, we used the signature length of $k = 10$,[5] which is half of the average local host's out-degree.

**User query logs.** Our second data set consisted of 820K tuples summarizing a set of queries issued by users to a data warehouse. The logs recorded which tables were queried, but not the attributes accessed within each table. The data contains 851 distinct users and 979 distinct tables. Given a sequence of (userID, tableID) "edges", we split the trace into windows covering five consecutive time periods. Here, the edge weight is the number of times that the user accessed the table within the time period. In all experiments, we used a signature length of $k = 3$, half the average number of tables a user accessed per period.

### B. Distance Functions

In our evaluation, we employed a variety of distance functions to compare signatures.[6] They are generalized from known measures, and take into account both set overlap as well as weighted occurrence. Formally, given two signatures $\sigma_1$ and $\sigma_2$, where $\sigma_i = \{(u_{ij}, w_{ij})|j = 1..k_i\}$ is of length $k_i$, let $S_i = \{u_{ij}|j = 1..k_i\}$ be the set of $u$'s in $\sigma_i$. We considered four distance functions:

$$\begin{aligned}
\mathrm{Dist}_{\mathrm{Jac}}(\sigma_1, \sigma_2) &= 1 - \frac{S_1 \cap S_2}{S_1 \cup S_2};\\
\mathrm{Dist}_{\mathrm{Dice}}(\sigma_1, \sigma_2) &= 1 - \frac{\sum_{j \in S_1 \cap S_2}(w_{1j} + w_{2j})}{\sum_{j \in S_1 \cup S_2}(w_{1j} + w_{2j})};\\
\mathrm{Dist}_{\mathrm{SDice}}(\sigma_1, \sigma_2) &= 1 - \frac{\sum_{j \in S_1 \cap S_2}\min(w_{1j}, w_{2j})}{\sum_{j \in S_1 \cup S_2}\max(w_{1j}, w_{2j})};\\
\mathrm{Dist}_{\mathrm{SHel}}(\sigma_1, \sigma_2) &= 1 - \frac{\sum_{j \in S_1 \cap S_2}\sqrt{w_{1j} \cdot w_{2j}}}{\sum_{j \in S_1 \cup S_2}\max(w_{1j}, w_{2j})}.
\end{aligned}$$

It is easy to verify that all these distance functions yield values in $[0, 1]$. $\mathrm{Dist}_{\mathrm{Jac}}$ is based on Jaccard coefficient, where the node weights are not taken into account; it is minimized when $S_1 = S_2$, and it equals 1 when their overlap is empty. $\mathrm{Dist}_{\mathrm{Dice}}$ is an extension of the Dice criterion [10], which factors in node weights; $\mathrm{Dist}_{\mathrm{SDice}}$ can be thought of as a scaled version of $\mathrm{Dist}_{\mathrm{Dice}}$: it gives an added premium if the individual weights in $S_1$ and $S_2$ are similar. By using $\min$ in the numerator, however, we may be penalizing too much for non-equal individual weights, since all that matters is the smaller one rather than some combination of the two. $\mathrm{Dist}_{\mathrm{SHel}}$ overcomes this based on Hellinger distance [10].

[5] Due to space limitations, we omit discussion about how we chose $k$. This issue was investigated in [10], and is beyond the scope of this paper.

[6] These functions were chosen based on their simplicity and naturalness, though other functions are certainly suitable.

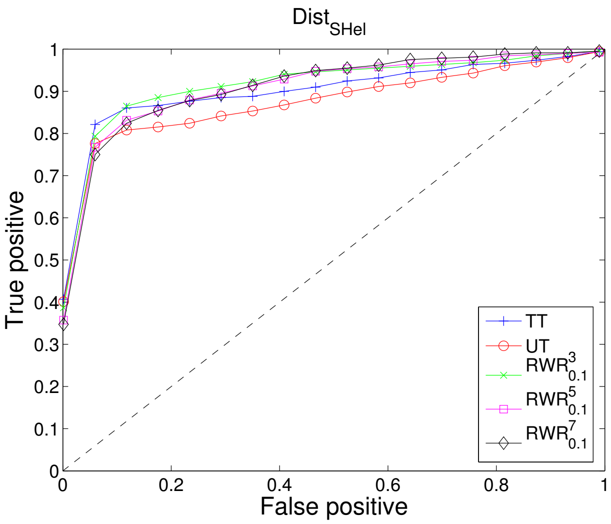


Fig. 2. ROC curves from network data.

| *AUC* | TT | UT | $\text{RWR}^{3}_{0.1}$ | $\text{RWR}^{5}_{0.1}$ | $\text{RWR}^{7}_{0.1}$ |
|---|---|---|---|---|---|
| $\text{Dist}_{\text{Jac}}$ | 0.9086 | 0.8827 | 0.9177 | 0.9087 | 0.9052 |
| $\text{Dist}_{\text{Dice}}$ | 0.9093 | 0.8826 | 0.9256 | 0.9172 | 0.9167 |
| $\text{Dist}_{\text{SDice}}$ | 0.9035 | 0.8812 | 0.9207 | 0.9086 | 0.9066 |
| $\text{Dist}_{\text{SHel}}$ | 0.9094 | 0.8827 | 0.9238 | 0.9162 | 0.9173 |

(a) AUC from network flow data.

| *AUC* | TT | UT | $\text{RWR}^{3}_{0.1}$ | $\text{RWR}^{5}_{0.1}$ | $\text{RWR}^{7}_{0.1}$ |
|---|---|---|---|---|---|
| $\text{Dist}_{\text{Jac}}$ | 0.9935 | 0.9969 | 0.9901 | 0.9882 | 0.9877 |
| $\text{Dist}_{\text{Dice}}$ | 0.9935 | 0.9969 | 0.9901 | 0.9882 | 0.9877 |
| $\text{Dist}_{\text{SDice}}$ | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| $\text{Dist}_{\text{SHel}}$ | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |

(b) AUC from user query logs.

Fig. 3. AUC across different signature schemes.

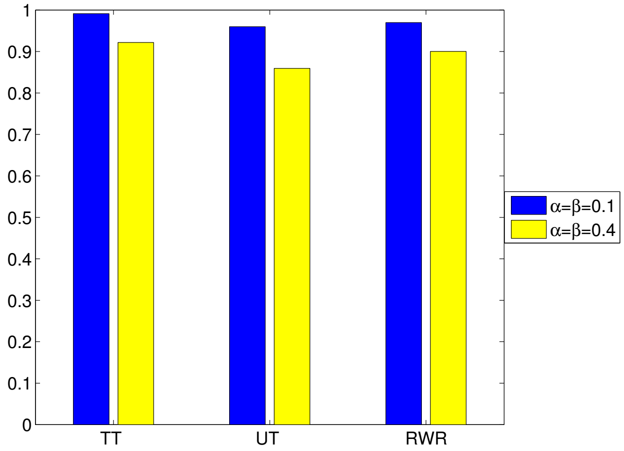


Fig. 4. Robustness on network data

### C. Experimental Results

**Signature persistence and uniqueness.** For each $t$, we summarize the persistence (resp. uniqueness) values using $\mu_p(t)$, $s_p(t)$ — the mean and standard deviation of $\{\text{persistence}_v(t)|v \in V\}$ (resp. $\mu_u(t)$, $s_u(t)$ — the mean and standard deviation of $\{\text{uniqueness}_{v,u}|v, u \in V, v \neq u\}$). We display the "span" of persistence and uniqueness values as an ellipse: its center is at $(\mu_p(t), \mu_u(t))$; $s_p(t)$ and $s_u(t)$ are the respective (x and y) diameters. Over all different time periods we observed very similar results. Figure 1 illustrates results from one time window in depth. We present results from TT, UT and $\text{RWR}^h_{0.1}$ with $h = 3, 5, 7$ and observe that TT lies between UT and $\text{RWR}^h_{0.1}$ in the plots, for both data sets and all distance functions. [7] This is consistent with our intuition that UT downweights universally popular nodes to enhance uniqueness; $\text{RWR}^h_{0.1}$ selects most relevant nodes to $i$ from beyond $i$'s immediate neighborhood to represent it persistently.

The above figures compare the signature schemes separately in terms of persistence and uniqueness but do not capture the trade-off between the two in a single statistic. For this, we use ROC Curves, a standard measure in statistics [17]. Given $G_t$ and $G_{t+1}$, for each node $v$ we computed $\text{Dist}(\sigma_t(v), \sigma_{t+1}(u))$ for all $u \in V$, and returned a ranked list, where $u$ with a smaller Dist-value to $v$ was ranked higher. Our hypothesis is that across time, a node behaves more similar to itself than to others. Therefore in $v$'s ranked list, $v$ should ideally be ranked the first. The ROC curve starts at the origin $(0, 0)$ and traverses the ranked list of nodes from the top. If the element is $v$, the ROC curve goes up by a step of 1; otherwise, the ROC curve goes to the right by a step of $1/(|V| - 1)$. That is, the $x$-axis is false positives and $y$-axis is true positives. We can then compute the Area Under the ROC Curve (AUC). If the AUC is 0.5, the signature scheme is no better than random selection; higher AUC values indicate better accuracy, up to 1 (perfect). We report the average AUC over all $v$'s. Figure 2 shows the results on the flow data using $\text{Dist}_{\text{SHel}}$; ROC curves from other distance measures look very similar.

Figure 3(a) summarizes AUC across different signature schemes per distance measure for the flow data. The multi-hop neighbors based schemes achieved better AUCs than their one-hop counterparts. Among $\text{RWR}^h_{0.1}$ schemes, $\text{RWR}^3_{0.1}$ outperformed the other two. A further observation is that the difference between the AUC from $\text{RWR}^5_{0.1}$ and $\text{RWR}^7_{0.1}$ is small enough to be ignored. Other experiments (not shown) with $\text{RWR}^h_{0.1}$ for $h > 7$ all converged to $\text{RWR}^7_{0.1}$, suggesting that having more than 5 hops does not bring in drastically "new information". This is due in part to the graph having a small diameter: for all $h$ larger than the diameter of the graph, $\text{RWR}^h$ coincides with $\text{RWR}^\infty$, the unbounded random walk. We repeated the experiments on user query logs, and summarize AUC values in Figure 3(b). All signature schemes behave almost equally well on this data set (almost perfectly), with UT being slightly better than the others. In what follows we use $\text{RWR}^3_{0.1}$ as the best representative of the RWR schemes, and do not show results for other parameter settings.

**Signature robustness.** To evaluate the robustness of the signature schemes, we randomly inserted and deleted edges to obtain a perturbed graph $G'_t$. Let $\sigma(v)$ and $\hat{\sigma}(v)$ denote $v$'s

[7] When $c$ is as large as 0.9, $\text{RWR}_c$ scheme converges to TT, so we focus on small $c$ values, such as $\text{RWR}^h_{0.1}$.

|  | TT | UT | RWR |
|---|---|---|---|
| *persistence* | medium | low | high |
| *uniqueness* | medium | high | low |
| *robustness* | high | low | medium |

TABLE IV
RELATIVE BEHAVIOR OF THE SIGNATURE SCHEMES.

signatures constructed from $G_t$ and $G'_t$, respectively. Given a bipartite graph $G_t$ and parameter $\alpha$, we inserted $\alpha|E_t|$ new edges. First, a node $v' \in V_1$ was sampled proportional to its outdegree, that is, with probability $|O(v')|/\sum_v |O(v)|$. Then a node $u' \in V_2$ was sampled proportional to its indegree, that is, with probability $|I(u')|/\sum_u |I(u)|$. The weight of $(v, u)$ (initially 0 if edge $(v, u)$ did not previously exist) was assigned independently of $C[v, u]$, but from the total distribution of all edge weights rather than uniformly. For deletions, we sampled existing edges proportional to their edge weights and decremented the weight by one unit, repeating $\beta|E_t|$ times.

Since our interest in this paper is in identity matching, we are interested in whether a signature is more similar to its perturbed self than to signatures of other nodes. We again used ROC curves to investigate this and used each $v \in V$ in $G_t$ as a query against $V$ in $G'_t$, reporting the AUC values in Figure 4 for two different parameter settings: $\alpha = \beta = 0.1$ and $\alpha = \beta = 0.4$. TT was the most robust, followed by RWR. UT was the least robust, which is to be expected due to nodes with high indegree (and thus high frequency) being discounted, although the relative difference between all methods is very small.

**Summary.** Table IV summarizes the relative behavior of the signature schemes. We observe an interesting trade off between the three considered schemes: none strictly dominates any other over all three properties. Next we see a clearer separation when applying signatures, which emphasize the properties to differing degrees.

## V. APPLICATION EVALUATION

We discuss two applications in detail, and evaluate them empirically on enterprise network flow data.

**Multiusage Detection.** Recall the discussion of multiusage detection in Section II-D. With network flow data, the problem is to find the set of IPs being multiple connection points (home, office, wireless hotspot) per individual. Our algorithm to detect such an IP set containing $v$ computes the uniqueness values $\text{Dist}(\sigma(v), \sigma(u))$ for all nodes $u$ observed within the same time window. We report those nodes $u$ with low Dist-values (high similarity).

To evaluate the use of signatures for this task, we obtained additional data mapping users to their registered IP addresses, and identified the set of users $U$ who made use of multiple addresses within the enterprise network (of course, this ground truth is not available to the signature-based algorithms). For each user $u \in U$, we denote its set of registered IPs as $S_u$. We expect a signature with high uniqueness and robustness to be best. Therefore, our hypothesis is that in one communication graph, signatures for IPs belonging to the same user look more similar to each other, compared to the pairwise similarities between IPs of different users.

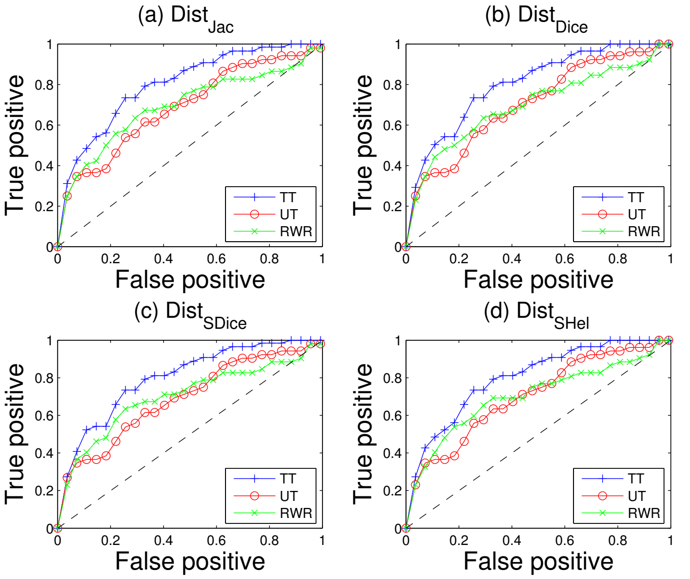


Fig. 5. Multiusage detection: ROC curves

For each $v \in S_u$ $(u \in U)$, we computed $\text{Dist}(\sigma(v), \sigma(w))$ for all $w \in V$, and derived a ranked list of $V$ sorted by these distances. From these we produced an average ROC curve over all $v \in \bigcup_{u \in U} S_u$, starting at the origin $(0, 0)$. When we traverse the ranked list from the top, if a node is in $S_u$, the ROC curve goes up by a step of $1/|S_u|$; otherwise, the ROC curve goes to the right by a step of $1/|V - S_u|$. So the $x$-axis measures false positives and $y$-axis true positives. If the hypothesis is correct, and we can use signatures for this task, then the IPs in $S_u$ should be ranked higher than others. We plot the results across the various schemes in Figure 5. Across all distance functions, TT consistently dominates the other two schemes. This agrees with our prediction in Section III that multiusage detection calls for TT, due to its emphasis on uniqueness and robustness.

**Label Masquerading.** For this problem, we simulated masquerading by perturbing $f|V|$ randomly selected nodes (denoted $P$) in $V$, for some fraction $f$. We created a bijective mapping between nodes in $P$, and applied this mapping to the communications. We denote the mapping as $E_P = \{(v, u)|v, u \in P\}$, where $(v, u)$ means that $v$ (and all of $v$'s communications) are relabelled with $u$. Given graphs $G_t$ and $G_{t+1}$ from consecutive time periods, a pair $(v, u) \in E_P$ means that node $v$ in $G_{t+1}$ is relabelled with $u$, while $v$'s label in $G_t$ remains unchanged. We evaluate our methods on how well they are able to recover $E_P$.

Based on the discussion in Section II-D, the detection algorithm is given in pseudo-code in Algorithm 1. Here $M$ returns the set of local hosts not identified as masqueraders; $O_P$ is the estimate for $E_P$. We see that an output $(v, u)$

**Algorithm 1** DETECTLABELMASQUERADING($G_t, G_{t+1}$)

```
1: Init $M := \emptyset$, $O_p := \emptyset$
2: for each $v \in V$ do
3:   if $1 - \text{Dist}(\sigma_t(v), \sigma_{t+1}(v)) > \delta$ then
4:     $M := M \cup \{v\}$
5:   else
6:     $\forall u \in V,\ A[u, v] := 1 - \text{Dist}(\sigma_t(v), \sigma_{t+1}(u))$
7:     if $\exists u \neq v$, $A[v, u]$ is among $v$'s top-$\ell$ largest and $A[u, u] \leq \delta$ then
8:       $O_P := O_P \cup \{(v, u)\}$
9:     else
10:      $M := M \cup \{v\}$
```

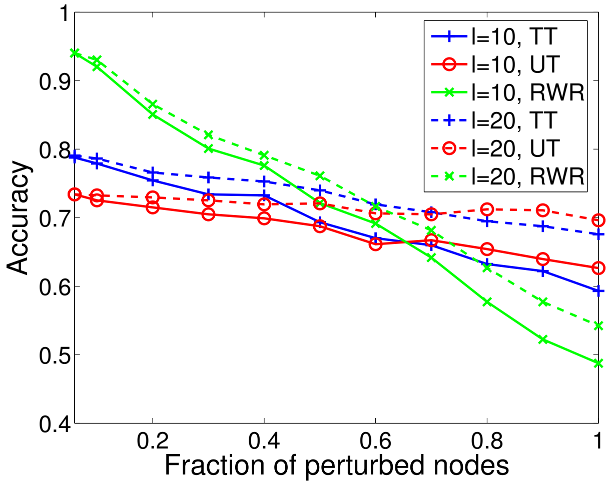


Fig. 6. Accuracy of label masquerading detection.

satisfies two conditions: (1) both $v$ and $u$ look different from themselves across time (i.e., low persistence values, from Step 3 and 7 in Algorithm 1); but (2) they look more similar to each other than to others (i.e., high persistence between themselves, from Step 7). We evaluate the various signature schemes for this problem based on the standard information retrieval criterion of accuracy $\frac{|M \cap (V-P)| + |O_P \cap E_P|}{|V|}$, which measures the percentage of correctly classified hosts, labeled either as "non-suspect" (i.e., $v \notin P$) or with the new label of the node. This combines notions of false positives and false negatives.

In our algorithm, the persistency threshold $\delta$ should be a good cutoff between local hosts whose signatures look persistent and those who are not. Based on empirical results, we set $\delta$ to $\frac{\sum_{v \in V}(1 - \text{Dist}(\sigma_t(v), \sigma_{t+1}(v)))}{c|V|}, c \in \mathbb{N}$, as a fraction of the average self-similarity across time (scaled by $c$). In particular, we considered $c = 3, 5, 7$ in our experiments, and observed very similar results. Figure 6 compares the accuracy of various schemes with $c = 5$, for various $\ell$-values, as a function of fraction of the nodes perturbed.

As expected, accuracy increase as $\ell$ increases. Label masquerading should only affect a small fraction of nodes, so we focus our discussion and conclusions on lower values of $f$. In this range, the RWR scheme outperforms TT and UT. This coincides with our expectations, since our analysis of this application indicated that label masquerading requires a signature with high persistence and high uniqueness. According to Figure 2, which evaluates signature schemes on these measures with network data, RWR is the method of choice.

## VI. EXTENSIONS

In general, communication graphs can become extremely large (e.g., the graph of all phone calls or internet connections made over the course of a week). Here, we outline some of the scalability issues that arise under such massive data.

**Scalable signature computation.** When the communication graphs are large, even storing the graph can become infeasible. Instead we need compact data structures that can process an observed sequence of communications (defining edges in the graph), from which we can extract (approximate) signatures. Our assumption is that although the total volume of edges is too massive, we can at least record some constant amount of information about each node in turn: this is the "semi-streaming" model of graph stream processing [19]. Any given signature scheme will need a different approach in this restricted model, here we outline methods for the signature schemes used here for illustration. For the Top Talkers, we need to find the approximate heaviest-weight neighbors of node $i$. If the communication is pre-aggregated, we can just keep a heap of heavy edges; but more realistically, we see each individual communication, and want to recover the most frequent. We can use summary structures such as a CM sketch for each node to find its heaviest outgoing edges, and hence its signature [3]. For Unexpected Talkers, the situation is more complex. Here, we can additionally keep an FM sketch for each node, to find its incoming degree [7]. To find the signature for a node $i$, we can use the CM sketch to estimate $C[i, j]$, and the FM sketch to estimate $|I(j)|$ for each node $j$; combining these gives an approximation of $C[i, j]/|I(j)|$ as required. For schemes based on Random Walk with Reset, there is less prior work to draw on. Techniques in [25] give approaches to make the comptuations more scalable, based on appropriate block-wise decompositions of the graph; extending these to the full semi-streaming model remains an open problem.

**Scalable signature comparison.** When there are large number of nodes with signatures, applications based on comparing many signatures together become expensive (potentially quadratic in the number of nodes). At the heart of many applications discussed above is the problem of, given a signature, finding the most similar signature(s) from a particular (sub)set. This fits the set up of the nearest neighbor problem. Even for moderately small signature sizes, this can become expensive, and so we can turn to approximate nearest neighbor algorithms. Here, different approaches are needed for each different distance function, rather than signature scheme. For example, efficient solutions exist where the distance function is the Jacard distance, by using an approach based on Locality Sensitive Hashing [14].

## VII. RELATED WORK

Signatures (and fingerprints) were classically studied as an application of statistical pattern recognition, an area of study with a long history of techniques for feature selection and classification [15]. However, communication graphs give more contextual information to work with over such generic techniques.

Usage profiling in graphs (and networks) has been studied in multiple settings [22], [27], [16], [25], [13], [21]. However, the goal of these works is to model behavior aggregated at the level of the entire graph to detect network-wide anomalies. In [26], the so-called usage entropy of each IP address is computed, but this is for identifying dynamic IP address ranges rather than profiling single users. Usage profiling at the granularity of the *individual* has been studied for activity monitoring [6], for user recognition [24], and for enterprise security [18]. The approach uses complex rules on records of individual activity requiring detailed data storage.

Cortes *et al.* initiated the study of "COI-based" signatures, which makes use of communication graph topology to design individual's signatures in a concise way for detecting fraudulent users [4], [5], identifying repetitive debtors [10] and predicting links for viral marketing [12]; similar techniques were also applied to identify authors from bibliographical citations in [11]. However, they only focused on particular signatures and their applicability to the motivating tasks. We do not know of any prior work that proposed a principled approach such as ours for a detailed classification of properties of signatures and studying applications based on what properties they need for signatures to be useful.

## VIII. Conclusions and Future Work

We have attempted to take a very general approach to problems of defining and analyzing signatures in communication graphs. We proposed a framework for understanding and analyzing such signatures based on the three fundamental and natural properties of persistence, uniqueness and robustness. We justified these properties by showing how they impact a broad set of applications. We explored several signature schemes in our framework and evaluated them on real data in terms of these properties. In particular, our study on two concrete applications demonstrate their effectiveness in practice. This study underlined the fact that there is not one single signature scheme which is good for all applications, but rather that different signatures are needed, depending on what balance of the three properties they provide. We believe that a larger suite of properties of signatures are needed for the space of all applications that signatures will be useful for.

We have highlighted issues of scalability for building and applying signatures as an important extension; as communication graphs grow ever larger, it will be increasingly vital to ensure that these schemes scale to such massive settings. It remains the case that finding suitable signatures for any task is more of an art than a science, with effectiveness determined experimentally. In this sense, our proposal of specific signatures for communication graphs and their application to the specific tasks is such a study. But beyond this, our framework is general and can be applied broadly. One significant challenge of practical importance will be to automate this process to the extent possible.

## Acknowledgment

We would like to thank Theodore Johnson and Oliver Spatscheck at AT&T for their help to collect data.

## References

[1] J. Baumes, M. Goldberg, M. Hayvanovych, M. Magdon-Ismail, W. Wallace, and M. J. Zaki. Finding hidden group structure in a stream of communications. In *ISI*, 2006.

[2] P. Berkhin. A survey on pagerank computing. *Internet Mathematics*, 2(1):73–120, 2005.

[3] G. Cormode and S. Muthukrishnan. An improved data stream summary: The count-min sketch and its applications. In *LATIN*, 2004.

[4] C. Cortes and D. Pregibon. Signature-based methods for data streams. *Data Min. Knowl. Discov.*, 5(3):167–182, 2001.

[5] C. Cortes, D. Pregibon, and C. Volinsky. Communities of interest. *Lecture Notes in Computer Science*, 2189, 2001.

[6] T. Fawcett and F. Provost. Activity monitoring: Noticing interesting changes in behavior. In *SIGKDD*, 1999.

[7] P. Flajolet and G.N. Martin. Probabilistic counting. In *FOCS*, 1983.

[8] D. Gibson, R. Kumar, and A. Tomkins. Discovering large dense subgraphs in massive graphs. In *VLDB*, 2005.

[9] T. H. Haveliwala. Topic-sensitive pagerank. In *WWW*, 2002.

[10] S. Hill, D. Agarwal, R. Bell, and C. Volinsky. Building an effective representation for dynamic network. *Computational and Graphical Statistics*, 15(3):584–608(25), 2006.

[11] S. Hill and F. Provost. The myth of the double-blind review? Author identification using only citations. *SIGKDD Explorations*, 5(2):179–184, 2003.

[12] S. Hill, F. Provost, and C. Volinsky. Network-based marketing: Identifying likely adopters via consumer networks. *Statistical Science*, 21(2):256–276, 2006.

[13] A. Hussain, J. Heidemann, and C. Papadopoulos. Identification of repeated DOS attacks. In *INFOCOM*, 2006.

[14] P. Indyk, and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *STOC*, 1998.

[15] A. Jain, R. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[16] A. Lakhina, M. Crovella, and C. Diot. Diagnosing network-wide traffic anomalies. In *SIGCOMM*, 2004.

[17] S. Mason and N. Graham. Areas beneath the relative operating characteristics (ROC) and relative operating levels (ROL) curves: Statistical significance and interpretation. *Q. J. R. Meteorol. Soc*, 30:291–303, 1982.

[18] P. McDaniel, S. Sen, O. Spatscheck, J. van der Merwe, B. Aiello, and C. Kalmanek. Enterprise security: A community of interest based approach. In *NDSS*, 2006.

[19] S. Muthukrishnan. Data streams: Algorithms and applications. *Foundations and Trends in Theoretical Computer Science*, 1(2), 2005.

[20] Cisco netflow. `http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.htm`.

[21] J. Newsome, B. Karp, and D. Song. Polygraph: Automatically generating signatures for polymorphic worms. In *Symposium on Security and Privacy*, 2005.

[22] C. Noble and D. Cook. Graph-based anomaly detection. In *SIGKDD*, 2003.

[23] J. Novak, P. Raghavan, and A. Tomkins. Anti-aliasing on the web. In *WWW*, 2004.

[24] D. Song, P. Venable, and A. Perrig. User recognition by keystroke latency pattern analysis. *Technical Report*, 1997.

[25] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos. Relevance search and anomaly detection in bipartite graphs. *SIGKDD Explorations Special Issue on Link Mining*, 7(2):48–55, 2005.

[26] Y. Xie, F. Yu, K. Achan, E. Gillum, M. Goldszmidt and T. Wobber How Dynamic are IP Addresses In *SIGCOMM*, 2007.

[27] K. Xu, Z. Zhang, and S. Bhattacharyya. Profiling internet backbone traffic: behavior models and applications. *SIGCOMM Comput. Commun. Rev.*, 35(4):169–180, 2005.