Detection of Interdomain Routing Anomalies Based on Higher-Order Path Analysis

Murat Can Ganiz[1], William M. Pottenger[1], Sudhan Kanitkar[1], Mooi Choo Chuah[1]

[1] Lehigh University, Computer Science and Engineering Department,
19 Memorial Drive W., Bethlehem, PA, 18015, U.S.A.
{mug3, billp, sgk205, mcc7}@lehigh.edu

Abstract. Internet routing dynamics have been extensively studied in the past few years. However, dynamics such as interdomain Border Gateway Protocol (BGP) behavior are still poorly understood. Anomalous BGP events including misconfigurations, attacks and large-scale power failures often affect the global routing infrastructure. Since anomalous BGP events often cause major disruptions in the Internet, the ability to detect and categorize such events is extremely useful. In this article we present a novel anomaly detection technique for interdomain routing exchanges that distinguishes between different anomalies in BGP traffic. This technique is termed Higher Order Path Analysis (HOPA) and focuses on the discovery and analysis of patterns in higher order paths in supervised learning datasets. Our results indicate that not only worm events but also different types of worms as well as blackout events may be separable. This novel approach to supervised learning has potential applications in cybersecurity, cyberforensics, and text/data mining in general.

Keywords: Cybersecurity, Anomaly Detection, Graph Algorithms, Higher-Order Co-occurrence, Supervised Learning, Border Gateway Protocol, Internet Worms, Internet Routing, Cybertrust, Counterterrorism

1 Introduction

Within the last few years, internet routing dynamics have been extensively studied [1], [2], [3], [4]. However, inter-domain routing dynamics such as Border Gateway Protocol (BGP) activities are still poorly understood. Abnormal BGP events including misconfigurations [5], attacks [6], and large-scale power failures [7] often affect global routing infrastructure. For example, in January 2003, the Slammer worm caused a surge of BGP updates [8]. In August 2003, the East Coast electricity blackout affected 3175 networks and many BGP routers were shut down [9]. Since BGP anomaly events often cause major disruptions in Internet, the ability to detect and categorize BGP events is extremely useful. BGP is also very vulnerable to malicious attacks. On May 7th, 2005, an AS falsely claimed to originate Google's prefix [32] and parts of the Internet could not reach Google's search engine for roughly an hour as traffic was misdirected to the attacking AS. Thus, techniques that can detect and mitigate against such attacks will ensure continuous access to Internet.

In this article we propose a novel data mining approach termed Higher Order Path Analysis (HOPA) that distinguishes different anomalous events in BGP traffic. HOPA focuses on discovering higher-order link patterns in data based on co-occurrence relationships between entities. In this context, a higher-order link can be represented as a chain of co-occurrences of entities in different records as seen in figure 1. We

also refer to such a link as a higher-order path. Given a supervised learning dataset (i.e., labeled training data), we attempt to discover patterns in sets of higher-order links that distinguish between the classes in the labeled data.

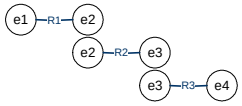

Figure 1: Higher-order path as a chain of co-occurrences

The work in [10] has some similarity with ours. Both employ co-occurrence as the relationship between entities and concentrate on the higher order co-occurrence relations or paths. The order of the relation (i.e., the length of the path) ranges from second order (e.g., as in [11]) on up. The approach in [10], however, is focused on discovering significant paths between entities such as a link between two terrorism suspects. In contrast, our approach focuses on discovering patterns in sets of the higher-order links themselves. In other words, we study the characteristics of sets of higher-order paths with the overall goal of performing classification of labeled instances based on the characteristics of these higher-order path sets. The effort discussed in [12] employs a supervised machine learning algorithm and labeled training data. Our work is similar in that we also employ labeled training data. The goal in [12] however is to learn higher-order link rules; i.e., rules that are themselves higher-order links between sets of entities. In contrast, as noted our goal is to discover the characteristics of sets of higher-order paths with the goal of leveraging these characteristics in classification. Naturally, such a technique would have wide application in supervised machine learning, including the link analysis research field.

Our results are based on a set of BGP data extracted from the RouteViews archive [13]. Our target is to characterize and distinguish different anomalous BGP events such as worm attacks (e.g., slammer, witty) and power failures.

The rest of the article is organized as follows: in Section 2 we briefly review related work. In Section 3, we present our approach followed by results in Section 4 and discussion in Section 5. Section 6 outlines some interesting research issues that we wish to explore in future work, and our conclusions are drawn in Section 7.

2 Related Work

Anomaly Detection In [15] the authors use attributes derived from BGP traffic to detect internet routing anomalies. They use data mining techniques, in particular a decision tree machine learning algorithm, to train a model using labeled data. The authors extract abnormal events from the RouteViews [13] archive and process it to reflect the counts of different types of BGP messages using one minute bins. This model consists of rules learned, and is used to detect occurrences of abnormal events. Basically their system can distinguish between two classes – worm and normal – but cannot differentiate between different types of worms. Several other efforts have been undertaken in [28], [29], [30] and [31] of a similar nature. [30] proposes two approaches, signature based and statistics-based detection. [31] employs wavelets and k-means clustering to build an instance-learning framework that identifies anomalies for a given prefix as well as across prefixes. Most of these efforts follow the same

basic steps: first the system is trained using non-event data, then the system examines test data and flags anomalies. Our approach differs in the sense that we characterize anomalous events and use models of these anomalous events to classify test data. Since our goal was to distinguish between different types of worm attacks, we focused on the data set used in [15].

Higher Order Co-Occurrence Higher order co-occurrence is closely related to our HOPA technique. In our previous work in [17], we proved mathematically that Latent Semantic Indexing (LSI), a well-known approach to information retrieval, implicitly depends on higher-order co-occurrences. We also demonstrated empirically that higher-order co-occurrences play a key role in the effectiveness of systems based on LSI. LSI can reveal hidden or latent relationships among terms, as terms semantically similar lie closer to each other in the LSI vector space. This can be demonstrated using the LSI term-term co-occurrence matrix. Let's assume a simple document collection where D1 is {human, interface} and D2 is {interface, user}. Clearly the terms "human" and "user" do not co-occur in the co-occurrence matrix of this simple two-document collection. After applying LSI, however, the reduced representation co-occurrence matrix may have a non-zero entry for "human" and "user" thus implying a similarity between the two terms. This is an example of second-order co-occurrence; in other words, there is a second-order path between "human" and "user" through "interface." In a related effort in [18], Edmonds uses higher order co-occurrence to solve a component of the problem of lexical choice, which identifies synonyms in a given context. In another effort, Zhang et al. [19] use second-order co-occurrence to improve the runtime performance of LSI. Second- and higher-order co-occurrence has also been used in a number of other applications including word sense disambiguation [20] and in a stemming algorithm [21].

Algorithms in Higher Order Path Analysis One of the challenges facing us in this work is the complexity of enumerating the various higher-order paths. In this area too, fortunately, there has been prior work on which we can build. In [22] Galil surveys techniques used for designing efficient algorithms for finding a maximum cardinality or weighted matching in (general or bipartite) graphs. For a bipartite graph G = (V, E), perfect matchings are defined as matchings such that all vertices are incident to some matching edge. On the other hand, maximum matchings are defined as matchings whose cardinalities are maximum among all matchings, and maximal matchings are matchings which are contained in no other matching. In [23], Uno propose efficient algorithms for enumerating chordless s-t paths and cycles of a given graph G = (V, E). An algorithm taking O(|E|) time for each chordless cycle is proposed. The performance of the algorithm is evaluated by computational experiments for random graphs, and showed that the computation time is constant per chordless cycle for not so dense random graphs. For the s-t paths this algorithm takes O(|V| |E|) time for each path. Additionally, in his other work, Uno [24] presents enumerating algorithms for perfect, maximum and maximal matchings in a bipartite graphs G = (V_1 U V_2, E). An algorithm that has a time complexity of O(| V_1 U V_2|) per matching is proposed for maximum matchings in bipartite graphs.

3 Approach

We focus on discovering higher-order link patterns in BGP traffic based on higher-order associations between elements of data termed entities. In this context, entities

can be aggregate counts of announce or withdraw BGP updates, and a higher-order link is represented as a chain of co-occurrences of such entities in different snapshots of BGP traffic taken over time. As noted we also refer to such a link as a higher-order path. Given a supervised learning dataset (i.e., labeled training data), we attempt to discover patterns in sets of higher-order links that distinguish between the classes in the labeled data. As such, our approach is a supervised learning technique.

Our definition of a higher-order path is similar to that found in graph theory, which states that given a non-empty graph G = (V, E) of the form V = {x_0, x_1, … , x_k }, E = { x_0x_1, x_1x_2, …, $x_{k-1}x_k$ } with nodes x_i distinct, two vertices x_i and x_k are linked by a path P where the number of edges in P is its length. Such a path is often referred to by the natural sequence of its vertices $x_0x_1…x_k$ [25]. Our definition of a higher-order path differs from this in a couple of respects. First, vertices V = {e_0, e_1, …, e_k} represent entities, and edges E = {r_0, r_1, …, r_m} represents records, documents or instances. Several edges may exist between given entities. Finally and most importantly, in a higher-order path both vertices and edges must be distinct. Figure 2 gives an example of several higher-order paths such as e_1-r_1-e_2, e_2-r_2-e_3-r_3-e_4, e_2-r_1-e_4, etc. We are interested in enumerating all such paths.

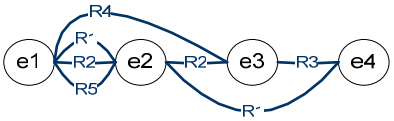

Figure 2: An example of higher-order paths with multiple edges between vertices

It is not straightforward, however, to represent higher-order paths in conventional graph structures. In order to use conventional graph structures and algorithms, we have divided the above representation into two graph structures. First, we form a co-occurrence graph G_c = (V, E) in which the vertices are the entities and there is an edge between two entities if they co-occur in one or more records. A path (length ≥ 2) extracted from G_c satisfies the first requirement of our higher-order path definition since the vertices in this path are distinct. The second requirement entails that records on a path must be distinct, and another data structure that contains lists of records for each edge is needed. We term this structure a *path group*.

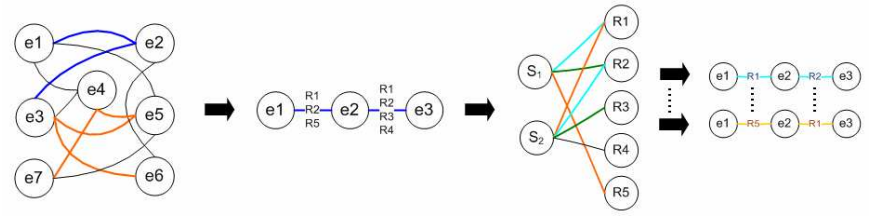

Figure 3: Process of getting a higher-order path from a co-occurrence graph via path group structure and a maximum matching of the bipartite graph representation of it.

Using the path group representation it is possible to satisfy the second requirement of our higher-order path definition. In effect, we need to identify the system of distinct representatives (SDR) for the path group. Each distinct representative in the path group corresponds to a higher order path. In order to enumerate all the distinct

representatives in a given path group, a bipartite graph $G_b = (V_1 \cup V_2, E)$ is formed such that V_1 is the sets of records (S_1, S_2,…) in a given path group and V_2 is the records themselves. A maximum matching with cardinality |V1| in this bipartite graph yields the SDR for the higher order path. This process is summarized in figure 3 where we can see an example 2nd order path group that is extracted from the co-occurrence graph G_c. This particular 2nd order path group includes two sets of records: S_1={1,2,5} and S_2={1,2,3,4}. S_1 corresponds to the records in which e_1 and e_2 co-occur, and S_2 is the set of records in which e_2 and e_3 co-occur. As noted, path group may be composed of several higher-order paths. In the third diagram in figure 3, a bipartite graph $G_b = (V_1 \cup V_2, E)$ is formed where V_1 is the two sets of records and V_2 is the all records in these sets. Enumerating all maximum matchings in this graph yields all higher-order paths in the path group. The fourth diagram in figure 3 shows one of the many paths in this path group. Edge labels R_1 and R_3 are records in S_1 and S_2, and the path corresponds to a maximum matching in the bipartite graph.

Our goal is to characterize the set of higher-order paths – in other words, we are seeking patterns in the higher-level path data itself. As a result, we need to enumerate the paths in a given dataset. This required the development of special data structures, and we based our implementation on the Text Mining Infrastructure (TMI) developed in our Parallel and Distributed Text Mining Lab [26]. The TMI is an open-source framework designed for high-end, scalable text mining, and aims to provide a robust software core for research and development of text mining applications. The TMI has an inverted index class that provides an easy and efficient way to extract co-occurrence relations between entities. Algorithms 1 through 3 below summarize our approach to enumerating higher order paths.

Algorithm 1. (ENUMERATING HIGHER-ORDER PATHS)
EnumHOPaths (dataset, order)

```
1.  G ← co-occurrence graph of dataset
2.   EnumPathGroups ( G )
3.  for i = 1 to order do
4.    read paths groups of given order from file
5.    for j = 1 to numPathGroups do
6.      form bipartite graph of pathGroup[j]
7.      enumerate and output all maximum matchings in this bipartite graph
```

Algorithm 2. (ENUMERATING ALL PATH GROUPS IN G)
EnumPathGroups (G)

```
1.  for i = 1 to numNodes do
2.    for j = i to numNodes do
3.      v_i ← i^th node of G
4.      v_j ← j^th node of G
5.      Enumerate and FormPathGroup( path ) all paths between v_i and v_j
```

Algorithm 3. (FORMING A PATH GROUP GIVEN A PATH FROM G)
FormPathGroup (path)

```
1.  for a = 2 to pathLength do
2.    form set of records in which v_{a-1} and v_a co-occur
3.    output path group to file
```

For performance reasons, we implemented our own method to discover frequent itemsets in the higher-order paths. However, our definition of frequent itemsets is a

bit different from the standard definition used in association rule mining. Itemsets in our framework are ordered, and must appear in order in a given supporting path. Additionally, the items (entities) in an itemset must be adjacent in the higher-order path. During computational enumeration of the paths, statistics are gathered. Specifically, in order to characterize a given set of records/instances, we compute frequencies of the various second- and higher-order itemsets in the set of all higher-order paths generated from the set of instances. When dealing with labeled training data used in supervised machine learning, we divide the instances by class and then characterize the resulting sets by higher-order itemset frequencies. The end result is a distribution of itemset frequencies for a given class. Actually we compute two distributions. The first is the frequencies of higher-order itemsets for particular order paths (e.g., 3-itemsets from 4^{th} order paths). These frequencies are similar to the support metric in Apriori, a well-known ARM algorithm [33]. However, instead of counting the number of records containing a given k-itemset, we count the number of higher order paths containing a given higher-order k-itemset. The second distribution is the counts of same-frequency itemsets. Either of these distributions can be compared for different classes using simple statistical measures such as the t-test. If two distributions are statistically significantly different, we conclude that the higher-order path patterns (i.e., itemset frequencies) separate the classes.

4 Results

As noted previously, the implementation of our algorithm is based on the TMI [26] and thus implemented in C++. We performed the experiments to discover the higher-order path statistics on the National Center for Supercomputing Applications (NCSA) Tungsten Supercluster (Xeon Linux). The choice to employ the TMI also opens the way to explore patterns in higher order paths in textual data sources in the future.

In our prior work [27], we analyzed a machine learning dataset from the UCI repository and concluded that the classes of instances in labeled training data may be separable using the characteristics of higher-order paths. (For more detail please refer to [27].) Based on this promising prior work, we performed experiments to discover the higher order path statistics in sets of BGP data extracted from the RouteViews archive [13]. As noted our goal was to distinguish different anomalous BGP events such as worm attacks and power failures. The anomalous events we experimented with were a slammer worm attack, a witty worm attack and a blackout (i.e., power failure). Data from a period of six hours prior to a given event and six hours following the start of the event was collected and divided into one-minute bins. Each bin became one instance in our training data and was labeled with the appropriate event class (slammer, witty or blackout). We employed the first six attributes used in [15] since they were easy to extract and appeared to represent BGP dynamics well. In preliminary experiments we used the entire 12 hour period for each event type and applied our higher order path analysis technique to discover both 3-itemset frequency and same-frequency 3-itemset counts from 4^{th} order paths for each class. We compared higher-order itemset distributions for the various classes of data, and results showed that higher order path patterns distinguished between the different event classes. Next, we compared itemset frequency distributions before and after events, again using the full 12 hours of data for each event. Our results indicated that higher-order path patterns differentiate the slammer worm period from the non-event period,

but initially we could not reliably distinguish the before and after event periods for the witty and blackout classes.

Recall that previously we were able to distinguish two different classes in a machine learning dataset (a nominal dataset). But BGP data is integer valued – attributes are basically counts of different types of BGP messages and have a large variance, which in turn affects the number of co-occurrences in higher order path analysis. As a result, the BGP co-occurrence graph was very sparse compared to the graph for the nominal dataset used in our prior work. This implied that there were many fewer co-occurrences in the BGP datasets, with the result that the number of higher order paths was also small. In order to address this issue we applied a common approach employed in machine learning, and normalized the BGP datasets by combining all before and after event datasets and dividing each attribute value by the maximum value for the attribute. This resulted in decimal numbers between zero and one. The dataset was now amenable to discretization, another useful preprocessing technique employed in machine learning. In this case, discretization was especially needful since it increased the density of the co-occurrence graph, resulting in a greater number of higher-order paths. In addition to improving classification performance, this approach enabled us to use smaller datasets to distinguish worm attacks and other events, thereby increasing both the scalability and precision of our HOPA-based detection system. This approach also enabled us to explore the use of several different time periods including two through five hour periods. After several experiments we found that HOPA distinguished between two-hour non-event periods and the respective two-hour event periods with extremely high confidence. This is depicted in the two-tailed t-test probabilities in table 1. For comparison purposes, we also show the performance of decision tree induction using the same dataset. Although accuracy is not directly comparable with confidence, clearly the HOPA technique distinguishes events with very high confidence (over 99% in all three cases). In contrast, the decision tree performance is poor for witty and only acceptable for slammer.

These results were obtained using the counts of same-frequency itemsets. Our observation is that HOPA is less sensitive when using counts of same-frequency itemsets. We speculate that due to the numeric nature of the BGP data, itemset frequency was too sensitive. Same-frequency itemset counts, on the other hand, captured the anomalous event patterns better. The decision to use either frequency of itemsets or counts of same-frequency itemsets is a parameter of the HOPA technique and may differ for particular domains or datasets. As can be seen from table 2, when we compared successive two hour periods within a given six hour event, we see no significant (≥ 95% confidence) differences. This confirms that we have successfully modeled each type of event with just two hours of data because the model is consistent across all six hours of a given event. This is important since it enables us to distinguish anomalous events from normal traffic using these models.

Table 1. Prior vs. during event period comparisons with HOPA and Decision Tree

Class 1 (prior)	Class 2 (during)	t-test results	J48 Accuracy
Slammer		1.26101E-05	96.3%
Witty		0.005349763	78.3%
Blackout		2.63894E-08	87.5%

Table 2. Event-event comparisons. f2h: first two hrs. / n2h: next two hrs. / t2h: third two hrs.

Class 1 (during)	Class 2 (during)	t-test results
Slammer-f2h	Slammer-n2h	0.543716704
Witty-f2h	Witty-n2h	0.244647853
Blackout-f2h	Blackout-n2h	0.105197985
Slammer-n2h	Slammer-t2h	0.426097536
Witty-n2h	Witty-t2h	0.566716456
Blackout-n2h	Blackout-t2h	0.191959955

The final evaluation we performed involved cross-event comparison. This is necessary in order to distinguish between event types. As portrayed in table 3, the HOPA technique successfully distinguished between worm types, something that no previous anomaly detection technique has successfully accomplished. In the case of Witty vs. Blackout, we used three-hour models.

Table 3. Event vs. Event comparison

Class 1	**Class 2**	**t-test results**
Witty	Slammer	0.00843118
Witty	Blackout	0.010703014
Blackout	Slammer	0.016590733

These results demonstrate that we are able to distinguish between different anomalous BGP events including the slammer worm attack, the witty worm attack and the 2003 USA East Coast blackout. These patterns can be used in a detection system to detect similar anomalous events.

5 Discussion

Why do patterns in higher-order paths seem to correlate with the class? In a sense it hearkens back to our prior work with Latent Semantic Indexing (LSI) [17] – in that work, as noted, we determined that the 'Latent' aspects of term similarity that LSI reveals are dependent on the higher-order paths between terms. Likewise, in real-world supervised machine learning datasets, the goal is to learn the relation between the attributes and the class. It is noteworthy that attributes are certainly not equally important. In addition, neither attributes nor instances are independent of one another, given the class. As we found with LSI, it is our contention that the 'latent semantics', if you will, of attribute-attribute relations also depend on the higher-order paths linking attribute-value pairs. By taking attribute-value pairs as our base unit of 'semantics' and linking them via higher-order co-occurrence relations, we reveal these latent semantics, or patterns, that distinguish instances of different classes. These results are extremely interesting given that we have uncovered evidence of separability based on the higher order path patterns alone. We consider this achievement significant, and something that can be exploited in many different applications using a variety of datasets as long as there is a meaningful context of entities that allows us to leverage co-occurrence relations.

Using our approach we captured patterns for several different anomalous BGP events. These patterns can be used as models in an anomalous event detection system for BGP routing. To do so, however, we need to employ a sliding window so that

events can be recognized in real time. If a given sliding window model matches a BGP data stream, we have detected the event corresponding to the model. Because we have efficient algorithms for the enumeration of higher-order paths and the HOPA algorithms are readily parallelizable, it is feasible to implement an incremental approach to higher order path analysis. This incremental HOPA algorithm will update path group structures and higher-order path counts in real time using incoming new data. We discuss this further in the following section.

6 Future Work

In future work we plan to explore the development of an incremental HOPA algorithm for use in BGP anomaly detection as well as other similar applications. In our algorithm the main data structure is the path group which consists of entities and sets of records. There are several path groups formed from the dataset. In the incremental algorithm, as we move the sliding window forward in time, new records will be added and old records deleted. This will result in changes in some of the path groups. Specifically for the BGP dataset, since it is numeric and the co-occurrence graph is less dense than a nominal dataset, it is likely that only a small fraction of these path groups will need to be updated. There are two update operations. First we check to see if any new entities are introduced by new records or existing entities are no longer referenced after deletion. If so, we need to first update the co-occurrence graph and the corresponding path groups impacted by the change. Recall from the approach that extracting path groups is quite fast. For deleted records we need only modify the path groups which include deleted records by deleting those records from the record sets and re-enumerating these path groups. Second, after modifying only those path groups necessary, we apply steps 6 and 7 of Algorithm 1 to enumerate the distinct representatives in the bipartite graph formed from each such path group. Once this step is complete, either the itemset counts or the counts of same-frequency itemsets can be updated and a statistical test performed to ascertain significance.

A second area of future work involves path formation. Specific to BGP routing data, we observed that the data differs from traditional machine learning datasets because each instance represents a particular snapshot in time. Changes in values across time may encapsulate important information for characterization of events such as different worm attacks, blackouts, broken links between routers, etc. In order to exploit this information, we propose to explore a different relation between items: time adjacency. In this case we employ a directed graph, but the graph can be based on more than one relation. Higher-order paths extracted from this graph may encapsulate changes of attributes across time and thereby capture patterns in the time dimension. This approach falls somewhere between sequence mining and traditional association rule mining (in which instances are assumed independent). An example dataset, a directed graph created using this new context definition and a higher-order path are shown in figure 4.

7 Conclusion

Several efforts employ machine learning approaches to link analysis, but few consider mining meta-level patterns in higher-order links. In our previous work [27] we focused on the discovery of such patterns in higher-order paths generated from supervised machine learning data, and developed both theoretical and algorithmic

approaches to enumerating and characterizing higher-order paths between attribute-value pairs. Based on statistical comparisons of distributions of higher-order path itemset frequencies, we discovered evidence that classes of instances in labeled training data may be separable based on the characteristics of higher-order paths.

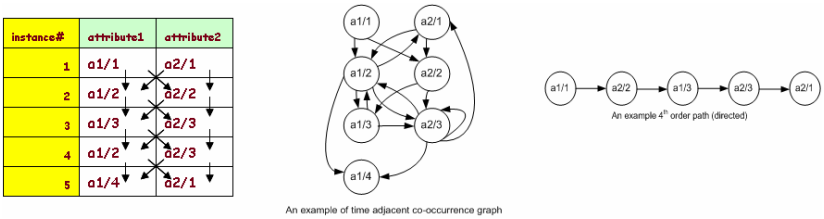

Figure 4: Incorporating the time adjacency relationship in higher order path analysis

Based on these results, in this work we analyzed higher-order path patterns in data generated during interdomain routing. We represent the data as a machine learning dataset composed of instances that correspond to one minute samples of Border Gateway Protocol (BGP) traffic. We successfully classified anomalous BGP events caused by power failures and particular worm attacks. Specific to BGP routing data, we observe that the data differs from traditional machine learning datasets because each instance represents a particular snapshot in time. This implies that a partial order may be imposed on the co-occurrence graph formed from the BGP data, potentially leading to better precision in the detection of anomalous events.

Our higher-order path analysis technique has applications in text mining as well. For instance, by considering a document or paragraph as an instance, we may determine higher order path characteristics that aid in classifying text. In fact this approach has important applications in security, counterterrorism and law enforcement.

Acknowledgments

The authors wish to thank Lehigh University, the Pennsylvania State Police, the Lockheed-Martin Corporation, the City of Bethlehem Police Department, the National Science Foundation and the National Institute of Justice, US Department of Justice. This work was supported in part by NSF grant number 0534276 and NIJ grant numbers 2005-93045-PA-IJ and 2005-93046-PA-IJ. Points of view in this document are those of the authors and do not necessarily represent the official position or policies of Lehigh University, the US Department of Justice, the National Science Foundation, the Pennsylvania State Police or the Lockheed Martin Corporation.

We are also grateful for the help of other co-workers, family members and friends. Co-author William M. Pottenger also gratefully acknowledges the continuing help of his Lord and Savior, Yeshua the Messiah (Jesus the Christ) in his life and work. Amen.

References

1. Griffin, T. What is the Sound of One Route Flapping?, IPAM 2002.
2. Caesar, M., Subramanian, L., and Katz, R.H. Route Cause Analysis of Internet Routing Dynamics. Tech Report, UCB/CSD-04-1302, 2003.

3. Lad, M., Nanavati, A., and Massey, D. An Algorithmic Approach to Identifying Link Failures. Proceedings of Pacific Rim Dependable Computing Symposium, March, 2004.
4. Mao, Z.M., Bush, R., Griffin, T.G., and Roughan, M. BGP Beacons. Proceedings of ACM IMC, 2003.
5. Mahajan, R., Wetherall, D., and Anderson, T. Understanding BGP Misconfigurations. Proceedings of ACM Sigcomm, Aug, 2002.
6. Wang, L., Zhao, X., Pei, D., Bush, R., Massey, D., and Mankin, A. Observation and Analysis Of BGP Behavior Under Stress. Proceedings of Internet Measurement Workshop, Nov, 2002.
7. Wu, Z., Purous, E. S., and Li, J. BGP Behavior Analysis During the August 2003 Blackout. In International Symposium on Integrated Network Management, 2005.
8. Lad, M., Zhao, X., Zhang, B., Massey, D., and Zhang, L. An Analysis of BGP Updates Surge During Slammer Attack. Proceedings of International Workshop on Distributed Computing (IWDC), 2003.
9. Cowie, J., Ogielski, A., Premore, B., Smith, E., and Underwood, T. Impact of 2003 blackouts on Internet Communications., Tech Report, Renesys, Nov, 2003.
10. Xu, J. J. and Chen, H. Fighting Organized Crimes: Using Shortest-Path Algorithms to Identify Associations in Criminal Networks. In Decision Support Systems 38(3), 2004, pp. 473-487.
11. Swanson, D. R. Migraine and Magnesium: Eleven Neglected Connections. Perspectives in Biology and Medicine, 31(4), 1988, pp. 526-557.
12. Mooney, R.J., Melville, P., Tang, L.R., Shavlik, J., Dutra, I.C., Page, D., and Costa, V.S. Relational Data Mining with Inductive Logic Programming for Link Discovery. Proceedings of the National Science Foundation Workshop on Next Generation Data Mining, Nov. 2002, Baltimore, MD.
13. University of Oregon Route Views Project. http://antc.uoregon.edu/route-views/.
14. Newman, J., Hettich, S., Blake, C.L., and Merz, C. J. UCI Repository of machine learning databases. http://www.ics.uci.edu/~mlearn/. MLRepository.html. University of California, Irvine, Department of Information and Computer Science, 1998.
15. Li, J., Dou, D., Wu, Z., Kim, S. An Internet Routing Forensics Framework for Discovering Rules of Abnormal BGP Events. ACM Sigcomm CCR, (35)5:55-66, Oct, 2005.
16. Feamster, N., Balakrishnan, H., Rexford, J. Some Foundational Problems in Inter-domain Routing. Proceedings of ACM Hotnets, 2004.
17. Kontostathis, A., and Pottenger, W.M. A Framework For Understanding LSI Performance. Information Processing & Management, 42(1), 2006, pp. 56-73.
18. Edmonds, P.: Choosing the word most typical in context using a lexical co-occurrence network. In Proceedings of the Thirty-fifth Annual Meeting of the Association for Computational Linguistics, 1997, pp. 507-509.
19. Zhang, X., Berry, M., and Raghavan, P. Level search schemes for information filtering and retrieval. Information Processing and Management 37 (2), 2000, pp. 313-334.
20. Schütze, H. Automatic Word Sense Discrimination. Computational Linguistics 24 (1), 1998, pp. 97-124.
21. Xu, J., Croft, W.B. Corpus-Based Stemming Using Co-Occurrence of Word Variants. ACM Transactions on Information Systems 16 (1), 1998, pp. 61-81.
22. Galil, Z. Efficient Algorithms for Finding Maximum Matching in Graphs. Computing Surveys, Vol. 18, No. 1, March 1986
23. Uno, T. An Output Linear Time Algorithm for Enumerating Chordless Cycles. 92nd SIGAL of Information Processing Society Japan, 47-53, 2003.
24. Uno, T. Algorithms for Enumerating All Perfect, Maximum and Maximal Matchings in Bipartite Graphs. Lecture Notes in Computer Science, Vol. 1350. Proceedings of the 8th International Symposium on Algorithms and Computation, 1997, pp. 92 – 101, ISBN: 3-540-63890-3, Springer-Verlag , London, UK

25. Diestel, R. Graph Theory. Springer Press, 2000, ISBN 0-387-95014-1
26. Holzman, L.E., Fisher, T.A., Galitsky, L.M., Kontostathis, A., and Pottenger, W.M. A Software Infrastructure for Research in Textual Data Mining. The International Journal on Artificial Intelligence Tools, 14 (4), 2004, pp. 829-849.
27. Ganiz, M.C., Pottenger, W.M., and Yang, X. Link Analysis of Higher-Order Paths in Supervised Learning Datasets. In the Proceedings of the Workshop on Link Analysis, Counterterrorism and Security, 2006 SIAM Conference on Data Mining. Bethesda, MD, April 2006.
28. Zhao, X., Pei, D., Wang, L., Massey, D., Mankin, A., Wu, S., and Zhang, L. Detection of Invalid Routing Announcement in the Internet. Proceedings of Dependable Systems and Networks, 2002
29. Kruegel, C., Mutz, D., Robertson, W., and Valeur, F. Topology-based detection of anomalous BGP messages. Proceedings of ACM Symposium on Recent Advances in Intrusion Detection, (28)20:17-35, Sept, 2003.
30. Zhang, K., Yen, A., Zhao, X., Massey, D., Wu, S.F., and Zhang, L. On Detection of Anomalous Routing Dynamics in BGP. Networking 2004, 3042, pp. 259 - 270
31. Zhang, J., Rexford, J., and Feigenbaum, J. Learning-Based Anomaly Detection in BGP Updates. Proceeding of the 2005 ACM SIGCOMM Workshop on Mining Network Data. 219 - 220, 2005
32. Wan, T. Analysis of BGP prefix origins during googles –May-2005 outage. Manuscript, available from Authors.
33. Agrawal, R., Imielinski, T., and Swami, A.N. Mining Association Rules Between Sets of Items in Large Databases. In P. Buneman and S. Jajodia, editors, Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pages 207–216, Washington, D.C., 26–28 1993.