Understanding and Utilizing the Hierarchy of Abnormal BGP Events*

Dejing Dou, Jun Li, Han Qin and Shiwoong Kim
Computer and Information Science
University of Oregon
Eugene, Oregon 97403
{dou,lijun,qinhan,shkim}@cs.uoregon.edu

Sheng Zhong
Computer Science and Engineering
State University of New York at Buffalo
Buffalo, NY 14260
szhong@cse.buffalo.edu

Abstract

Abnormal events, such as security attacks, misconfigurations, or electricity failures, could have severe consequences toward the normal operation of the Border Gateway Protocol (BGP) that is in charge of the delivery of packets between different autonomous domains, a key operation for the Internet to function. Unfortunately, it has been a difficult task for network security researchers and engineers to classify and detect these events. In our previous work, we have shown that with classification (which relies on the labeling with domain knowledge from BGP experts), it is feasible to effectively detect and distinguish some worms and blackouts from normal BGP behaviors. In this paper, we move one important step forward—we show that we can automatically detect and classify between different abnormal BGP events based on a hierarchy discovered by clustering. As a systematic application of data mining, we devise a clustering method based on normalized BGP data that forms a tree-like hierarchy of abnormal BGP event classes. We then obtain a set of classification rules for each class (node) in the hierarchy, thus able to label unknown BGP data to a closest class. Our method works even as the BGP dynamics evolve over time, as shown in our experiments with seven different abnormal events during a four-year period. Our work, in a more general context, shows it is promising to conduct an interdisciplinary research between network security and data mining in solving real-world problems.

Keywords: clustering, hierarchy based classification, BGP, abnormal events, worm, blackout

1 Introduction

One effective application of data mining techniques is to address a serious concern facing today's Internet: knowing the negative impacts from abnormal events that affect the Internet infrastructure, especially the Border Gateway Protocol (BGP)[12]. Various abnormal BGP events, such as large-scale power outage or fast-spreading worms, can not only affect the reachability of certain networks, but also worsen the stability, latency and reliability of reaching those networks. It is therefore compelling to devise an approach to understand these events, classify them accurately, and detect them quickly.

In our previous study, we have applied classification in our Internet Routing Forensics framework[8]. The training data are labeled based on the domain knowledge from BGP experts. We have shown that it is feasible to apply and devise classification techniques to effectively detect and distinguish some worms and blackouts from normal BGP behaviors. However, BGP is a very complex protocol and there are thousands of routers running BGP in today's Internet. Several factors could influence the effectiveness of a human based process. Such factors include the human resource cost of monitoring Internet health, the error-prone nature of human interaction and delay due to abnormal event discovery.

In this paper, we introduce clustering to advance our study on this problem. We focus first on those *global-level* events, such as worms and large-scale blackouts, which tend to affect the largest number of networks over the Internet. It is worth noting that, although few global-level events have significantly impacted BGP, studying global-level events is still extremely important. One single unknown or timely undetected global-level event could seriously damage the Internet. Understanding their hierarchy and then distinguishing and identifying them based on that hierarchy is a critical step in their detection and prevention.

To accomplish this, however, is challenging. First, the normal data and the six already-known abnormal BGP events that we study in this paper span a four-year duration (i.e., 2001-2005), a long period for the fast-evolving Internet. Recent studies on BGP dynamics[9] have shown that while the Internet size is becoming bigger, BGP routers are also becoming busier, making it difficult to compare those events without normalization. Second, the impact from different abnormal events could be different in subtle ways. For example, we have found that the impact of different worms may be different and some worm's impact is more close to a blackout event than other

*This paper is based upon work supported by the National Science Foundation under Grant No. 0520326. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

worms. Or, the impact on BGP from certain abnormal events could be very slight whereas such event still must be studied.

We introduce a systematic data mining and statistics based methodology to cluster normalized temporal BGP data to form a tree-like hierarchy of abnormal BGP events. We then derive sets of classification rules based on the hierarchy—without BGP expert's supervising. As a result, we can detect (label) if some unknown BGP data match one of the classes in the hierarchy, or detect which class is the closest match.

2 Related Works

Networking researchers have conducted several studies to discover anomalies by mining massive amounts of network data. In an early work, Lee *et al.* [7, 6] applied standard data mining algorithms on traffic flows for misuse detection. In a very recent work, Lakhina *et al.* [5] used traffic feature distributions to mine anomalies from network data. Not only did they show that the existence of anomalies can be detected from traffic flow, but also that they can use unsupervised learning to automatically classify different anomalies. Specific network data has also been mined.

Data mining approaches have been found effective for the detection of BGP anomalies. For example, Zhang *et al.* [18] proposed an instance-learning framework to identify deviations from "normal" dynamics of BGP updates, where BGP update behaviors are represented by a vector of quantified features. El-Arini *et al.* [3] employed a Bayesian framework to identify statistical anomalies in router configurations, which could be applied to discover BGP misconfigurations.

Our research is essentially using hierarchical clustering to guide the process of classification. It has some similarity with conceptual clustering, which can produce a classification schema over unlabeled objects. For example, COBWEB [4] has created a hierarchical clustering in the form of a classification tree by incrementally incorporating objects into the tree. In this tree each node refers to a concept and contains a probabilistic description of that concept. It places new objects to a node by computing the best category utility (i.e., a heuristic evaluation measure) from placing the object in each node. There are also some techniques which automatically create a hierarchy of clusters, such as bisecting k-means[14], or a hierarchy of classifiers, such as Punera *et al.*'s work[10] and Vural and Dy's work[17] for extending SVM to multi-class classification.

3 Data Preparation and Normalization

Our work in this paper begins with *data normalization*: we apply our data mining algorithms to data that are normalized based on cubic spline, rather than directly to the raw data.

3.1 Data We retrieve BGP updates from BGP update archives at either RouteViews [16] or RIPE [13] for the period when an event occurred. We then calculate the values of 12 relevant attributes about these BGP updates. The detail description of those attributes can be found in [8]. We put the calculated values into a chronological sequence of 1-minute bins into a data table, where each bin is a data row and can be regarded as a summary of the BGP activity in a 1-minute window (i.e., the smallest time interval) in terms of selected attributes.

In this paper, we will focus on six already-known abnormal events and six normal periods. The six events include three worm events and three blackout events. The three worm events are the CodeRed worm, Nimda worm, and Slammer worm; the three blackout events are the Eastcoast Blackout, Florida Blackout, and Katrina Blackout. Each of the six normal periods begins eleven days before each abnormal event, and ends one day before the abnormal event. The three worm events occurred during 2001 to 2003 and the three blackout events happened from 2003 to 2005.

3.2 Justification of normalization The Internet is changing and becoming busier. For example, the numbers of routers and address prefixes are both increasing over the past years. Our previous work [8] shows that, among other attributes, the number of announcements and the number of withdrawals of each data row (per minute) are useful attributes to distinguish between normal and abnormal BGP behavior. We gathered some statistics for the average number of announcements and withdrawals for one normal day before each event. Our study shows that the number of announcements per minute in a normal day of 2005 is about 1,000, or nearly 5 times as much as during a normal day of 2001. The number of withdrawals also shows about a five-fold increase. Without normalization, it is very possible to classify a normal time period from 2005 as a "worm" event based on one highly accurate and effective classification rule (reported in [8]) discovered from the worm data in 2001 and 2002.

3.3 Cubic spline based normalization We extended a standard and simple data fitting technique—cubic splines[15] to normalize the BGP data based on the values of 12 attributes (e.g., announcements and withdrawals). For each attribute, suppose we have n data rows in one normal period (e.g., 10 days) from the raw data. We try to normalize some data rows which are randomly selected from n data rows. For any selected data point to be normalized, we basically use the cubic spline going through two nearest time points to normalize it. Suppose $y[t]^{'}$ is the y-coordinate value at x[t] in the cubic spline going through $(x[low], y[low])$ and $(x[high], y[high])$ which are two nearest time points to $(x[t], y[t])$. The formulas for normalizing $y[t]$ to $\overline{y[t]}$ are as the following:

$$\overline{y[t]} = \frac{(y[t] - y[t]^{'})}{(\alpha * y[t] + \beta * y[t]')} + 1$$

where

$$\alpha + \beta = 1$$

In most cases, since we choose a small value for α (e.g. 0.01) and a big value for β (e.g., 0.99), $\overline{y[t]}$ is very close to the value of $\frac{y[t]}{y[t]'}$ except for the case where $y[t]^{'} = 0$, which is possible for some attributes (e.g., withdrawals).

3.4 Normalization result To evaluate the result of our normalization algorithm, we calculated the standard deviation of the normal data from six dates. For announcements, the standard deviation of the 6 normal events' raw (e.g., un-normalized) data is 265.06 while the average of these 6 events is 408.12. The standard deviation of normalized data is 0.206 while the average is 0.863. For withdrawals, the standard deviation of original data is 29.34 while the average is 36.09. The standard deviation of normalized data is 0.22 while the average is 0.79. It clearly shows that normalizing the data decreases the change in the number of announcements and withdrawals. The data points from different years can be comparable.

4 Clustering

Instead of relying on BGP experts to determine what classes of abnormal events may exist and label whether or not data from certain periods belong to a specific class of abnormal events, we now develop a clustering-based methodology to show that BGP data can be "automatically" clustered and then classified. Our methodology is based on the impact on BGP from abnormal events (or no impact from normal periods). A variety of factors contribute to the observed impact. For example, the duration of an event can be long and sustained, or it can be brief. Also, the volume of updates can be dramatic such as in the Slammer worm event, or volume changes can be subtle.

After normalizing the data points related to the CodeRed worm, Nimda worm, Slammer worm, East-coast blackout, Florida blackout, Katrina blackout and the normal periods before they happened, we can try to find the relationships among the impacts of those events and normal periods. We are mainly interested in two issues: (i) whether the impacts of worms to BGP are similar to each other and if the same is true for blackouts, (ii) whether all worms or blackouts can be distinguishable from normal periods.

4.1 Expectation-Maximization-based clustering To get the hierarchy of BGP events, we have used Expectation-Maximization (EM) [2] clustering in a hierarchical way with both top-down and bottom-up strategies. After normalizing the data we chose approximately 400 data rows for each abnormal event. We then randomly selected 400 data rows of normal events from all events occurring 10 days before an abnormality. Then we used the tool based on the Expectation-Maximization algorithm provided by WEKA [1] to do the clustering.

In the top-down approach, we put all the data from seven events (i.e., data rows from six abnormal events and data rows randomly chosen from normal periods) together in one cluster. Our goals is to sub-divide this cluster into 2 clusters by setting up the number of clusters. Then we repeatedly sub-divide each cluster until the majority of data rows from each event forms a cluster. In each step, the data rows of a particular event (differentiated by timestamp automatically) may go to different clusters. We always keep the majority of data rows for each event in one cluster but take out those in other clusters. In bottom-up approach, we put the data rows into seven clusters first based on which event they come from. Then we try to merge them to 6 clusters by setting up the number of clusters. Again, we only keep the majority data rows for each event. We further merge them into fewer clusters until all data rows can be put into a one cluster.

4.2 Clustering result For the seven events just described (including normal events), both top-down and bottom-up clustering approaches result in the same hierarchy shown in Figure 1.

We use the first day data (August 1, 2004) from the Florida blackout and the first day data (August 29, 2005) from the Katrina blackout. Both the Florida and Katrina blackout periods spanned several days. We have conducted a preliminary study to find the

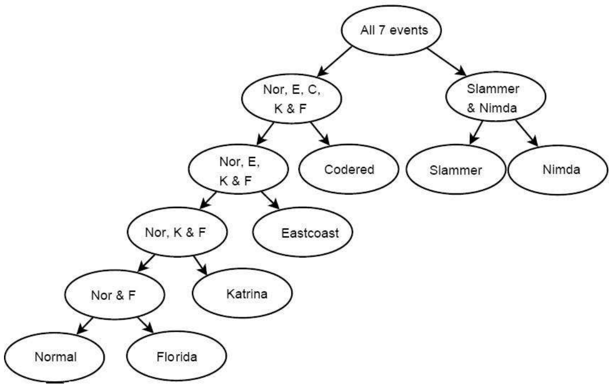

Figure 1: The hierarchy graph of six abnormal events together with normal events, where "Nor" means Normal event, "E" means Eastcoast blackout, "C" means CodeRed blackout, "K" means Katrina blackout and "F" means Florida blackout.

day which has biggest impact to BGP during the Florida and Katrina periods. We basically chose all four days of data for the Florida blackout together with normal data to do the clustering. We found that the first and the second days of the Florida blackout had the most impact to BGP, compared with normal days. Choosing the first day or the second day of the Florida blackout does not change the hierarchy we got in Figure 1. We did a similar study of all two days of the Katrina blackout and found that the first day had the most impact.

The hierarchy in Figure 1 shows that the impacts of the Slammer and Nimda worms are more similar to each other than the CodeRed worm, three blackouts and normal events. On the other hand, it is interesting that the impacts of the CodeRed worm is more similar to blackouts and normal events than the other two worms.

5 Hierarchy-Based Classification

In this section, we describe our approach to classifying different abnormal BGP events and normal periods based on the hierarchy we obtained through clustering. Our general approach is to derive rules for different classes of abnormal events through a training process and then verify (test) these rules. Every class here corresponds to a leaf or non-leaf node in the hierarchy.

5.1 Training with "Polluted" Data During the period of an abnormal event, not every single minute will display anomalies. In fact, depending on the scale and magnitude of an event's impact on BGP, some 1-minute bins may be completely normal, and some 1-minute bins may be just slightly abnormal. On the other hand, BGP data over a normal period could contain outliers. In spite of the training data my be "polluted" and complicated, the training process needs to obtain classification rules with as much accuracy as possible. Since not all training bins from an abnormal event are guaranteed to display abnormalities, when bins that are actually normal are labeled to map to an abnormal event related class (node), the training process can produce invalid rules. Manually removing these problematic bins is costly. If a large portion of training bins for an abnormal event class is normal (thus not really used), the classifier may not have a sufficient amount of useful training data to produce valid rules. In our classification study, it is effective to simply duplicate the training data for such an abnormal event class a few times (i.e., boosting).

5.2 Probability based Representation We use *C4.5* classification algorithm [11] to get classification rules. Each rule returned has an accuracy value between 0% and 100%. We take that accuracy and treat it as the probability that the rule will be correct when it matches a testing bin during the detection phase. Now, when a given rule R with accuracy R_{acc} matches a testing bin, the probability that R will incorrectly identify that bin's class label is then

$$1.0 - R_{acc}.$$

For a given class of abnormal events, say C, there may be multiple rules for C that all match a testing bin. Say $R^1, R^2, ..., R^m$ match the testing bin with rule accuracy $R^1_{acc}, R^2_{acc}, ..., R^m_{acc}$, respectively. The probability that *all* of these rules incorrectly label the testing bin as class C is then

$$P^C_{incorrect} = \prod_1^m (1.0 - R^i_{acc})$$

The probability that the testing bin matches class C is the probability that at least one of $R^1, R^2, ..., R^m$ correctly labels the testing bin as class C, which is:

$$P^C_{correct} = 1 - \prod_1^m (1.0 - R^i_{acc})$$

5.3 Alerting with "Polluted" Data We collect and test data that we have not used for the training process, which simulates new (unknown) abnormal events, and see which nodes (classes) the events should belong to. We also call this step the *detection phase.* The testing data used to verify classification rules may not be "clean" either. For testing bins

Table 1: Hierarchy based Classification Result.

Event	NoEKFC	SN	N	S	C	NoEKF	E	NoKF	K	NoF	F	No
CodeRed	**94.4**	2.4			**63.5**	26.2						
Slammer	15.9	**84.1**	21.4	**78.6**								
Nimda	17.5	**80.2**	**97.6**	2.4								
Eastcoast	**90.5**	6.3			4.8	**81.7**	**41.3**	34.1				
Florida	**99.2**	0.8			0.0	**100.0**	9.5	**68.3**	8.7	**91.3**	22.0	**74.5**
Katrina	**88.9**	7.1			0.0	**100.0**	0.0	**95.2**	**51.6**	48.4		
Normal	**99.2**	2.4			2.4	**92.9**	19.8	**70.6**	15.9	**84.1**	18.6	**78.6**
LosAngeles	**78.0**	7.9			14.2	**70.1**	7.1	**73.2**	34.6	**45.7**	35.7	**77.9**

collected from the period of an abnormal event, some testing bins are simply *not* abnormal, thus actually map to a class (node) in the path of the hierarchy which includes the normal class (node), while some map to the class of an abnormal event—but only with a certain probability.

Our alerting algorithm works as follows. First, if the probability that a testing bin matches a particular class (node) is greater than a threshold value ϵ, we call the testing bin a "hit" for that class. Then, if the percentage of "hits" for an abnormal related class (i.e., the class is related to one or more abnormal events) within a time window of W minutes exceeds a threshold Γ, an alert will be issued signifying that an event of the abnormal related class is occurring during the time window. In our results shown in the following section, we use the following values for these parameters: $\epsilon = 0.5, \Gamma = 40\%$, and $W \geq 100$.

5.4 The Result of Hierarchy based Classification We can conduct the classification based on the parent-children relationship between nodes on a clustering hierarchy. For example, in Figure 1, "Nor, C, E, K & F" and "Slammer & Nimda" are two child nodes of the root node (all events). We select the training data from normal, CodeRed worm, Eastcoast blackout, Katrina blackout and Florida blackout and label them as "Nor, C, E, K & F" respectively. We then select the training data from Slammer worm and Nimda worm and label them as "Slammer & Nimda". Using $C4.5$ algorithm with such data as the input for the training process, some example output classification rules look like:

```
Withdraw > 1.93136
Withdraw_prefix <= 4.56595
AW > 2.20632
  ->  class NorEKFC  [97.7%]

Withdraw > 5.20867
Withdraw <= 9.44596
Withdraw_prefix > 9.44532
  ->  class SlammerNimda  [93.6%]
```

where Withdraw, Withdraw_prefix and AW are attributes and all the values (numbers) in the rules are from normalized data. Similarly, we can obtain classification rules for all 12 classes (nodes) in Figure 1. Then we use all testing data from six abnormal events and normal periods to see whether the data will be labeled with the name(s) of certain node(s), or class(es). We also test a new abnormal event which has not been used for training process.

For any testing data, we first test them at the first level (e.g., "Nor, C, E, K & F" and "Slammer & Nimda") and see which class does the majority of the testing data belong to. Then we go to the next level (sub-tree) to conduct further testing. Such a hierarchy-based classification continues until we can no longer divide a class into more detailed subclasses (child nodes). As such, we are constructing a hierarchy-based classifier, using the same *C4.5* classifier at every level of the hierarchy. In our case study, this is for all six abnormal events and selected normal data, plus the data from a Los Angeles blackout. The Los Angeles blackout is the most recent abnormal event (September 12, 2005), and is used for testing only, i.e., to see which node (class) is found to be the best match.

Our results are reported in Table 1. The values in each row show the percentages of testing data bins within a time window that match a class (node) during the hierarchy based classification. We also use bold numbers to show which class (node) the majority of data matches at each level. It shows that, among the six already-known abnormal events, CodeRed, Slammer, Nimda, Eastcoast, and Katrina testing data are labeled as abnormal classes that relate to the themselves only. Florida testing data is labeled as "Normal" although it is labeled as "NoF" (a Florida related class) at a higher level. Normal testing data is labeled as "Normal." It is interesting to view the classification of recently collected Los Angeles blackout data. It is labeled as "Normal" and as "NoF" at a higher level. This shows that the impacts of the Florida and Los Angeles blackouts on BGP are similar to each other but not distinguishable from normal dates. On the other hand, the impact of all three studied worms, along with the Eastcoast and Katrina blackouts, are more severe and distinguishable.

6 Conclusions and Future Work

We discussed in this paper a systematic and automatic approach to clustering and classifying abnormal events that affect the Border Gateway Protocol (BGP), which is a critical component in today's Internet. Manually analyzing large sized temporal BGP data is daunting. We demonstrated that we can obtain classification rules based on the hierarchy discovered by clustering with normalized data from different abnormal events.

The potential for automatic, real-time applicability of our approach is likely even more important. Since our approach can be easily automated, detecting abnormal events as they occur (especially in their early stages) could be feasible. On the other hand, our approach can also continuously evolve the hierarchy and the associated rules. A newly detected abnormal event can be verified to see if it matches an already-known abnormal event class (or its related class) on the hierarchy; data from this event can also be used to update the hierarchy, and can be further used to update the classification rules. For example, after determining that the impact of a Los Angeles blackout is similar to the Florida blackout as well as close to normal days, we can easily put it in the hierarchy.

The research presented in this paper takes us one step further toward successful interdisciplinary research between network security and data mining in solving real-world problems, such as studying BGP events. In addition to the global events we studied in this paper, we will study abnormal events at a smaller scale (which tend to be harder to detect) that may only affect individual networks. The derivation of rules that can evolve over an extended period of time, the integration of other data mining techniques (such as outlier analysis and graph mining), and the applicability of our work towards BGP problem analysis and other BGP-based security research, also warrant further study.

References

[1] Weka 3: Data Mining Software in Java. `http://www.cs.waikato.ac.nz/ml/weka/`.

[2] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. Royal Statistical Society*, 39:1–38, 1977.

[3] K. El-Arini and K. Killourhy. Bayesian detection of router configuration anomalies. In *MineNet '05: Proceeding of the 2005 ACM SIGCOMM workshop on Mining network data*, pages 221–222, 2005.

[4] D. H. Fisher. Improving inference through conceptual clustering. In *AAAI*, pages 461–465, 1987.

[5] A. Lakhina, M. Crovella, and C. Diot. Mining anomalies using traffic feature distributions. In *ACM SIGCOMM '05*, pages 217–228, 2005.

[6] W. Lee, S. Stolfo, and K. Mok. A data mining framework for building intrusion detection models. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 120–132, 1999.

[7] W. Lee, S. Stolfo, and K. Mok. Mining in a data-flow environment: Experience in network intrusion detection. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, 1999.

[8] J. Li, D. Dou, Z. Wu, S. Kim, and V. Agarwal. An internet routing forensics framework for discovering rules of abnormal bgp events. *ACM Computer Communication Review*, 35(5):58–66, 2005.

[9] J. Li, M. Guidero, Z. Wu, E. Purpus, and T. Ehrenkranz. BGP routing dynamics revisited. 2007. Under review.

[10] K. Punera, S. Rajan, and J. Ghosh. Automatically learning document taxonomies for hierarchical classification. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 1010–1011, 2005.

[11] J. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, 1993.

[12] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271 (Draft Standard), January 2006.

[13] RIPE NCC. RIPE routing information service raw data. `http://data.ris.ripe.net/`.

[14] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *Proceedings of KDD Workshop on Text Mining 2000*, 2000.

[15] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis.* Springer-Verlag New York, Inc., 2002.

[16] University of Oregon Route Views Project. Route-Views. `http://antc.uoregon.edu/route-views/`.

[17] V. Vural and J. G. Dy. A hierarchical method for multi-class support vector machines. In *Proceedings of the twenty-first international conference on Machine learning*, page 105, 2004.

[18] J. Zhang, J. Rexford, and J. Feigenbaum. Learning-based anomaly detection in BGP updates. In *MineNet '05: Proceeding of the 2005 ACM SIGCOMM workshop on Mining network data*, pages 219–220, 2005.