Anomaly Detection Using Pagerank Algorithm

Deepak Shrivastava, M.Tech
CSE Dept, DIMAT, Chhattisgarh,

Somesh Kumar Dewangan, M.Tech
Associate Professor CSE Dept, DIMAT,

Abstract: — Anomaly detection techniques are widely used in a various type of applications. We explored proximity graphs for anomaly detection and the Page Rank algorithm. We used a different PageRank algorithm at peak in proximity graph collection of data points indicated by vertices, gives results a score quantifying the extent to which each data point is anomalous. In this way we requires first forming a density calculating using the training data, it was high calculative intensive for sets of high-dimensional data. In the case of mild assumptions and appropriately chosen parameters, we explored that PageRank probability in point-wise consistent density imagines for the data points in an asymptotic sense and decreased computational effort. With that heavy betterments in case of executing time are experienced while maintaining similar detection performance. This way is computationally tractable and scales perfectly to huge high-dimensional data sets.

Index Terms: Anomaly Detection, Proximity Graph, Personalized Page-Rank

I. INTRODUCTION

Distributed Denial of service (DDoS) attacks is the most undetectable attacks in the internet.

DDoS attacks can be detected in with firewall and at the network layer of TCP/IP. Network layer DDoS attacks such as ICMP flooding,

SYN flooding, and UDP flooding, which are called DDoS attacks can be detected by the firewall in general. 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 behavior. In different fields like online banking which gives credit card apps and other online shopping and also online social media networking anomaly detections are widely used. 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. There is a crisis with anomaly detection while find outing dataset patterns and data points. The existing systems work on IP log analysis and the paper is on graph based detection the normal user profile is represented as a graph with document as a node. In the testing

phase the current user profile is represented as a graph path. We can view anomaly detection as a binary classification problem, with one class being anomalous and the other normal.

II. SYSTEM STUDY

The necessity of a Web page is based on subjective, which depends on the users thinking and attitudes. And there is still a lot it was explained objectively about the relative main role of Web pages. This gives PageRank, a method for rating Web pages objectively and mechanically, perfectly calculating the user's choice. We differentiate PageRank to an idealized random Web surfer. We explore how to efficiently compute PageRank for large numbers of pages. And, we explore way of using PageRank to search and to user navigation. Incase to calculating the relative importance of web Pages, we often choose PageRank, a method for computing a ranking for each web page based on the graph. PageRank has applications in search, browsing, and traffic guess. It gives a mathematical description of PageRank and provides some intuitive justification. We explore how we efficiently compute PageRank for as many as 518 million hyperlinks. To test the utility of PageRank for search, we built a web search engine called Google. We also demonstrate how PageRank can be used as a browsing aid.

Link Structure of the Web

While estimates vary, the current graph of the crawlable Web has roughly 150 million nodes and 1.7 billion edges. Every page has some number of forward links and back links. We can never know whether we have found all the backlinks of a particular page but if we have downloaded it, we know its entire forward links at that time. Web pages vary greatly in terms of the number of backlinks they have. For example, the Netscape home page has 62,804 backlinks in our current database compared to most pages which have just a few backlinks. Generally, highly linked pages are more important than pages with few links. PageRank provides a more sophisticated method for doing citation counting. The reason that PageRank is interesting is that there are many cases where simple citation counting does not correspond to our common sense notion of importance. For example, if a web page has a link of the Yahoo home page, it may be just one link but it is a very important one. This page should be ranked higher than many pages with more links but from obscure places. Page Rank is an attempt to see how good an approximation to importance can be obtained just from the link structure.

Propagation of Ranking through Links

Based on the discussion above, we give the following intuitive description of PageRank: a page has high rank if the sum of the ranks of its backlinks is high. This covers both the case when a page has many backlinks and when a page has a few highly ranked backlinks.

Definition of PageRank

Let u be a web page. Then let Fu be the set of pages u points to and Bu be the set of pages that point to u. Let Nu = jFuj be the number of links from u and let c be a factor used for normalization. We begin by defining a simple ranking, R which is a slightly simplified version of PageRank:

R (u) = c ∑ R (v)/Nv

v€Bu

This formalizes the intuition in the previous section. Note that the rank of a page is divided among its forward links evenly to contribute to the ranks of the pages they point to. Note that c < 1 because there are a number of pages with no forward links and their weight is lost from the system. The equation is recursive but it may be computed by starting with any set of ranks and iterating the computation until it converges. The propagation of rank from one pair of pages to another. A consistent steady state solution for a set of pages. Stated another way, let A be a square matrix with the rows and column corresponding to web pages. Let Au; v = 1=Nu if there is an edge from u to v and Au; v = 0 if not. If we treat R as a vector over web pages, then we have R = cAR. So R is an eigenvector of A with Eigen value c. In fact, we want the dominant eigenvector of A. It may be computed by repeatedly applying A to any non degenerate start vector. There is a small problem with this simplified ranking function. Consider two web pages that point to each other but to no other page. And suppose there is some web page which points to one of them. Then, during iteration, this loop will accumulate rank but never distribute any rank. The loop forms a sort of trap which we call a rank sink.

III. SYSTEM DEVELOPMENT

NUMBER OF PHASES

1. Training Phase
2. Testing Phase

Training Phase

Access Log Parsing

The user accesses are stored in the access log file. The files cannot be used for direct comparison. The file is preprocessed to identify Client IP, Request, and Referrer from each user access log.

Document Matrix

The module identifies the access frequency for each document

It can be calculated as:

= (No.Of.Hits for a page per user)/ (Total Number of Logs)

The value always in between 0 to 1.

Training time Document Matrix represents the standard user access behavior.

Testing Phase

1. User Request Access

The module identifies the user requested (URI).

It also identifies the referrer URI.

The user profile is stored for further processing

2. Document Matrix

i) For every fixed interval of time, the user-profiles are processed for calculating the DM.

ii) Each individual user DM prepared.

iii) The DM rank indicates the document rank.

3. Anomaly Detection

1. User DM is cross compared with the training time DM.

2. If any URI crosses or under flows the Training Time DM for a predefined threshold.

3.The user is treated as an anomalous user.

4. The anomalous users are reported to the administrator.

4. Administration Interface

i) The system monitors the anomalous activity

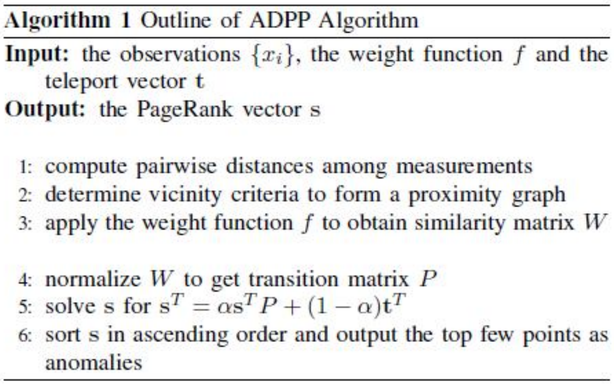
ii) The anomalous behavior of any user is reported to administrator.

iii) It allows login, view the anomalous activity.

ALGORITHM AND PROPERTIES

ALGORITHM

We call our framework Anomaly Detection using Proximity Graph and Page Rank (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 t
Output: the PageRank vector s

compute pairwise distances among measurements
determine vicinity criteria to form a proximity graph
apply the weight function $f$ to obtain similarity matrix $W$

normalize $W$ to get transition matrix $P$
solve s for $s^T = \alpha s^T P + (1 - \alpha) t^T$
sort s in ascending order and output the top few points as
    anomalies
```

The algorithm takes three input arguments,

- The observations fx1; : : : ; xng. Each measurement xi 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 t which specifies the jumping probability.

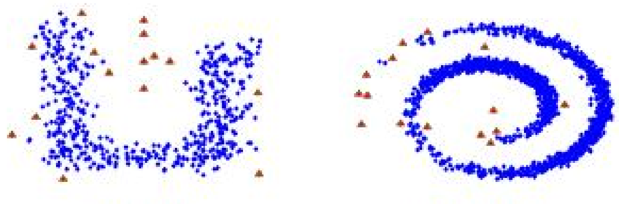
B. CHOOSING PROPER RADIUS FOR €GRAPH

We propose two criteria for radius selection in an €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 €-graph is motivated by a well-known sharp bound in the random geometric graph literature. 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 €-graphs. We denote by G (n; r) a random geometric graph with the radius r and the number of vertices n.

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. 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 are big jumps towards the right in the plots. It is worth noting that we do not use EMST lengths directly for anomaly detection purpose, but as a guideline for choosing radius.



(a) 65.3668s

(b) 170.6448s

IV. 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. Sch¨olkopf et al propose the one-class support vector machine (OCSVM) to learn the classification boundary where only nominal training data are available. Scott and Nowak 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, providing false alarm control. Later, Scott and Kolaczyk 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 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 example, a k-nearest neighbor (kNN) graph is constructed first, and then the distances from each data point to its k'th nearest neighbor are used to identify anomalies. These distances are ranked in descending order, and either a threshold is applied or the top m candidates are declared anomalous. Breunig et al., 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. extend the local outlier factor approach in an incremental online fashion. Zhao and Saligrama 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 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 is no longer required.

He 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 which uses graph-based methods for density estimation. Random walks for finding anomalies in time sequences. Investigation anomalous patterns also 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 develop methods to identify anomalous substructures in graph, purely based on the graph structure, and Chakrabarti focuses on identifying anomalous edges in graphs. In contrast, we aim to find anomalous nodes in a graph induced by high dimensional measurements. Similarly an algorithm is there Partition-Based Algorithm the fundamental shortcoming with the algorithms presented in the previous section is that they are computationally expensive. This is because for each point in the database we initiate the computation of D(P) , its distance from its Kth nearest neighbor. Since we are only interested in the top n outliers, and typically n is very small, the distance computations for most of the remaining points are of little use and can be altogether avoided. The partition-based algorithm proposed in this section prunes out points whose distances from their Kth nearest neighbors are so small that they cannot possibly make it to the top n outliers. Furthermore, by partitioning the data set, it is able to make this determination for a point without actually computing the precise value of D(p)Our experimental results indicate that this pruning strategy can result in substantial performance speedups due to savings in both computation and I/O.

V. CONCLUSION

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 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. 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

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. Sch¨olkopf, 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] 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.

[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.