Anomaly Detection using Proximity Graph and PageRank Algorithm

Zhe Yao, Philip Mark and Michael Rabbat

***Abstract*—Anomaly detection techniques are widely used in a variety of applications, e.g., computer networks, security systems, etc. This paper describes and analyzes an approach to anomaly detection using proximity graphs and the PageRank algorithm. We run a variant of the PageRank algorithm on top of a proximity graph comprised of data points as vertices, which produces a score quantifying the extent to which each data point is anomalous. Previous work in this direction requires first forming a density estimate using the training data, e.g., using kernel methods, and this step is very computationally intensive for high-dimensional data sets. Under mild assumptions and appropriately chosen parameters, we show that PageRank produces point-wise consistent probability density estimates for the data points in an asymptotic sense, and with much less computational effort. As a result, big improvements in terms of running time are witnessed while maintaining similar detection performance. Experiments with synthetic and real-world data sets illustrate that the proposed approach is computationally tractable and scales well to large high-dimensional data sets.**

***Index Terms*—Anomaly Detection, Proximity Graph, Personalized PageRank**

I. INTRODUCTION

Anomaly detection, also known as outlier detection, refers to the problem of discovering data points or patterns in a given dataset that do not conform to some normal behaviour. Anomaly detection techniques are applied in a variety of domains, including credit card fraud prevention, financial turbulence detection, virus or system intrusion discovery, and network monitoring, to name a few. For a broad review of different anomaly detection approaches and techniques, see the comprehensive survey by Chandola et al. [1].

We can view anomaly detection as a binary classification problem, with one class being anomalous and the other normal. In the classic supervised learning literature, labeled training data from both classes are required for the construction of a classifier. However, anomaly detection is different from traditional classification problems. While the latter usually deal with the case where both classes are of relatively equal size, this is not the case in anomaly detection. Since anomalies, by definition, deviate from the normal pattern, they usually represent situations where something goes wrong with the system (e.g., a malfunction, misuse, or malevolent behavior), and thus they are rarely observed. It is often impractical to collect sufficient observations to learn the anomalous pattern accurately. Moreover, manual labelling each data point is time consuming and error prone, and when the data is difficult to visualize or interpret it may not even be possible for a human to identify all anomalies. Therefore, although the supervised approach is well defined and thoroughly investigated, it is not a always appropriate for practical use. For this reason, we focus on unsupervised approaches.

In this work, we propose an unsupervised anomaly detection scheme using proximity graphs and the PageRank algorithm. Our algorithm takes as input a set of unlabeled data points and determines a ranking of which are most anomalous. We construct a proximity graph from data measurements, with one node for each data point and edges between nodes indicating similar data points. We then examine the stationary distribution of a random walk on this graph, following a variation of the well-known PageRank [2] algorithm. The stationary distribution of the random walk is used as a surrogate for density estimates at the locations of data points, allowing us to bypass running more intensive kernel density estimation procedures and leading to a faster and more scalable algorithm.

A. RELATED WORK

The standard approach in unsupervised statistical anomaly detection has been to assume that the data are drawn from a mixture of outlier and nominal distributions, and to estimate level sets of the nominal density. Schölkopf et al. [3] propose the one-class support vector machine (OCSVM) to learn the classification boundary where only nominal training data are available. Scott and Nowak [4] extend the Neyman-Pearson hypothesis testing framework to general supervised learning problems. Based on this extension, they derive a decision region using minimum volume (MV) sets in [5], providing false alarm control. Later, Scott and Kolaczyk [6] generalize this hypothesis testing framework to the unsupervised case, where measurements are no longer assumed to come from the nominal distribution alone. Meanwhile, they incorporate a multiple testing framework, where the false discovery rate is controlled rather than false alarm errors. Hero [7] introduces geometric entropy minimization to a extract minimal set covering the training samples while also ensuring false alarm guarantees. All of the methods mentioned above involve intensive computation, which is undesirable especially for large, high-dimensional data. We address this problem by taking an alternative graph-based approach.

Another line of previous work is based on forming a graph from the data using the distances between data points. For

The authors are with Electrical and Computer Engineering Department, McGill University, Montréal, QC, Canada. Email: zhe.yao@mail.mcgill.ca, philip.mark@mail.mcgill.ca, michael.rabbat@mcgill.ca

example, a k-nearest neighbor (kNN) graph is constructed first, and then the distances from each data point to its kth nearest neighbour are used to identify anomalies. These distances are ranked in descending order, and either a threshold is applied [8] or the top m candidates are declared anomalous [9]. Breunig et al. [10], [11] define a related quantity called local outlier factor, which is a degree depending on how isolated one data point is with respect to the surrounding neighborhood, to better accommodate heteroscedastic data sources. Pokrajac et al. [12] extend the local outlier factor approach in an incremental online fashion. Zhao and Saligrama [13] propose a non-parametric anomaly detection algorithm based on kNN graphs trained using only nominal data points, which provides optimal false alarm control asymptotically.

Our work is motivated by both directions mentioned above. We combine the graph approach together with random walk models, providing false alarm controls in an asymptotic sense. We note that we are not the first to use random walks or the PageRank algorithm for anomaly detection. Janeja and Atluri [14] apply random walk models to detect anomalous spatial area regions in graphs where, in contrast to conventional scan-statistic methods, a regular-shaped scan window (e.g., a rectangle) is no longer required. He et al. [15] propose a graph-based anomaly detection algorithm in an active learning setting, where the density information is used to reduce the number of inquiries made to the oracle; their algorithm builds on earlier work [16] which uses graph-based methods for density estimation. Cheng et al. [17] exploit random walks for finding anomalies in time sequences. Sun et al. [18] also investigate anomalous patterns using a PageRank-like method. However, they focus mainly on bipartite graphs, while we are discussing much more general distributions and graphs. Noble and Cook [19] develop methods to identify anomalous substructures in graph, purely based on the graph structure, and Chakrabarti [20] focuses on identifying anomalous edges in graphs. In contrast, we aim to find anomalous nodes in a graph induced by high dimensional measurements.

B. PAPER ORGANIZATION

The paper is structured as follows. After defining formally the problem in Section II, we present our approach at a high level in Section III. The two conceptual phases of our method are introduced and discussed separately. Then our main algorithm is described in Section IV, together with its properties. An upper bound on the computational complexity and statistical performance guarantees are discussed in Sections V and VI respectively. In Section VII, we evaluate the performance of our framework on both synthetic datasets and data from real world applications. Comparisons with other algorithms are also presented. Conclusions and possible extensions are included in Section VIII.

II. PROBLEM STATEMENT

In the anomaly detection literature, it is quite common to assume that observations come from one of two distinct classes: one represented by a nominal distribution $p(x)$ and the other by an anomalous distribution $\mu(x)$. We observe independent and identically distributed (i.i.d.) random measurements $x_i \in \mathbb{R}^d, i = 1, 2, \ldots, n$ from the mixture of these two, i.e.,

$$x_i \sim Q(x) = (1-\pi)p(x) + \pi\mu(x), \tag{1}$$

where π is the prior probability of a particular observation coming from the anomalous distribution. Our task is to assign a label, either nominal or anomalous, to each measurement x_i, possibly along with some confidence levels or rankings.

In this work, we follow the assumption that all the observations are from the mixed distribution $Q(x)$. The only quantities available are the measurements $x_i \in \mathbb{R}^d$ themselves. No additional training data or label information are given, and neither the nominal distribution $p(x)$ nor the prior probability π in Eq. (1) is provided. The anomalous distribution $\mu(x)$ is assumed to be uniform, which is a natural choice when no other information is available. Additionally, this assumption leads to a nice reduction to the Neyman-Pearson test as we see next.

In MV set approaches, an observation x_i is declared to be an anomaly if it falls outside of a particular level set [6], [21] of $Q(x)$, i.e.,

$$x_i \in \{x \mid Q(x) \leq \lambda\}, \tag{2}$$

with λ being a prescribed threshold. It turns out that this criterion is identical to the Neyman-Pearson test, under the assumption that $\mu(x)$ is uniform over the measurement space. To show this, combining Eq. (1) and Eq. (2), we have

$$(1-\pi)p(x) + \pi\mu(x) \leq \lambda \tag{3}$$

$$\frac{p(x)}{\mu(x)} \leq \frac{\lambda - \pi\mu(x)}{(1-\pi)\mu(x)}, \tag{4}$$

which is the likelihood ratio between the two distributions. Note that the right hand side of the last inequality remains constant under the assumption that $\mu(x)$ is uniform over x.

If, somehow, $p(x)$ can be estimated from the data, then we have all the information needed to compute density levels, to perform hypothesis testing, or to make other statistical arguments. Consequently, some form of density estimation (e.g., kernel density estimation) seems to be a natural prerequisite for our task. However, kernel density estimation is itself an unnecessary intermediate step which estimates the continuous density for the whole data domain from discrete points, after which one level set parameter is calculated for each data point. The quantities we are actually interested in are the properties of the discrete observations, not the continuous space around them. As a result, specifying the full distribution throughout the whole space is inessential, introducing computational burden while accumulating estimation errors. The approach proposed in this paper circumvents density estimation through the used of an alternative graph-based approach. Before introducing the proposed method, preliminary background material is discussed in the next section.

III. PRELIMINARIES

We propose an alternative approach to anomaly detection, using random walks on graphs, and more specifically, an adaptation of the PageRank algorithm [2], to sidestep the

tedious intermediate stage of kernel density estimation. The basic idea is to first construct a graph induced by the data set, and then to examine the stationary distribution of a random walk on this graph.

The main challenge of this approach lies mainly in the first step, constructing the graph from the data set. The graph must be able to capture enough information about the density of the data points. At the same time, graph construction should require as little computational burden as possible. The following subsections discuss the two phases respectively in detail.

A. PROXIMITY GRAPHS

Proximity graphs [22] are widely used in machine learning, e.g., for clustering [23], manifold learning [24] and semi-supervised learning [25]. Given a cloud of points in a Euclidean space, the proximity graph becomes an intermediate representation of the similarities between each pair of points. More formally, given n points $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$, a proximity graph is a weighted graph $G = (V, E)$ with vertices $x_1, \ldots, x_n$. Here V denotes the vertex set, E denotes the edge set. If there exists an edge between x_i and x_j, then the weight of this edge is denoted as $w_{ij} \geq 0$, which measures the similarity between the end nodes. Different geometric requirements lead to a variety of proximity graphs. The following definitions cover the three types of graphs we will consider in this work. We denote by $dist(i, j)$ a distance metric between x_i and x_j. Throughout, we take $dist(i, j)$ to be Euclidean distance unless otherwise noted.

Definition 1 (kNN Graph)**.** *A k nearest neighbour (kNN) graph is a graph $G = (V, E)$ with vertices $x_1, \ldots, x_n \in \mathbb{R}^d$. We define $dist_k(i)$ to be the distance from x_i to its kth nearest neighbour. An edge (i, j) exists if and only if $dist(i, j) \leq dist_k(i)$.*

Definition 2 (ϵ-Graph)**.** *An ϵ-graph is a graph $G = (V, E)$ with vertices $x_1, \ldots, x_n \in \mathbb{R}^d$. An edge (i, j) exists if and only if $dist(i, j) \leq \epsilon$.*

Definition 3 (Euclidean Minimum Spanning Tree)**.** *A Euclidean minimum spanning tree (EMST) is a graph $G = (V, E)$ with vertices $x_1, \ldots, x_n \in \mathbb{R}^d$. The edges in G form a connected tree, while $\sum_{(i,j)\in E} dist(i, j)$ is minimized.*

Though the PageRank algorithm itself is originally designed for directed web hyperlinks, we assume undirected graph in this work due to more elegant results. The ϵ-graph and EMST are undirected in nature. However, for kNN graphs, some modifications are needed, since they are directional by definition. Two possible ways to convert a (directed) kNN graph into a symmetric (undirected) graph:

- Mutual kNN. $(i, j) \in E$ if and only if $dist(i, j) \leq dist_k(i)$ and $dist(i, j) \leq dist_k(j)$.
- Symmetric kNN. $(i, j) \in E$ if and only if $dist(i, j) \leq dist_k(i)$ or $dist(i, j) \leq dist_k(j)$.

The edge weights w_{ij} are defined by a weight function $f\left(dist(i, j)\right)$. We consider here two such functions: the identity weight $f(u) \equiv 1$, and the Gaussian weight $f(u) = \exp\left(-\frac{u^2}{2\sigma^2}\right)$, with σ being the bandwidth parameter. It is worth noting that as $\sigma \to \infty$, the Gaussian weight collapses to the identity weight. This reflects the fact that an unweighted graph is a special case of a weighted one, which ensures all the results for weighted graphs apply to unweighted ones exactly.

It is widely recognized that specifying the number of neighbours in a kNN graph or the radius for an ϵ-graph is not a trivial task. These parameters give us freedom on how we construct the underlying graph. The bottom line is that we would like to choose these values large enough so that most of the vertices are connected together, generating a meaningful graph. Meanwhile, they need to be small enough so that faraway vertices will not be connected to destroy the local density information.

A similar trade-off appears in traditional kernel density estimation, in the form of bandwidth selection for kernel functions. A typical recipe relies on cross validation or similar techniques from supervised learning. These methods make full use of training data to select a "suitable" bandwidth value, which yields minimum validation error. However, cross validation is computationally demanding, and thus unsuitable for time-constrained applications. In addition, since we do not have training data at hand, other methods have to be applied instead.

Several rules of thumb for selecting k exist in kNN approaches. For example, one might try $k = \sqrt{n}$ [26], due to good results on the basis of some empirical work, where n is the number of vertices in the graph. This could serve as a starting point, in the sense that k increases much slower than n does. In [27] it is suggested that k should be in the order of $\Theta\left(\log n\right)$, so that if vertices are distributed according to a homogeneous Poisson process, the resulting graph is connected with high probability.

In the rest of this work, we will focus mainly on ϵ-graphs for theoretical discussion, while similar results can be adapted to the kNN case. We propose two criteria for the radius selection. One is motivated by random geometric graph sampled from uniform distribution in the unit hypercube, the other is determined by the growing trend of edge lengths in the EMST. Detailed discussion can be found in Section IV-B.

B. PAGERANK

The PageRank algorithm is first introduced by Page and Brin [2], [28], and employed by Google to rank web pages. PageRank is closely related to random walks. In a classic discrete time finite state random walk model, we denote by P the $n \times n$ transition matrix where n is the number of states, with P_{ij} being the transition probability from state i to state j. Then the stationary distribution $\mathbf{s}$ is defined as

$$\mathbf{s}^T = \mathbf{s}^T P \tag{5}$$

$$\text{s.t.} \qquad s_i \geq 0 \qquad \text{and} \qquad \sum_{i=1}^{n} s_i = 1. \tag{6}$$

From Eq. (5), we observe that $\mathbf{s}^T$ is the left eigenvector of P corresponding to the eigenvalue 1.

PageRank is a modified version of the random walk model,

$$\mathbf{s}^T = \alpha \mathbf{s}^T P + (1 - \alpha)\mathbf{t}^T, \tag{7}$$

where $\mathbf{t}$ is another column vector called the *teleport* vector, satisfying $\sum_{i=1}^{n} t_i = 1$, and α is a scalar called the damping factor. It is well known that a unique stationary distribution $\mathbf{s}$ exists if P corresponds to an aperiodic and irreducible Markov chain. This is not the case if the graph itself is not connected. The teleport vector $\mathbf{t}$ and the damping factor α are introduced to treat this particular flaw. Effectively, the PageRank equation (7) corresponds to a mixture of two random walks, one with transition matrix P and the other there is a transition to state i from every other state with probability t_i, and with mixing parameter α. Consequently, since it is possible to go from any state to any other state (via the teleport vector), the corresponding graph is connected and thus the chain is aperiodic and irreducible.

The original PageRank algorithm [2] treats each web page equally, setting $t_i = 1/n$, and recommends using damping parameter $\alpha = 0.85$. This choice bounds the convergence rate and empirically mimics user behaviours — on average, after following 5 hyperlinks on web pages, the random surfer jumps once — showing scalable results in real applications [29]. Subsequetnly, several variants have been proposed, e.g., topic sensitive PageRank [30], modular PageRank [31], and block PageRank [32]. These approaches belong to a more general class called personalized PageRank. They all consider a nonuniform teleport vector $\mathbf{t}$, although defining and interpreting $\mathbf{t}$ differently. The basic idea is that, when the surfer decides to jump, he will probably jump to his homepage or search engines, much more frequently than an arbitrary page, reflecting his personalized preference.

This modification motivates our proposal. In Section IV-C, we show that if $\mathbf{t}$ is chosen properly, the PageRank algorithm can produce pointwise consistent density estimates in the asymptotic sense, without worrying about the damping factor α.

IV. ALGORITHM AND PROPERTIES

A. ALGORITHM

We call our framework Anomaly Detection using Proximity Graph and PageRank (ADPP). The steps of this framework are outlined in Algorithm 1.

Algorithm 1 Outline of ADPP Algorithm

Input: the observations $\{x_i\}$, the weight function f and the teleport vector $\mathbf{t}$

Output: the PageRank vector $\mathbf{s}$

1: compute pairwise distances among measurements
2: determine vicinity criteria to form a proximity graph
3: apply the weight function f to obtain similarity matrix W
4: normalize W to get transition matrix P
5: solve $\mathbf{s}$ for $\mathbf{s}^T = \alpha \mathbf{s}^T P + (1-\alpha)\mathbf{t}^T$
6: sort $\mathbf{s}$ in ascending order and output the top few points as anomalies

The algorithm takes three input arguments,

- The observations $\{x_1, \ldots, x_n\}$. Each measurement x_i is itself a d-dimensional point.
- The weight function f. We consider the identity weight and the Gaussian weight, both of which are non-increasing functions with respect to distances between nodes.
- The teleport vector $\mathbf{t}$ which specifies the jumping probability.

The distance metric used in Line 1 does not have to be Euclidean distance. For instance, geodesic distance approximated by a kNN graph, which is the pre-step for ISOMAP [33], can be used instead to follow the underlying manifold. Other domain specific distance measures also apply. In Line 2, determining the vicinity means choosing k in a kNN graph or the radius ϵ in an ϵ-graph. In Line 6, the number of anomalies announced depends largely on engineering needs. However, it can further be controlled using more formal statistical procedures, see Section VI.

The key idea of this framework is to assign large weights to close-by points. Since the random walk transition probabilities are proportional to the edge weights, the nodes with close neighbours get higher chances to be visited. As sufficiently long time passes, the stationary distribution converges, giving us information about the density of the data points. The points with lowest chances to be visited are announced as anomalies.

B. CHOOSING PROPER RADIUS FOR ϵ-GRAPH

Next, we propose two criteria for radius selection in an ϵ-graph, one borrowed from a sharp bound for random geometric graphs, the other motivated by the growing trend of edge lengths in Euclidean minimum spanning trees (EMST).

1) SHARP BOUND FOR RANDOM GEOMETRIC GRAPHS: The first criterion for radius selection in the ϵ-graph is motivated by a well-known sharp bound in the random geometric graph literature, see e.g., [34].

A random geometric graph consists of nodes sampled uniformly from the unit hypercube of any dimension. When the distance between two nodes is shorter than some predefined radius, the end nodes will be connected by an edge, which is quite similar to our definition of ϵ-graphs. We denote by $G(n, r)$ a random geometric graph with the radius r and the number of vertices n. It has been proven that in $\mathbb{R}^2$ [35], when vertex locations are drawn from a Poisson process in the unit square, a critical radius exists for $r_c = \sqrt{\frac{\log n - \log \mu}{\pi n}}$, where μ is the Poisson rate. If $\mu \to 0$, then $G(n, r)$ is connected asymptotically almost surely (a.a.s.). If $\mu = \Theta(1)$, then $G(n, r)$ has a giant component of size $\Theta(n)$. If $\mu \to \infty$, then $G(n, r)$ is disconnected a.a.s. Goel et al. [36] later prove that every monotone graph property has a sharp threshold at $r = \Theta\left(\sqrt[d]{\frac{\log n}{n}}\right)$, where d is the number of dimensions.

Hence we choose $\epsilon = \Theta\left(\sqrt[d]{\frac{\log n}{n}}\right)$. The only caveat is that the original results are for uniformly distributed vertices in the unit hypercube, while we are dealing with arbitrary distributions in $\mathbb{R}^d$. We observe that the uniform distribution is the most "scattered" distribution, while others display more

or less clustered structures. So if using the same radius value as in the uniform case, we tend to get a graph with well connected components. Although these components might not be connected to each other, the situation can be handled in the downstream PageRank algorithm. Also, we can always preprocess our data and renormalize it into the unit hypercube, and the results shall hold.

2) GROWING TREND OF EMST LENGTHS: To motivate the second criterion, we provide some examples of dramatically different graphs and their corresponding EMST lengths in Fig. 1. The top row are the original graphs, the middle row are the corresponding EMSTs, and the bottom row are the edge lengths sorted in ascending order. We notice that, although the original graphs look quite different, the lengths of the EMST edges share the same growing trend. Most of the edges are relatively short, while there is big jumps towards the right in the plots.

The idea of utilizing EMSTs to help with outlier detection dates back to 1970s. Rohlf [37] uses the length of the longest edge M_n as a test in multivariate data, which is an extension of the "gap test" [38] in univariate case. It is worth noting that we do not use EMST lengths directly for anomaly detection purpose, but as a guideline for choosing radius.

If here we choose M_n as the radius, then our graph is surely connected, which is desirable. However, an extreme outlier would enlarge this quantity too much. The nice local property of the ϵ-graph totally gets lost. To avoid this degradation, the influence of extreme outliers ought to be reduced. We propose to find the "knee" point of this trend, and set the radius accordingly. The influence of the remaining edges in the slow growing phase are still distinguishable from each other using our weight function f.

A natural choice of picking out the "knee", is to find the point where the curvature is maximized. If we connect n successive points with coordinates $\{(a_1, b_1), \ldots, (a_n, b_n)\}$ using straight lines, then the discrete curvature [39] $\mathcal{K}_i$ at point (a_i, b_i) can be computed

$$\mathcal{K}_i = \arctan \frac{b_{i+1} - b_i}{a_{i+1} - a_i} - \arctan \frac{b_i - b_{i-1}}{a_i - a_{i-1}}. \tag{8}$$

In this criterion, we set the radius of the ϵ-graph to be

$$\epsilon = \text{length}(j) \tag{9}$$

$$\text{s.t.} \qquad j = \underset{i \in \{2,\ldots,n-1\}}{\arg\max} \ \mathcal{K}_i, \tag{10}$$

where length(j) is the length of the jth longest edge in the EMST.

C. CHOOSING PERSONALIZED TELEPORT VECTOR

It is well known that, if a finite state Markov chain is both irreducible and aperiodic, then not only does the limiting distribution exist, but also it is unique [40]. As mentioned above, the use of a teleport vector is to ensure that the corresponding chain is irreducible and aperiodic so that the PageRank distribution is well-defined.

Before introducing how we set our teleport vector, let us first make the notation explicit. For an edge between node i and j, let w_{ij} be the associated weight. If there is no edge between vertices i and j, then $w_{ij} = 0$. The weighted degree of a vertex i is defined as $d_i = \sum_{j \in V \setminus \{i\}} w_{ij}$ for $i = 1, \ldots, n$. For succinctness, let W be the matrix with w_{ij} on its ith row and jth column. We do not allow self loops in our graph, hence $w_{ii} = 0$ for $i = 1, \ldots, n$. Let D denote the diagonal matrix with $D_{i,i} = d_i$ on its principal diagonal. Then our row stochastic transition matrix is $P = D^{-1}W$. We define $\mathbf{d}$ and $\mathbf{1}$ in $\mathbb{R}^n$ to be the column vectors with elements d_i and all 1 respectively. Furthermore, the volume of the graph is defined as $Vol(G) = \sum_{i \in V} d_i$, i.e., $Vol(G) = tr(D)$.

After fixing the radius ϵ for the graph, we connect all the node pairs which have their Euclidean distance less or equal to ϵ. The weight function f is applied for each edge to get w_{ij}. At this point, W, D and $\mathbf{d}$ are all determined.

Theorem 1. *For the PageRank algorithm on undirected graphs, if the teleport vector is set to be* $\mathbf{t} = \frac{\mathbf{d}}{Vol(G)}$*, then different choices of the damping factor* α *lead to the same unique stationary distribution* $\mathbf{s} = \frac{\mathbf{d}}{Vol(G)}$.

Proof: We start from two basic observations

$$D\mathbf{1} = \mathbf{d} \qquad \text{and} \qquad W\mathbf{1} = \mathbf{d}. \tag{11}$$

For now, only invertible D is considered here. The singular case is discussed in the remarks later. Based on the definitions of $\mathbf{d}$, D and W above, we have

$$\mathbf{1} = D^{-1}\mathbf{d} \qquad \text{and} \qquad WD^{-1}\mathbf{d} = \mathbf{d}. \tag{12}$$

This means WD^{-1} has an eigenvector $\mathbf{d}$ with respect to the eigenvalue 1, which implies $1 - \alpha$ and $\mathbf{d}$ are an eigenpair of the matrix $I - \alpha WD^{-1}$, i.e.,

$$(I - \alpha WD^{-1})\mathbf{d} = (1 - \alpha)\mathbf{d} \tag{13}$$

$$(I - \alpha WD^{-1})\frac{\mathbf{d}}{Vol(G)} = (1 - \alpha)\frac{\mathbf{d}}{Vol(G)}. \tag{14}$$

On the other hand, we derive from the definition of PageRank equation in Eq. (7),

$$\mathbf{s} = \alpha P^T \mathbf{s} + (1 - \alpha)\mathbf{t} \tag{15}$$

$$(I - \alpha P^T)\mathbf{s} = (1 - \alpha)\mathbf{t} \tag{16}$$

$$\left(I - \alpha (D^{-1}W)^T\right)\mathbf{s} = (1 - \alpha)\mathbf{t} \tag{17}$$

$$\left(I - \alpha WD^{-1}\right)\mathbf{s} = (1 - \alpha)\mathbf{t}. \tag{18}$$

The last equation holds due to the fact that both W and D, for undirected graphs, are symmetric matrices. Comparing Eq. (14) with Eq. (18), if we set $\mathbf{t} = \frac{\mathbf{d}}{Vol(G)}$, then $\mathbf{s} = \mathbf{t} = \frac{\mathbf{d}}{Vol(G)}$ regardless of the choice for α. Furthermore, since there is no self loop in the graph, the diagonal elements of W are all 0, which means that the matrix $I - \alpha WD^{-1}$ is diagonally dominant, and thus invertible. Therefore the solution $\mathbf{s} = \frac{\mathbf{d}}{Vol(G)}$ is unique. ∎

We now present without proof the lemma of degrees in the ϵ-graph.

Lemma 1 (Proposition 30, [41])**.** *Suppose in an* ϵ*-graph,* n *nodes are sampled from a bounded support density* $p(x)$ *of* $\mathbb{R}^d$*, where* $0 < p_{min} \leq p(x) \leq p_{max} < \infty$*. Let* η *denote the volume of a hyperball with unit radius in* $\mathbb{R}^d$*. Then for all*

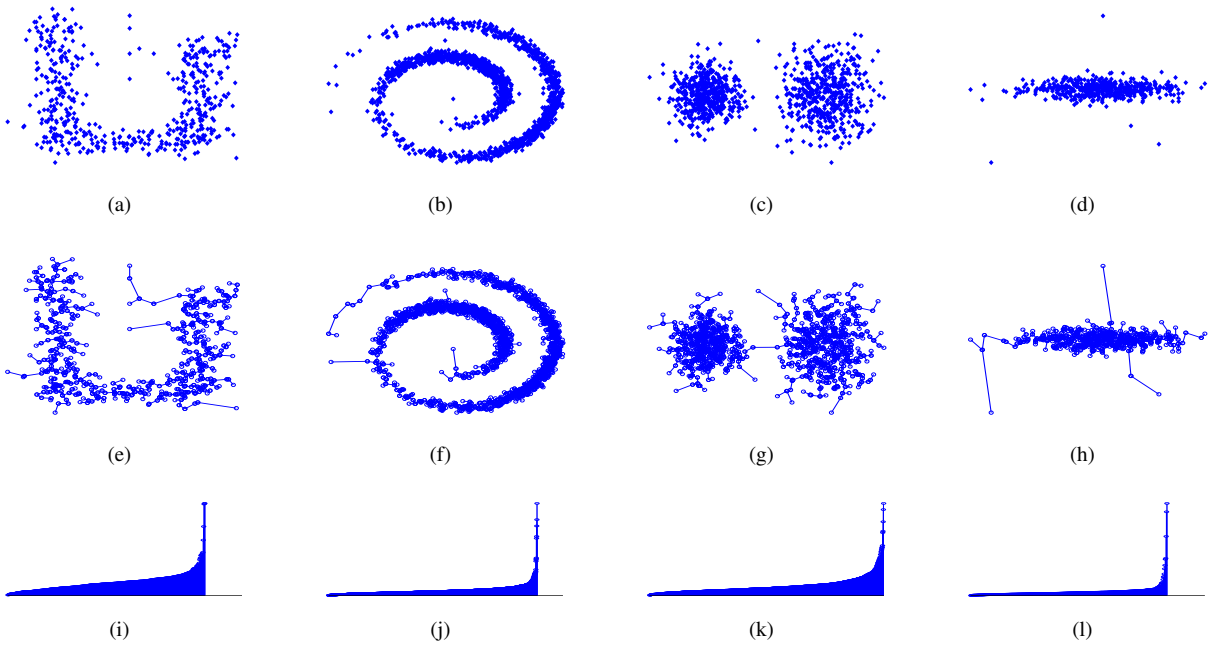


Fig. 1. Different graphs and their corresponding EMSTs. (a) "U"-shape. The points are uniformly sampled from 3 straight lines, perturbed by a Gaussian noise model. The points along the middle line are artificially injected. (b) 2-dimensional "swiss roll". The point in the heart area is artificially injected. (c) Mixture of Gaussian. The points are sampled from a mixture of two Gaussian distributions. (d) Gaussian. The points are sampled from a Gaussian distribution. The points at the top and the bottom are artificially injected. (e), (f), (g) and (h) are their corresponding EMSTs respectively. (i), (j), (k) and (l) are normalized edge length plots in ascending order, with the longest length being 1.

$\delta \in [0,1]$, *the minimal and maximal degrees* d_{min} *and* d_{max} *in the* ϵ*-graph satisfy*

$$\Pr\left(d_{max} \geq (1+\delta)n\eta\epsilon^d p_{max}\right) \leq n\exp\left(-\frac{\delta^2 n\eta\epsilon^d p_{max}}{3}\right) \tag{19}$$

$$\Pr\left(d_{min} \leq (1-\delta)n\eta\epsilon^d \beta p_{min}\right) \leq n\exp\left(-\frac{\delta^2 n\eta\epsilon^d \beta p_{min}}{3}\right), \tag{20}$$

where β *takes the support boundary effect into account.*

In general, Lemma 1 says the degrees of an ϵ-graph are concentrated within a range with high probability.

Theorem 2. *For an unweighted* ϵ*-graph, given an asymptotic radius sequence* $\{\epsilon_n\}_{n\in\mathbb{N}}$*, if* $\epsilon_n \to 0$ *as* $n \to \infty$*, the quantity* $\frac{\mathbf{d}}{Vol(G)}$ *is a pointwise consistent density estimate for a given dataset. More formally, suppose* $\{x_i \in \mathbb{R}^d, i = 1,2,\ldots,n\}$ *are i.i.d. samples from an underlying distribution* $Q(x)$ *with bounded values and finite derivatives. We construct a* ϵ*-graph on top of* x_i*. Let* d_i *be the number of neighbours for* x_i*, then the following statement is true*

$$\lim_{n\to\infty}\left|\frac{d_i}{\sum_{k=1}^n d_k} - \frac{Q(x_i)}{\sum_{k=1}^n Q(x_k)}\right| = 0, \quad \forall i = 1,2,\ldots,n. \tag{21}$$

Proof: We denote by $B(x,r)$ the hyperball centered at x with radius r, and η the volume of the hyperball with unit radius in $\mathbb{R}^d$. Then the volume of $B(x,r)$ is equal to ηr^d, i.e.,

$$\int_{B(x,r)} \mathrm{d}x = \eta r^d. \tag{22}$$

Let $\mathcal{P}_i$ be the probability measure captured inside $B(x_i,\epsilon_n)$,

$$\mathcal{P}_i = \lim_{n\to\infty}\frac{\sum_{k=1}^n \mathbb{1}_{\{x_k\in B(x_i,\epsilon_n)\}}}{n} = \lim_{n\to\infty}\frac{1+d_i}{n}, \tag{23}$$

where $\mathbb{1}$ is the indicator function. The condition $0 < m \leq \frac{1}{n}\sum_{k=1}^n d_k \leq M < \infty$ is straightforward, otherwise $\mathcal{P}_i$ cannot be estimated properly, if the number of points inside each hyperball $B(x,\epsilon_n)$ is either too small or it explodes with $n \to \infty$.

On the other hand, as $n \to \infty$, $\epsilon_n \to 0$, we have

$$\mathcal{P}_i = \int_{B(x_i,\epsilon_n)} Q(x)\mathrm{d}x \tag{24}$$

$$= \int_{B(x_i,\epsilon_n)} \left(Q(x_i) + O\left(Q'(x_i)(x-x_i)\right)\right)\mathrm{d}x \tag{25}$$

$$= Q(x_i)\int_{B(x_i,\epsilon_n)} \mathrm{d}x + \int_{B(x_i,\epsilon_n)} O\left(Q'(x_i)(x-x_i)\right)\mathrm{d}x \tag{26}$$

$$= Q(x_i)\eta\epsilon_n^d, \tag{27}$$

where Eq. (25) holds for the Taylor expansion of $Q(x)$ at x_i, and Eq. (27) holds since as $\epsilon_n \to 0$, $x \to x_i$, the second term in Eq. (26) vanishes.

Combine Eq. (27) with Eq. (23), as $n \to \infty$,

$$Q(x_i)\eta\epsilon_n^d = \frac{1+d_i}{n} \quad \Leftrightarrow \quad Q(x_i) = \frac{1+d_i}{n\eta\epsilon_n^d}. \tag{28}$$

Now we place Eq. (28) back into Eq. (21),

$$\lim_{n\to\infty}\left|\frac{d_i}{\sum_{k=1}^n d_k}-\frac{1+d_i}{n+\sum_{k=1}^n d_k}\right| \tag{29}$$

$$=\lim_{n\to\infty}\left|\frac{nd_i+d_i\sum_{k=1}^n d_k-\sum_{k=1}^n d_k-d_i\sum_{k=1}^n d_k}{(n+\sum_{k=1}^n d_k)\sum_{k=1}^n d_k}\right| \tag{30}$$

$$=\lim_{n\to\infty}\frac{1}{n}\left|\frac{d_i-\frac{1}{n}\sum_{k=1}^n d_k}{\left(1+\frac{1}{n}\sum_{k=1}^n d_k\right)\frac{1}{n}\sum_{k=1}^n d_k}\right|. \tag{31}$$

Since by Lemma 1 and $n\to\infty$, the probability of d_i falling out of some finite interval goes to 0, $\frac{1}{n}\sum_{k=1}^n d_k$ is bounded away from both 0 and ∞. Therefore, Eq. (31) $\to 0$, completing the proof. ∎

Theorem 3. *Given two asymptotic sequences $\{\sigma_n\}_{n\in\mathbb{N}}$ and $\{\epsilon_n\}_{n\in\mathbb{N}}$, if $\sigma_n\to 0$, $\epsilon_n\to 0$ and $\frac{\sigma_n}{\epsilon_n}\to\infty$, as $n\to\infty$, using the weight functions $f_n(u)=\exp\left(-\frac{u^2}{2\sigma_n^2}\right)$, the PageRank algorithm produces a point-wise consistent probability density estimate regardless of the choice for the damping factor $\alpha\in[0,1]$.*

Proof: Since we only apply the weight functions f_n to the connected edges, more precisely,

$$f_n(u)=\begin{cases}\exp\left(-\frac{u^2}{2\sigma_n^2}\right) & |u_n|\le|\epsilon_n|\\ 0 & \text{otherwise,}\end{cases} \tag{32}$$

This means that in each infinitesimal hyperball with radius ϵ_n, the edge weights become approximately constant. Therefore, the proximity graph turns out to be unweighted effectively. According to Theorem 1 and Theorem 2, Theorem 3 holds. ∎

Remarks:

1) *Why do we choose personalized PageRank over the vanilla version?* The vanilla version of the PageRank algorithm uses the uniform distribution in the transition matrix P and the teleport vector $\mathbf{t}$. However, we make adaptations in both places. We assign edge weights using f, a monotonic non-increasing function of Euclidean distances between vertices, so that even within the same radius, nearby points get more chances to be visited. For the teleport vector, we set $\mathbf{t}=\frac{\mathbf{d}}{Vol(G)}$, i.e., t_i is proportional to weighted degree for each vertex. This Matthew effect of "points in the dense area get more chances to be visited" is a positive feedback from a control system point of view, which helps the outliers to stand out.
2) *Why do we care about $\frac{\sigma_n}{\epsilon_n}$?* As mentioned earlier, in traditional kernel density estimation, the bandwidth parameter controls what is considered to be the vicinity, or how nearby nodes contribute to the position in query. In our case, both the radius and the bandwidth interact with each other, serving the same role. The net effect of selecting radius and applying the Gaussian weight function results in a truncated version of the Gaussian curve. We consider the case where $\frac{\sigma_n}{\epsilon_n}\to\infty$ as $n\to\infty$, meaning that within each infinitesimal hyperball, the contributions to the ball center from other vertices are not too different from each other. This is the key approximation to the pointwise probability density in the asymptotic sense, since intuitively if we randomly throw points onto a space, empirically the probability mass within an arbitrary shape is proportional to the number points contained in that shape, assuming the mass is a constant across the shape approximately. If otherwise $\frac{\sigma_n}{\epsilon_n}\nrightarrow\infty$ as $n\to\infty$, the constant approximation condition is violated, hence the probability density cannot be estimated accurately.
3) *What are the caveats?* Although our framework can handle both undirected and directed graphs, the statement of Theorem 1 is only valid for the undirected case. For directed graphs, the similarity matrix W is not necessarily symmetric, i.e., $W\neq W^T$, hence, we cannot obtain Eq. (18). And the final solution depends on the value of α.
The other possible catch is the existence of isolated vertices in the graph, whose weighted degrees are exactly 0, e.g., if vertex i is an isolated node, then the corresponding row of transition matrix $P_{i:}$ and the elements of teleport vector t_i are all 0. In this case, both the irreducibility and aperiodicity conditions are violated, and the degree matrix D becomes singular. However, the computation can still apply, since the corresponding elements of the stationary distribution $s_i=0$. It is equivalent to the Markov chain without those isolated states. The values of the remaining nodes in the set $V\setminus\{\text{isolated nodes}\}$ still form a unique stationary distribution of their own. When we announce the lowest PageRank values of the stationary distribution $\mathbf{s}$, the isolated nodes will be picked out automatically.

Finally, we note that the consistency results shown above (Theorem 2, in particular) apply even if the EMST edge length trend is used in place of the sharp bound for determining ϵ. Penrose [42] shows that for n i.i.d. samples from any distribution in $\mathbb{R}^d$ with connected compact support, M_n is in the order of $\Theta\left(\sqrt[d]{\frac{\theta\log n}{n}}\right)$, where θ is a distribution dependent constant. Hence our chosen radius $\epsilon\le M_n\to 0$ asymptotically, which is a prerequisite for the density convergence in Theorem 2.

V. TIME COMPLEXITY ANALYSIS

To look at the time complexity of our framework, let us discuss each step respectively.

- Pairwise distance computation. There are $\binom{n}{2}$ pairs of nodes in total, so $O\left(n^2\right)$ operations are needed.
- Radius selection for ϵ-graphs.
  1) Sharp bound criterion. Closed-form calculation takes $O(1)$.
  2) EMST trend criterion. Suppose in a graph $G=(V,E)$ with $|V|=n$. To construct an EMST for a given graph, Prim's algorithm runs in $O\left(n^2\right)$, while both Prim's and Kruskal's algorithm run in $O\left(|E|\log n\right)$ for sparse graphs. Karger et al. [43] propose a randomized algorithm running in linear time $O\left(|E|\right)$. However, in our case, we want to

find the EMST for the whole dataset, which implies the algorithms are input a dense graph, with $|E| = \Theta\left(n^2\right)$. Hence the best running time for constructing the EMST will be at least $O\left(n^2\right)$, using Prim's algorithm and array representation of edge lengths. Then sorting the lengths of the EMST takes $O\left(n \log n\right)$, and maximizing the curvature needs $O\left(n\right)$ operations. Therefore, the complexity for this criterion is $O\left(n^2\right)$.

- Weight assignment. It takes $O\left(n^2\right)$ operations to obtain the similarity matrix W, by applying the weight function f to each element in the $n \times n$ adjacency matrix.
- PageRank iteration. From Eq. (18), we can compute the stationary distribution directly with

$$\mathbf{s} = (1-\alpha)\big(I - \alpha W D^{-1}\big)^{-1}\mathbf{t}. \tag{33}$$

In general, the matrix inversion operator takes $O\left(n^3\right)$ and an $n \times n$ matrix multiplying an $n \times 1$ vector takes $O\left(n^2\right)$. This appears to be the bottleneck of the whole framework. To speed up the computation, the power method [44] is often used. The power method is an iterative procedure to find the dominant eigenvector ν of a matrix A with a unique dominant eigenvalue. Each iteration involves one matrix-vector multiplication and one norm computation ($O(n^2)$ and $O(n)$ operations respectively. The convergence rate is related to the ratio $|\lambda_2/\lambda_1|$ of the first two eigenvalues of $A = \alpha P^T + (1-\alpha)\mathbf{t}\mathbf{1}^T$. Based on the structure of A, it follows that its largest eigenvalue is $\lambda_1 = 1$, and its second largest eigenvalue satisfies $|\lambda_2| \leq \alpha$, where α is the damping parameter. Consequently, to achieve desired error tolerance ε we must run $O(\log_\alpha \varepsilon)$ iterations of the power method, and thus the computational complexity is $O(n^2 \log_\alpha \varepsilon)$. If the teleport vector is set to $\mathbf{t} = \frac{\mathbf{d}}{Vol(G)}$ as suggested, then the PageRank iteration boils down to counting the weighted degree for each vertex, which has computational complexity $\Theta(n^2)$.

- Final sort. If the lowest m PageRank values are announced as anomalies, the time complexity will be $O\left(m \log n\right)$.

As a result, the total time complexity of the ADPP framework is $O\left(n^2\right)$. It is worth noting that although not discussed in this work, if the constructed graph has linear correlations and block-wise structures, then further approximation can be used to speed up the computation, e.g., [45], [46].

VI. CONTROLLING FALSE ALARMS

The PageRank algorithm returns an ordering rank of anomalous scores on all observations. Given the PageRank vector, choosing the right number of lowest values to be anomalies depends largely on domain knowledge or engineering needs. The simplest way is to specify the number of anomalies m beforehand, and to announce the lowest m score points as anomalies, which alleviates the complicated process of parameter setting, while providing a good interaction facility between the algorithm and domain experts. Analogous approaches have been applied in top-m kNN [9] and top-m LOF (Local Outlier Factor) [47] detectors.

Note that in many applications, the ultimate goal of anomaly detection is not only identifying anomalies, but also taking actions to treat the cause. For instance, in network monitoring, anomalies often imply network congestion or system error, hence engineers are expected to fix these issues. Missed detections of course may introduce cost due to potential harm to the system or degraded service quality. However, even a small portion of false alarms can also bring huge cost, as argued in [48]. We would like to formulate the problem as a cost sensitive hypothesis testing scheme, where different costs are assigned to missed detections and false alarms.

We denote by C_{md} and C_{fa} the costs associated with missed detections and false alarms respectively. Then the ratio $\gamma = \frac{C_{md}}{C_{fa}}$ gives an explanation of how the end-user weighs the relative importance of these two errors. Given a user determined ratio γ, our task is to minimize the total cost

$$C_{total} = C_{md} \Pr\{\text{missed detection}\} + C_{fa} \Pr\{\text{false alarm}\}. \tag{34}$$

Theorem 4. *Given a PageRank vector* $\mathbf{s}$*, the cost ratio* $\gamma = \frac{C_{md}}{C_{fa}}$*, and measurements* $\{x_i \in \mathbb{R}^d, i = 1, 2, \ldots, n\}$ *sampled from distribution* $Q(x) = (1-\pi)P(x) + \pi\mu(x)$ *within* $[0,1]^d$*, where* $\mu(x)$ *is assumed to be uniform, in order to minimize total cost, the anomaly detection criterion should be set to the indicator* $\mathbb{1}_{[s_i \leq (1+\gamma)\pi]}$.

Proof: We view this problem as a hypothesis testing scheme where $H_0 : x_i \sim P$ and $H_1 : x_i \sim \mu$, and denote by $\mathcal{G}$ the rejection region. We would like to minimize the total cost C_{total},

$$\begin{aligned} C_{total} &= C_{md}\pi \Pr\{\text{nominal} \mid H_1\} \\ &\quad + C_{fa}(1-\pi) \Pr\{\text{anomalous} \mid H_0\} && (35) \\ &= C_{md}\pi \int_{\mathcal{G}^c} \mu(x)\mathrm{d}x + C_{fa}(1-\pi) \int_{\mathcal{G}} P(x)\mathrm{d}x && (36) \\ &= C_{md}\pi + \int_{\mathcal{G}} \left(C_{fa}(1-\pi)P(x) - C_{md}\pi\mu(x)\right)\mathrm{d}x. && (37) \end{aligned}$$

To minimize C_{total}, the integrand should be kept negative. Thus, we want to announce an anomaly if

$$C_{fa}(1-\pi)P(x) \leq C_{md}\pi\mu(x). \tag{38}$$

Plugging in $Q(x) = (1-\pi)P(x) + \pi\mu(x)$, we have

$$\begin{aligned} C_{fa}\left(Q(x) - \pi\mu(x)\right) &\leq C_{md}\pi\mu(x) && (39) \\ C_{fa}Q(x) &\leq (C_{fa} + C_{md})\pi\mu(x) && (40) \\ Q(x) &\leq (1+\gamma)\pi\mu(x). && (41) \end{aligned}$$

For uniform $\mu(x)$, we have $\mu(x) \equiv 1$, for all $x \in [0,1]^d$, therefore

$$Q(x) \leq (1+\gamma)\pi. \tag{42}$$

According to Theorem 3, since the PageRank vector $\mathbf{s}$ is a pointwise consistent density estimate for $Q(x)$, the optimal threshold should be $\mathbb{1}_{[s_i \leq (1+\gamma)\pi]}$. ∎

Since in this work, π is assumed unknown in general, the remaining problem is how to estimate π from data. If the null and alternative hypotheses are reversed in our setting,

this problem is identical to estimating the prior probability of the null hypothesis, which has been well studied in the statistics literature. The key idea is that the p-values form a uniform distribution when H_0 is true. For further readings, a survey on this topic is also available in [49]. In the experiments presented in the next section, rather than fixing a threshold (which requires knowing or estimating the value of π), we fix the number of anomalies which will be reported.

VII. EXPERIMENTS

A. SYNTHETIC DATA

First, let us see how the ADPP framework performs on the synthetic datasets displayed in Fig. 1. Since our problem setting is similar to Scott and Kolaczyk's work in [6], we compare their framework (MN-SCAnn) with ours. The radius of the ϵ-graph is set using the EMST growing trend criterion and the teleport vector is set to $\mathbf{t} = \frac{\mathbf{d}}{Vol(G)}$ as suggested. We use the MatlabBGL [50] package for computing the EMST of each dataset throughout our experiments. For a fair comparison, we choose the bandwidth of our Gaussian weight function to be the same as the bandwidth parameter in their kernel density estimator, which is determined by minimum integrated volume (MIV) criterion [51].

The most probable 15 anomalies from both approaches are shown in Fig. 2. The three rows show the anomalies identified by MN-SCAnn, ADPP with Gaussian weights, and ADPP with identity weights respectively. The runtime for each algorithm instance is also shown. Our objective in this first comparison is to illustrate that, under the right parameter settings, ADPP identifies a similar set of anomalous points as MN-SCAnn, but ADPP is considerably faster. To measure similarity of the two approaches, we first determine the set of 15 anomalies identified by each approach; call these $\mathcal{X}_M$ and $\mathcal{X}_A$ for MN-SCAnn and ADPP respectively. Then we compute the Jaccard index, $\frac{|\mathcal{X}_M \cap \mathcal{X}_A|}{|\mathcal{X}_M \cup \mathcal{X}_A|}$, the ratio of the size of the intersection and union of the two sets. The Jaccard coefficient is a number between 0 and 1, and the closer to 1 the more similar are the two sets. The Jaccard indices for the four particular datasets shown in Fig. 2 are 1, 0.875, 0.875 and 0.7647, respectively. We also calculate the Jaccard indices to be 0.9003 over another 200 randomly generated datasets each with 1000 data points. From these results we conclude that, with the same bandwidth parameter, ADPP can produce almost identical results to MN-SCAnn, although no kernel density estimation is required. In terms of running time, we see significant improvements for ADPP, about 100 times faster. Even if we want to save computational cost further and simply use the unweighted version instead, the detection performance is still reasonable, with another 3-8 fold speedup.

Next, we see the results of using the sharp bound criterion. We show in Fig. 3 the influence of the constant hidden behind $O\left(\sqrt[d]{\frac{\log n}{n}}\right)$. We take the "U"-shape dataset for illustration. To eliminate possible impact from edge weights, the identity weight function is used instead of Gaussian weights. We see from Fig. 3 that as the constant varies, the algorithm behaves differently. This is due to the fact that, when the constant is small, the graph concentrates on local areas, hence lots of nodes are disconnected from the main component, producing unsatisfactory results. On the other hand, a big constant introduces shortcuts into the graph, ruining the density information. For instance, the points in the middle line are well connected to the rest of the graph, which is clearly undesirable. Comparing to the EMST growing trend criterion, the sharp bound criterion does not need any complicated computation. It gives us a guideline when the number of measurements goes to infinity. For finite cases, the constant still needs to be chosen carefully. In the coming experiments, we will avoid tuning this subtle parameter and stick with the EMST growing tread criterion.

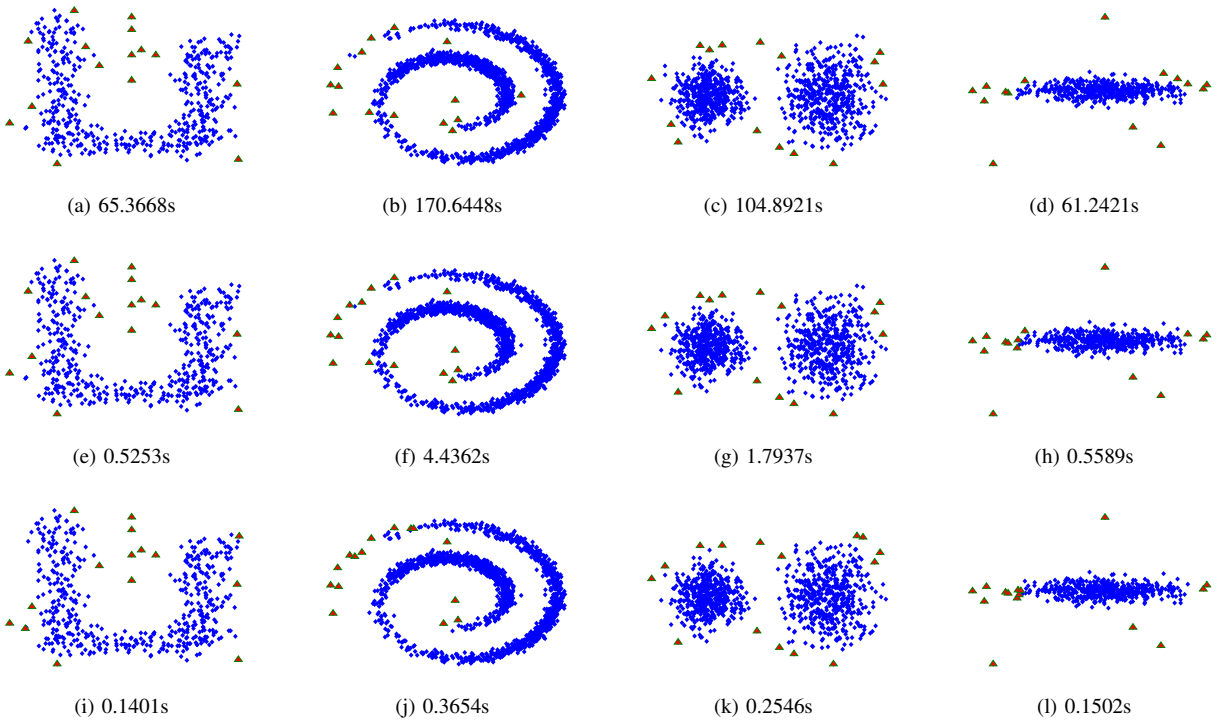
Finally, we see how ADPP scales as the number of data instances increases. We generate mixtures of Gaussian like in Fig. 1c, with different number of data points n. For each choice of n, we average over 100 runs and time successive phases of ADPP as in Table I. The running time listed in the table validates our complexity analysis in Section V. We also notice that the bottleneck of ADPP lies on the pairwise distance and EMST construction. It is worth noting that, any popular linear model, e.g., PCA [52], although fast, will not provide reliable accuracy in these nonlinear datasets, since no meaningful principal components will be identified.

B. REAL WORLD APPLICATION

1) KDDCUP 99: We first examine the KDDCUP99 dataset [53], which is a subset from the DARPA 1998 intrusion detection survey, developed by MIT Lincoln Labs. The dataset features 41 attributes to describe a connection record, as well as symbolic labels to distinguish normal traffic from attacks. All the categorical features are converted beforehand into unique numerical representations.

We compare 4 algorithms on a 4000 observations subset of the original data, which are PCA [52], Kernel PCA (KPCA) [54], Kernel recursive least square Online Anomaly Detecion (KOAD) [55] and ADPP. For PCA and KPCA, we project our data to a lower dimensional feature space so that 98% of the energy preserved. This gives us 4 dimensions for PCA and 20 dimensions for KPCA. The residual energy is used to indicate whether or not some observation deviates from the main pattern. KOAD uses former observations as a dictionary to predict new coming data. If the new data can be approximated by the dictionary reasonably well, then it is considered as nominal. Otherwise, an anomaly is declared. We use the default parameters recommended in [55] since no other automatic approach is available. For ADPP, we adopt the EMST growing trend criterion and the identity weight function for graph construction. Note that we do not compare with the MN-SCAnn algorithm since it cannot scale to handle data with more than 10 dimensions due to the curse of dimensionality [6].

Fig. 4 illustrates the performances of the four different algorithms over two different time intervals. Fig. 4a corresponds to the records from time steps 1100 to 1500, and Fig. 4b from 1500 to 1900. We notice that around time steps 1140 and 1170, some spikes appear in ADPP. Although those observations are not labelled as attacks in the original dataset, ADPP scores

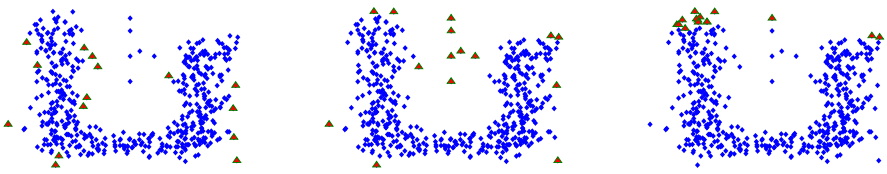


(a) 65.3668s (b) 170.6448s (c) 104.8921s (d) 61.2421s

(e) 0.5253s (f) 4.4362s (g) 1.7937s (h) 0.5589s

(i) 0.1401s (j) 0.3654s (k) 0.2546s (l) 0.1502s

Fig. 2. (a), (b), (c) and (d) are from MN-SCAnn. (e), (f), (g) and (h) are from ADPP with Gaussian weights. (i), (j), (k) and (l) are from ADPP with the identity weight function (unweighted). The red triangles (▲) denote potential anomalies. The associated value below each plot is the running time in seconds. The number of points n in these four datasets are 506, 1602, 1000 and 504 respectively.



(a) $\epsilon = 0.3\sqrt{\frac{\log n}{n}}$ (b) $\epsilon = \sqrt{\frac{\log n}{n}}$ (c) $\epsilon = 3\sqrt{\frac{\log n}{n}}$

Fig. 3. Influence of the constant for the sharp bound criterion. The red triangles (▲) denote potential anomalies.

TABLE I
RUNNING TIME IN SECONDS FOR DIFFERENT PHASES OF ADPP. ($d = 2$)

n	Pairwise distance	EMST radius	Weight assignment	PageRank	Total
200	0.0012 ± 0.0001	0.0025 ± 0.0003	$5.2257 \times 10^{-4} \pm 5.5551 \times 10^{-5}$	$1.0274 \times 10^{-4} \pm 1.2637 \times 10^{-5}$	0.0042
400	0.0054 ± 0.0005	0.0083 ± 0.0008	0.0024 ± 0.0003	$3.7261 \times 10^{-4} \pm 3.9830 \times 10^{-5}$	0.0165
800	0.0289 ± 0.0029	0.0343 ± 0.0035	0.0151 ± 0.0016	0.0012 ± 0.0001	0.0795
1600	0.1172 ± 0.0118	0.1307 ± 0.0132	0.0618 ± 0.0065	0.0041 ± 0.0004	0.3138
3200	0.4252 ± 0.0430	0.4627 ± 0.0468	0.1869 ± 0.0211	0.0155 ± 0.0016	1.0903
6400	1.7147 ± 0.1732	1.8077 ± 0.1827	0.7231 ± 0.0819	0.0589 ± 0.0060	4.3044
12800	7.0034 ± 0.7076	7.1144 ± 0.7188	2.8038 ± 0.3142	0.2308 ± 0.0235	17.1524

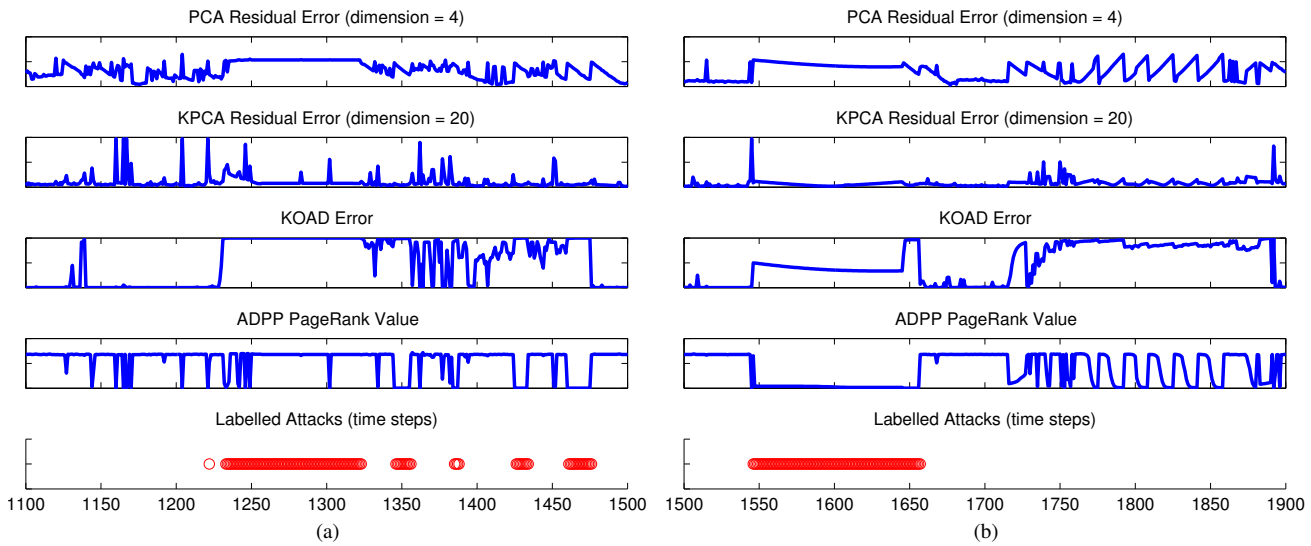


Fig. 4. Comparisons of four algorithms on KDDCUP 99 dataset at different time sections. The red circles indicate labelled attacks. The running time for PCA, KPCA, KOAD and ADPP are respectively: 0.0373s, 432.3144s, 1.7238s and 2.3755s.

low PageRank values due to the irregularity of the traffic type at those locations, which are also identified by KPCA or KOAD. The first attack at time step 1220, is only picked up by ADPP with fairly high confidence. The second set of attacks from time steps 1230 to 1320, are not identified by ADPP, since these records are clustered and last for a long time, effectively violating our uniform anomaly assumption. All the other attacks in this interval are identified by ADPP more reliably than the other methods. In Fig. 4b, we see that ADPP performs exceptionally better than the alternatives. However, we do notice weird patterns from time steps 1720 to 1900 detected by all the algorithms in comparison. The records in that range are DNS queries, which show up a total of 177 times in our 4000 records. This possibly causes some skew in the results, and ultimately, how each algorithm performs.

In terms of running time, ADPP only needs around 2.4 seconds for a 41 dimensional dataset with 4000 instances, on a par with an online algorithm KOAD, which is extremely fast. It gives a better detection rate than another nonlinear method KPCA, however, the speedup is evident, see Fig. 4.

2) USPS: USPS dataset is a well known data source for handwritten digits recognition. Each data point is a 16×16 gray scale image taken from US postal envelopes. For each picture, we stack columns on top of one another, yielding a 256-dimensional vector $x_i \in \mathbb{R}^{256}$, in which case every pixel is considered as a feature. We use part of the dataset for our experiment. Our algorithm is performed directly on the data. Neither dimension reduction nor feature selection pre-steps are taken.

First, we explore how different digits are distributed in this 256-dimensional space. For each digit, we take 1100 instances, plus 50 uniform random images. We run ADPP with the EMST growing trend criterion and the identity weight function. We check how many uniform random images appear in the top 50 positions of the PageRank vector in ascending order, and then average over 10000 such tries. We see from Table II that digit

TABLE II
THE AVERAGE NUMBER OF TIMES UNIFORM RANDOM IMAGES DETECTED IN THE TOP 50 POSITIONS FOR DIFFERENT DIGITS.

Digit	# of detection	Digit	# of detection
1	43.1899	6	40.8048
2	31.6678	7	47.5460
3	38.9360	8	1.0687
4	37.0230	9	45.4823
5	43.3232	0	41.6728

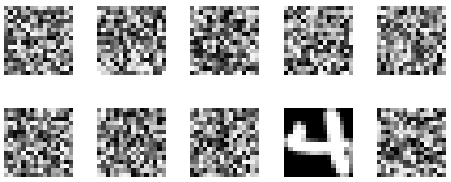
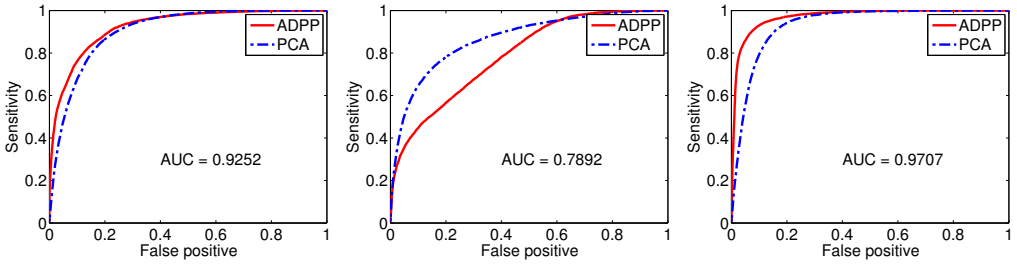


Fig. 5. Images with the lowest 10 PageRank values.

"7" and "9" have the most random images detected, while digit "8" takes an exceptionally low value. These numbers strongly indicate that "7" and "9" present good cluster structures in the 256-dimensional Euclidean embedding, while "8" is far more scattered.

Next, we give an example to visualize the results. We assume in our framework that the anomalies come from a uniform distribution, therefore we still inject random images as contaminants. Since the value for digit "4" in Table II is close to the average value for all the 10 digits, we choose "4" as the representative to form the nominal population pool. We randomly sample with replacement 1000 instances from digit "4", together with 10 uniform random images as the input of our algorithm. We examine the images with the lowest 10 PageRank values. Fig. 5 shows the results. It is worth noting that the last "noise" image ranks in the 11th position in this particular example.



(a) Digit "3" (b) Digit "8" (c) Digit "0"

Fig. 6. Selected ROC curves for different digits versus the rest 9 digits, along with the corresponding AUC values respectively.

Finally, a more realistic scenario is considered. Digits are no longer compared with random images. Each time, we fix one digit as nominal and the remaining nine digits are considered anomalous. We randomly sample with replacement 1100 images for the nominal digit, and 100 for the anomalous class. We compare ADPP with PCA, where ADPP uses the unweighted graph and the EMST rules, and PCA projects the data onto a subspace which preserves 98% of the energy. The experiment is repeated 1000 times to average out the receiver operating characteristic (ROC) curves and the area under curve (AUC) values for each digit with ADPP shown in Fig. 6.

We can see from Fig. 6 that the ROC curve for digit "8" is indeed disappointing. For instance, even if we can tolerate false alarm rate at the level of 0.1, the detection rate is still as low as 0.4. This agrees with the values listed in Table II, indicating that "8" is much more uniformly distributed, which makes it extremely difficult to distinguish from the uniform contaminants as we assumed. Digit "0" favors ADPP, while it is almost neutral for digit "3".

Since our method do not use label information, uniform anomalies are the best we can obtain. However, alternative distance metrics can be used to make the nominal points more clustered. For example, tangent distance [56], invariant to linear transformations (e.g., translation, rotation, scaling, etc.), has been shown very suitable for optical character recognition (OCR) tasks, which should result in much better performance.

VIII. SUMMARY AND FUTURE WORK

In this work, we propose a framework for anomaly detection using proximity graphs and the PageRank algorithm. This is an unsupervised, nonparametric, density estimation-free approach, readily extending to high dimensions. Various parameter selection, time complexity guarantees and possible extensions are discussed and investigated.

We see several possible directions for future development. One straightforward extension is to formalize the problem of semi-supervised anomaly detection, when partial labels are available. The label information can be adapted into our framework without difficulty by changing the teleport vector $\mathbf{t}$ accordingly in a more deliberate way.

Another direction is to make the framework online. At this stage, our algorithm operates in a batch mode. Given a set of observations, after announcing the potential anomalies once, the algorithm terminates. However, in practice, it is quite common for successive measurements to come incrementally as time passes by. Once a new observation is available, we do not want to run the whole algorithm from start again. The time complexity of our framework has already been shown to be $O(n^2)$, which is not desirable in the online fashion. We are aiming to adapt our approach to update the model in a much faster way.

Moreover, given measurements in $\mathbb{R}^d$, we use all the dimensions instead of only a subset to compute the full dimension distance. This is to say, if our algorithm produces meaningful results, all dimensions are assumed to contribute useful information for our anomaly detection task. However, in reality, especially in high dimension cases, not all of them are helpful. The inclusion of noisy dimensions may even hurt the performance. Therefore, it will be better if our framework has some feature selection ability support built in, to filter out those unwanted dimensions.

ACKNOWLEDGEMENT

We would like to thank Professor Clayton Scott for his quick response concerning the work in [6].

REFERENCES

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 15:1–15:58, 2009.

[2] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," Stanford InforLab, Tech. Rep. 1999-66, 1999.

[3] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[4] C. Scott and R. Nowak, "A Neyman-Pearson approach to statistical learning," *IEEE Transactions on Information Theory*, vol. 51, no. 11, pp. 3806–3819, 2005.

[5] ——, "Learning minimum volume sets," *Journal of Machine Learning Research*, vol. 7, pp. 665–704, 2006.

[6] C. Scott and E. Kolaczyk, "Nonparametric assessment of contamination in multivariate data using generalized quantile sets and fdr," *Journal of Computational and Graphical Statistics*, vol. 19, no. 2, pp. 439–456, 2010.

[7] A. Hero III, "Geometric entropy minimization (GEM) for anomaly detection and localization," in *Proc. Advances in Neural Information Processing Systems*, vol. 19, Vancouver, BC, Canada, 2006, pp. 585–592.

[8] S. Byers and A. Raftery, "Nearest-neighbor clutter removal for estimating features in spatial point processes," *Journal of the American Statistical Association*, vol. 93, no. 442, pp. 577–584, 1998.

[9] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," *ACM SIGMOD Record*, vol. 29, no. 2, pp. 427–438, 2000.

[10] M. Breunig, H. Kriegel, R. Ng, and J. Sander, "OPTICS-OF: Identifying local outliers," in *Proc. European Conference on Principles of Data Mining and Knowledge Discovery*, Prague, Czech Republic, 1999, pp. 262–270.

[11] ——, "LOF: Identifying density-based local outliers," *ACM SIGMOD Record*, vol. 29, no. 2, pp. 93–104, 2000.

[12] D. Pokrajac, A. Lazarevic, and L. Latecki, "Incremental local outlier detection for data streams," in *Proc. IEEE Symposium on Computational Intelligence and Data Mining*, Honolulu, HI, USA, 2007, pp. 504–515.

[13] M. Zhao and V. Saligrama, "Anomaly detection with score functions based on nearest neighbor graphs," in *Proc. Advances in Neural Information Processing Systems*, vol. 22, Vancouver, BC, Canada, 2009, pp. 2250–2258.

[14] V. Janeja and V. Atluri, "Random walks to identify anomalous free-form spatial scan windows," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 10, pp. 1378–1392, 2008.

[15] J. He, Y. Liu, and R. Lawrence, "Graph-based rare category detection," in *Proc. IEEE International Conference on Data Mining*, Houston, TX, USA, 2005, pp. 418–425.

[16] J. He, J. Carbonell, and Y. Liu, "Graph-based semi-supervised learning as a generative model," in *Proc. International Joint Conference on Artificial Intelligence*, Hyderabad, India, 2007, pp. 2429–2497.

[17] H. Cheng, P. Tan, C. Potter, and S. Klooster, "Detection and characterization of anomalies in multivariate time series," in *Proc. SIAM International Conference on Data Mining*, Sparks, NV, USA, 2009, pp. 413–424.

[18] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos, "Neighborhood formation and anomaly detection in bipartite graphs," in *Proc. IEEE International Conference on Data Mining*, Houston, TX, USA, 2005, pp. 418–425.

[19] C. Noble and D. Cook, "Graph-based anomaly detection," in *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, USA, 2003, pp. 631–636.

[20] D. Chakrabarti, "AutoPart: Parameter-free graph partitioning and outlier detection," in *Proc. European Conference on Principles and Practice of Knowledge Discovery in Databases*, Pisa, Italy, 2004, pp. 112–124.

[21] G. Nunez, Z. Kutalik, K. Cho, and O. Wolkenhauer, "Level sets and minimum volume sets of probability density functions," *International Journal of Approximate Reasoning*, vol. 34, no. 1, pp. 25–47, 2003.

[22] M. Carreira-Perpinán and R. Zemel, "Proximity graphs for clustering and manifold learning," in *Proc. Advances in Neural Information Processing Systems*, vol. 17, Vancouver, BC, Canada, 2004, pp. 225–232.

[23] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[24] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[25] X. Zhu, "Semi-supervised learning literature survey," Computer Sciences, University of Wisconsin-Madison, Tech. Rep. 1530, 2005.

[26] D. O. Loftsgaarden and C. P. Quesenberry, "A nonparametric estimate of a multivariate density function," *The Annals of Mathematical Statistics*, vol. 36, no. 3, pp. 1049–1051, 1965.

[27] P. Balister, B. Bollobás, A. Sarkar, and M. Walters, "Connectivity of random k-nearest-neighbour graphs," *Advances in Applied Probability*, vol. 37, no. 1, pp. 1–24, 2005.

[28] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1-7, pp. 107–117, 1998.

[29] A. Langville and C. Meyer, "Deeper inside PageRank," *Internet Mathematics*, vol. 1, no. 3, pp. 335–380, 2004.

[30] T. Haveliwala, "Topic-sensitive PageRank: A context-sensitive ranking algorithm for web search," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 784–796, 2003.

[31] G. Jeh and J. Widom, "Scaling personalized web search," in *Proc. International World Wide Web Conference*, Budapest, Hungary, 2003, pp. 271–279.

[32] S. Kamvar, T. Haveliwala, C. Manning, and G. Golub, "Exploiting the block structure of the web for computing PageRank," Stanford InfoLab, Tech. Rep. 2003-17, 2003.

[33] J. Tenenbaum, V. Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[34] M. Penrose, *Random geometric graphs*. Oxford University Press, 2003.

[35] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[36] A. Goel, S. Rai, and B. Krishnamachari, "Sharp thresholds for monotone properties in random geometric graphs," in *Proc. ACM Symposium on Theory of Computing*, Chicago, IL, USA, 2004, pp. 580–586.

[37] F. Rohlf, "Generalization of the gap test for the detection of multivariate outliers," *Biometrics*, vol. 31, no. 1, pp. 93–101, 1975.

[38] W. Dixon, "Analysis of extreme values," *The Annals of Mathematical Statistics*, vol. 21, no. 4, pp. 488–506, 1950.

[39] E. Grinspun. A first look at DDG: discrete curves. Columbia Univeristy. [Online]. Available: http://ddg.cs.columbia.edu/SIGGRAPH05/Didactic.pdf

[40] S. Ross, *Introduction to probability models*, 10th ed. Academic Press, 2009.

[41] U. von Luxburg, A. Radl, and M. Hein, "Hitting, commute times in large graphs are often misleading," ACM Computing Research Repository, 2011, `arXiv:1003.1266v2 [cs.DS]`. [Online]. Available: http://arxiv.org/abs/1003.1266v2

[42] M. Penrose, "A strong law for the longest edge of the minimal spanning tree," *The Annals of Probability*, vol. 27, no. 1, pp. 246–260, 1999.

[43] D. Karger, P. Klein, and R. Tarjan, "A randomized linear-time algorithm to find minimum spanning trees," *Journal of the ACM*, vol. 42, no. 2, pp. 321–328, 1995.

[44] E. Kreyszig, *Advanced engineering mathematics*, 9th ed. John Wiley & Sons, 2007.

[45] H. Tong, S. Papadimitriou, P. Yu, and C. Faloutsos, "Proximity tracking on time-evolving bipartite graphs," in *SIAM Conference on Data Mining*, Atlanta, GA, USA, 2008, pp. 704–715.

[46] H. Tong, C. Faloutsos, and J.-Y. Pan, "Random walk with restart: fast solutions and applications," *Knowledge and Information Systems*, vol. 14, no. 3, pp. 327–346, 2008.

[47] J. Tang, Z. Chen, A. Fu, and D. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," in *Proc. Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, Taipei, Taiwan, 2002, pp. 535–548.

[48] H. Ringberg, M. Roughan, and J. Rexford, "The need for simulation in evaluating anomaly detectors," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 1, pp. 55–59, 2008.

[49] M. Langaas, B. Lindqvist, and E. Ferkingstad, "Estimating the proportion of true null hypotheses, with application to dna microarray data," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 4, pp. 555–572, 2005.

[50] D. Gleich, "MatlabBGL." [Online]. Available: http://www.stanford.edu/~dgleich/programs/matlab_bgl/

[51] G. Lee and C. Scott, "The one class support vector machine solution path," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, Honolulu, HI, USA, 2007, pp. 521–524.

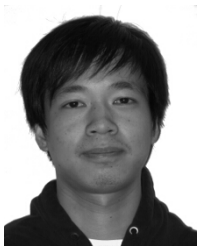
[52] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *Proc. ACM SIGCOMM*, Portland, OR, USA, 2004, pp. 219–230.

[53] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml/

[54] H. Hoffmann, "Kernel PCA for novelty detection," *Pattern Recognition*, vol. 40, pp. 863–874, 2007.

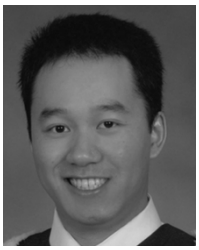
[55] T. Ahmed, M. Coates, and A. Lakhina, "Multivariate online anomaly detection using kernel recursive least squares," in *INFOCOM 2007. Proc. IEEE International Conference on Computer Communications*, Anchorage, AK, USA, 2007, pp. 625–633.

[56] P. Simard, Y. LeCun, J. Denker, and B. Victorri, "Transformation invariance in pattern recognition — tangent distance and tangent propagation," in *Neural Networks: Tricks of the Trade*, London, UK, 1998, pp. 239–274.



Zhe Yao received respectively his bachelor's degree in Telecommunication in 2005 from Nanjing University of Posts and Telecommunications, Jiangsu, China and his master's degree in Communication and Information Systems in 2009 from Beijing University of Posts and Telecommunications, Beijing, China. Since 2009, he has been a Ph.D. student at Electrical and Computer Engineering Department, McGill University, Montréal, Québec, Canada.

His research interests involve machine learning, optimization, and parallel computing.



Philip Mark received the B.E. degree in Computer Engineering from McGill University, Montréal, Québec, Canada, in 2011. He is currently with Ericsson Canada, and is planning on doing his M.A.Sc at the University of British Columnbia in 2012.

His research interests are in telecommunications and wireless systems.

Michael Rabbat (S'02–M'12) received the B.Sc. degree from the University of Illinois, Urbana-Champaign, in 2001, the M.Sc. degree from Rice University, Houston, TX, in 2003, and the Ph.D. degree from the University of Wisconsin, Madison, in 2006, all in electrical engineering.

He is currently an Assistant Professor at McGill University, Montréal, QC, Canada. He was a Visiting Researcher at Applied Signal Technology, Inc., during the summer of 2003. His research interests include distributed information processing, network monitoring, and network inference. He is currently an Associate Editor for the *ACM Transactions on Sensor Networks* and for *IEEE Signal Processing Letters*.

Dr. Rabbat received the Best Paper Award (Signal Processing and Information Theory Track) at the 2010 IEEE Conference on Distributed Computing in Sensor Systems, Outstanding Student Paper Honorable Mention at the 2006 Conference on Neural Information Processing Systems, and the Best Student Paper Award at the 2004 ACM/IEEE Conference on Information Processing in Sensor Networks.