Link-based Anomaly Detection in Communication Networks

Xiaomeng Wan, Evangelos Milios, Nauzer Kalyaniwalla and Jeannette Janssen
Faculty of Computer Science
Dalhousie University
xwan,eem,nauzerk@cs.dal.ca,janssen@mathstat.dal.ca

Abstract

Communication networks, such as networks formed by phone calls and email communications, can be modeled as dynamic graphs with vertices representing agents and edges representing communications. Anomaly detection is to identify abnormal behaviour occurring in these networks. This is crucial for anti-terrorism, resource allocation and network management. The contents of the communications are often unavailable or protected by regulations or encryption, which makes linkage information the only type of data we can rely on in order to identify anomalies. In this paper, we propose a link-based anomaly detection method that considers deviations from individual patterns by taking into account the behaviour pattern of the cluster to which the individual belongs. Clusters can be formed by a standard clustering procedure or based on a specific attribute depending on the dataset. Experiments show that this method performs well on both network traffic and email communication data.

1 Introduction

Communications between a set of agents form a communication network; such communications can be phone calls, emails, financial transactions and network traffic. Anomalies in these networks usually mean frauds, congestion or even terrorism. Most of these network data have the form as continuous data streams of vast volume, which makes it very difficult to analyze their contents. Moreover, the contents of the communications are often either unavailable (contents of a phone conversation), or are protected by privacy regulations or encryption (contents of e-mail messages). Under such circumstances, linkage information becomes the only resource available. Link-based anomaly detection identifies anomalies using linkage information only. Thus, it is fast and uses less storage. A graph is built to represent a communication network with vertices representing agents and edges, communications. Since communications in these networks take place over time, the edges of the graphs appear and disappear in time, resulting in a dynamic graph, the form of which changes over time. The graph formed is a manifestation of the underlying social interactions. Consequently, mining the dynamic graph may potentially identify abnormal behaviour (e.g. mining phone call graphs for telephone fraud detection [3]).

Most link-based anomaly detection methods focus on the variations of one feature, most commonly the communication volume. We believe that it is the interaction of multiple features that defines anomalies. For example, communication bursts with a large number of recipients may be normal, but if the number of recipients is small, it becomes a strong indication of something happening between a small group of vertices. In our method, we use multiple features to capture different aspects of the network. Most link-based methods consider anomalies as behaviours deviating from previous behaviour patterns [5]. Their results are plagued by false alarms because of new normal behaviour patterns developing. To reduce false alarms, we introduce a clustering process to cluster vertices with similar behaviour patterns together and build profiles for the clusters. New patterns are checked against cluster profiles when deciding whether they represent anomalies.

Our method is evaluated on both network traffic and email communication data. Our experimental results show that it effectively reduces false alarms on both datasets.

2 Our method

Most anomaly detection methods identify anomalies by building a profile for each vertex based on its historic behaviour, and detecting behaviour that deviates from the profiles. However, behaviour not seen before does not necessarily mean that an anomaly occurred. Without further discrimination, this approach leads to false alarms. To distinguish new normal behaviour patterns from anomalies, we need take into account the behaviours of other vertices of the same kind. In our method, we introduce a clustering process to cluster vertices with similar communication be-

haviours together and build profiles for the clusters. Once a new communication pattern is discovered, it will be checked against the cluster profile to see whether it is also abnormal with respect to other vertices in the same cluster. If not, it will be regarded as a false alarm.

We call the deviations from individual profiles individual deviations and deviations from cluster profiles, cluster deviations. Let S be entire feature space, C be the space of cluster profile and A be space of individual profile of a single vertex in this cluster. Under normal circumstances, a vertex's behaviour should be close to the typical behaviour of its cluster (cluster center) and at the same time, some variation around its own typical behaviour (individual center) is allowed. Approaches based on individual deviation identify anomalies in the area outside of the individual profile, which is $\{S - A\}$. The problem is that some individual deviations may move towards the cluster center, while others move away from it. In the first case, the behaviours will still remain in normal scope; while in the second case, they will result in even larger cluster deviations. Individual deviation approaches treat the first cases (area $\{C - A\}$) as anomalies and, therefore, introduce false alarms. With the cluster profile built for each cluster, the two cases can be easily distinguished by their cluster deviations.

The advantages of our approach include: first, it uses link-based features only; Second, it uses multiple features that capture different aspects of the network and as a result, can detect various anomalies.

2.1 Preprocessing

First, the time span of the dynamic graph G is split into disjoint time intervals (e.g. one hour, one day). Within each time interval t, a static graph $g(t)$ is built to summarize the dynamic graph. In other words, all the edges that ever appeared during this time interval are part of the static graph. For each vertex v_i in $g(t)$, link-based features are extracted and a feature vector $V_i(t)$ is generated. Individual and cluster profiles are built based on these feature vectors in the training period.

2.2 Individual deviation

The profile of each vertex v_i is built based on its feature vectors in the training period (T, T'). It consists of individual center μ_i defined as the mean vector of its time series of feature vectors and covariance matrix Σ_i that captures the correlations of the features. After the profile is built, individual deviation $di(v_i, t)$ of v_i at time interval t is computed as the Mahalanobis distance from its current feature vector $V_i(t)$ to its individual center μ_i.

2.3 Cluster deviation

The cluster profile is built based on the vertices' median feature values in the training period. We use medians because they are resistant to outliers. The vector of median feature values for a vertex v_i in the training period (T, T') is denoted as M_i. The profile for cluster c_j with n vertices consists of cluster center μ_{c_j} defined as the mean vector of its members' median vectors, and covariance matrix Σ_{c_j} as well. Cluster deviation $dc(v_i, t)$ of v_i at time interval t is defined as the Mahalanobis distance from its current feature vector $V_i(t)$ to its cluster center μ_{c_j}.

After these two deviations are computed, "anomalies" are identified as vertices whose behaviours have high scores on both individual and cluster deviations (i.e. exceeding thresholds α and β respectively).

$$\{v_i | di(v_i, t) > \alpha \text{ and } dc(v_i, t) > \beta\}, \quad (1)$$

3 Experiments and results

3.1 DARPA98 dataset

We first applied our method to the 1998 DARPA Intrusion Detection Evaluation Data [1]. Its training data consists of seven weeks of network-based attacks in the midst of normal background data. The dataset consists of files that record information for each connection session, including the time, duration, source IP and port, destination IP and port as well as the type of each attack. In [4], several detection schemes based on dimensional distances are evaluated on this data. They use three groups of features: content-based, time-based and connection-based. Connections with large distance from the rest in the feature space are identified as attacks.

Because of the nature of our link-based approach, our method can only detect bursty attacks that involve multiple connections. Single-connection attacks have no difference with other normal connections from the linkage point of view. First, we use the list files to build the dynamic graph with vertices representing IP+port combination (e.g. 172.016.112.050:23) and edges representing connection sessions. The time interval we use to build static graphs is one hour. In this project, we focus on a vertex's inlinks to identify who is being attacked. We use three features to characterize each vertex's inlinks: the total number of connection requests received, the average number of requests received from each IP and the average number of requests received from each vertex.

A special characteristic of this data is that, because network services are usually provided through specific ports (e.g. port 80 for HTTP), there is a natural clustering of the vertices based on the ports that they are associated with.

Vertices associated with the same port but different IPs usually provide the same kind of service. So we simply put them into one cluster (e.g. cluster for port 80). Since a number of samples are needed to estimate the covariance matrix, we only consider those clusters with at least 10 vertices. Using day one as the training period, we get 736 vertices in 6 clusters associated with different ports. The cluster and individual profiles are built for them with attacking connections eliminated. In real operational settings where attacks are unlabeled, robust estimation should be applied to build the profiles. Table 1 shows the partial list of a total of 66 bursty attacks targeting these 736 vertices in 35 days of training data. Our method is applied to detect these attacks.

Table 1. List of attacks

day	start time	target	attack
3	11:55:15	172.016.112.050:23	neptune
11	09:37:06	172.016.112.050:79	satan
17	23:15:08	172.016.114.050:23	teardrop
21	10:12:00	172.016.114.050:79	satan
			...

Fig. 1 and 2 show the series of maximal individual deviations and cluster deviations at each time interval on day 3. The only real attack in day 3 targets a vertex with port 23 at timestep 4. As we can see, if we set up a threshold for the individual deviations (e.g. $di > 20$), we can detect the real attack but at the same time, we will get many false alarms. But when we take the cluster deviations into considerations and use a different threshold for them (i.e. $di > 20$ and $dc > 30$), the only anomaly that satisfies this criterion is the real attack (note that maxima are achieved by different vertices at different time intervals). This result demonstrates the validity of our previous analysis.

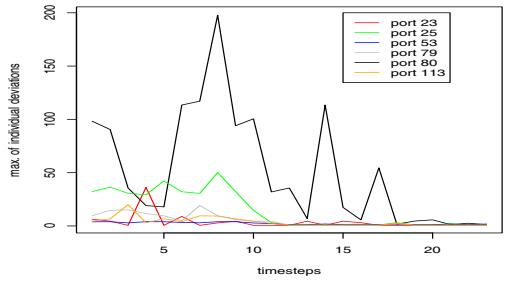

Figure 1. Maximal individual deviations for day 3

Precision is defined as the percentage of identified anomalies that are real attacks. Fig. 3 shows the precision of detection on all 66 attacks for different threshold combinations (i.e. different α and β). When the threshold for cluster deviation (β) is zero, it means that we are using individual deviations only without considering the cluster deviation. While all these combinations achieve higher than 85% recall, the precision tells us that our method performs much better than the individual deviation approach at eliminating false alarms.

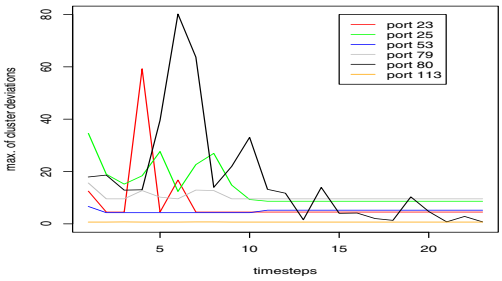

Figure 2. Maximal cluster deviations for day 3

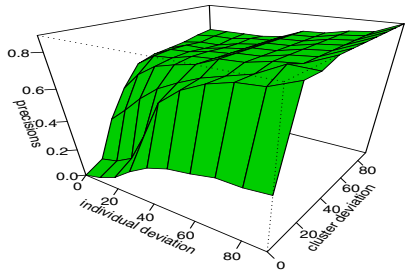

Figure 3. Precision from different α and β

3.2 CS email data

The CS email dataset is derived from the log files on the email server of the Faculty of Computer Science. Overall, there are 26,425 email addresses involved. We only examine active CS accounts (defined as accounts that have sent out at least one message in at least 10 weeks in the first 50 weeks). This results in 649 unique e-mail accounts. The size of the time interval we use for this data is one week and the training period is the first 50 weeks. The feature set is shown in Table 2. A "new" email means that the sender never sent email to the recipient before.

Vertices are first clustered using model-based clustering [2]. It assumes that the data are generated by a mixture of probability distributions in which each distribution forms a cluster. Since anomalies are not labeled for this data, we

Table 2. Overview of features

Features	Description
IN	number of people received emails from
OUT	number of people send emails to
NO	number of new recipients
IR	number of received and replied emails
OR	number of sending out emails get replied
AR	average emails to each recipient
RG	times of current recipients appear as recipients in previous 10 weeks
CC	number of emails between recipients
ICC	number of emails between people received emails from
NN	number of new emails between recipients

cannot ensure clean data for training purpose and have no ground truth for evaluation. We made two changes to our method to adapt to this situation. First, we use robust estimation to compute individual profiles. Robust estimation can estimate the center and covariance matrix for data with outliers [6]. Second, we develop a novel evaluation metric, *Back to Normal (BTN) ratio*, based on the assumption that anomalies do not last longer than one time interval (i.e. one week here). It is defined as the percentage of detected anomalies in time interval t that return back to normal in the next time interval $t+1$.

$$BTN(t) = \frac{|\{v_i|V_i(t) \text{ is anomaly and } V_i(t+1) \text{ is normal}\}|}{|\{v_i|V_i(t) \text{ is anomaly}\}|} \quad (2)$$

As long as false alarms at t still get caught as anomalies in next time interval $t+1$, BTN ratio is equal to $precision \times \lambda$, where λ is the percentage of real anomalies lasting less than one time interval. When the assumption is true (i.e. λ is 1), the BTN ratio is equal to the precision. If the assumption is only partially true, the BTN ratios are still comparable assuming that, for each dataset, λ is fixed. Since we have no ground truth in this CS data and cannot compute the precisions, we use the BTN ratios to compare the detection results. We apply our method to detect anomalies in the CS email data from week 51 to 55. Fig. 4 shows the BTN ratios for different α and β combinations. Once again, our method achieves much better BTN ratios than the individual deviation approach.

4 Conclusions and future work

A link-based anomaly detection method is proposed in this paper, which is based on two different aspects: deviation from the individual pattern, and deviation from the cluster pattern. Experiments on both network traffic and email communication data demonstrate the potential of this approach to capture anomaly more accurately than individual-based approaches while being robust to false alarms. The two parameters α and β are data-dependent and thus, need to be tuned for each dataset. A training process could be introduced for this purpose. Our long-term goal is to build a framework for detecting anomalies for several other kinds of communication networks. According to our results, even though these networks are in different contexts and have different feature sets, our general concept of anomaly detection is applicable to most of them.

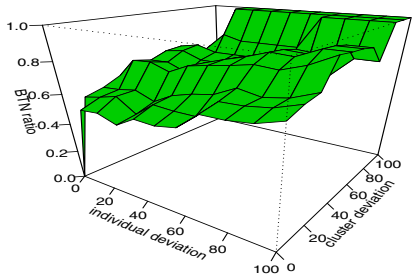

Figure 4. BTN ratios from different α and β

Acknowledgements

The authors gratefully acknowledge the financial support from the MITACS NCE and NSERC.

References

[1] Darpa 1998 intrusion detection data. www.ll.mit.edu/mission/communications/ist/corpora/ideval/.

[2] J. D. Banfield and A. E. Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, 49(3):803–821, Sep. 1993.

[3] C. Cortes, D. Pregibon, and C. Volinsky. Computational Methods for Dynamic Graphs. *Journal of Computational and Graphical Statistics*, 12(4):950–970, 2003.

[4] A. Lazarevic, L. Ertöz, V. Kumar, A. Ozgur, and J. Srivastava. A comparative study of anomaly detection schemes in network intrusion detection. In *Proceedings of the Third SIAM International Conference on Data Mining*, 2003.

[5] C. Priebe, J. Conroy, D. Marchette, and Y. Park. Scan Statistics on Enron Graphs. *Computational and Mathematical Organization Theory*, 11(3):229–247, 2005.

[6] P. Rousseeuw and K. V. Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.