An effective method for computer network anomaly detection

Dongpo Wang¹, Chengli Zhao¹, Xue Zhang¹, Boyu Liu², Xiang Li¹, Dongyun Yi¹

¹College of Liberal Arts and Sciences, National University of Defense Technology, Changsha, 410073, China
²Xinxiang College, Xinxiang, 453000, China
wdpmailbox@foxmail.com
donpowang@nudt.edu.cn

Abstract. With the increasing network threat, network anomaly detection has become a very challenging and indispensable task. In this article, we propose an anomaly detection algorithm through modeling computer network as temporal network. The active subnetwork is extracted from the original computer communication network and then projected to an undirected weighted network series, finally the abnormal behavior patterns of network series are detected based on eigenvectors. Data test experiment shows that the proposed algorithm has achieved very good results.

1. Introduction
Generally, according to nature, environment, behavior, and correlation cardinalities, the anomalies of the computing network can be divided into three categories, namely point anomalies, environmental anomalies, and integration anomalies [1]. Detection methods can be divided into supervised anomaly detection methods and unsupervised anomaly methods. Depending on the specific detection technology used, network anomaly detection can be divided into four categories, namely classification methods, statistical methods, and clustering methods, and the information theory method [2]. Among them, the development of statistical theory and network science has provided great help for the research of network anomaly detection. Ref [3] used a localized principal component analysis algorithm to continuously track the characteristics of the network neighborhood, Ref [4] proposed a fault detection algorithm based on eigenvectors, and applied it to the multilayer web network system represented by time-series graphs, Ref [5] modelled the computer network as a bipartite graph and then the bipartite graph is projected as a directed weighted graph. Different from these methods, in this article, we extract active subnet from the network, divide it by fixed time slot, and represent the divided networks with time-series undirected weighted graphs, and then perform feature decomposition for each network segment separately, after that we use sliding window to merge network features at different times and realize dynamic detection of network abnormalities. In the second part, we will introduce how to extract active subnets from the network. In the third part, we will introduce the heterogeneous representation of the network. The fourth part introduces the process of anomaly discovery. Finally, we will introduce our experiment.

2. Active subnet discovery
We model the communication record of a time slice as two parts including “Source IP” and “Destination IP”. The “Source IP” part represents all the nodes that send a connection message in a
time slice, and the “Destination IP” part represents all the nodes in the time slice that accept the message. Some nodes in “Source IP” only send connection messages, Some nodes in “Destination IP” accept messages, The rest of nodes existing both in “Source IP” and “Destination IP” not only send but also receive messages. Therefore, we can just use “Source IP” or “Destination IP” communication status to indicate the communication status of the entire network, which will cause a part of the communication details to be lost but will greatly reduce the computational complexity.

2.1. Definition
According to the previous discussion, we will sort the “Source IP” node and the “Destination IP” node based on the number of records related to them, to get a set of descending list of “Source IP” node and “Destination IP” node based on the number of records recorded as $S_A$ and $D_A$, $n_x(i)$ defines the number of network communication records of the $x$th element in the representation, and $n_x(i) \geq n_x(j)$, if and only if $0 < i \leq j < |S_A|$, $|S_A|$ represents the modulus of the set $S_A$. Similarly, the number of communication records corresponding to the IP in the representation of $D_A$ can be defined as $n_x(i)$, which can be known as a monotone non-increasing function by definition. For $S_A$ and $D_A$, we take the node set with larger difference of $n(x)$ as the selection set of active nodes. The reason is that the faster the decline of $n(x)$, the higher the concentration of active nodes, and the more beneficial it is to reduce the amount of calculation and find “super-active nodes”. For the convenience of discussion, we assume $S_A$ as a selection set of active nodes.

In a given time window, the network will be divided into several time slices, assuming that there are a total of $k$ time slices, with the node sequence number $x$ as the independent variable, and the nodes in the node sequence set as the dependent variable, to obtain a node set about the node sequence function $s(x)$,

$$ s(x) = \{S_A(i) \mid 0 < i \leq x\} \tag{1} $$

Thus, the nodes set functions in $k$ time slices can be obtained respectively, $s_x(i), 1 \leq i \leq k$.

The definition function $f(x)$ represents the modulus of the intersection of node functions in $k$ time slices,

$$ f(x) = \bigcap_{i=1}^{k} s_i(x) \tag{2} $$

The definition function $g(x)$ represents the modulus of the union of node functions in $k$ time slices,

$$ g(x) = \bigcup_{i=1}^{k} s_i(x) \tag{3} $$

Intuitively understand that among the top $x$ nodes with the highest activity in each time slice, $f(x)$ represents the number of the most active nodes present in each time slice, and $g(x)$ represents all these most active nodes in $k$ time slices total.

2.2. Optimal objective function
It can be seen from the definition in 2.1 that $f(x)$ and $g(x)$ are non-decreasing functions. According to the above discussion, in order to ensure the consistent continuity of the network, our goal should be to ensure the stability of the network structure within a certain time interval, from these we can get the objective function $h(x)$,
From the definition, we know that $h(x)$ is a discrete function, and our goal is to find the maximum value $h(x^*) = \max(h(x))$, the set of hyperactive nodes we are looking for is $A = \bigcup_{i=1}^{k} \chi_i(x^*)$. We call the network composed of “super-active nodes” active subnet.

3. Network heterogeneous representation

For any two nodes in the "super-active node" set $A$, we define $d_{i,j}$ as the number of links sent by the node $i$ to the node $j$. $d_{i,j}$ indicates the strength of the connection between nodes, $d_{i,j}^k$ indicates the joint strength of the node $i$, $j$ and the outgoing degree of the node $k$,

$$d_{i,j}^k = \frac{1}{2}(d_{i,k} + d_{j,k})$$

considering the suddenness of the change of the network communication flow, the logarithmic conversion of the traffic is used to calculate,

$$\overline{d}_{i,j}^k = \ln(d_{i,j}^k + 1)$$

node $i$ and node $j$ share $m$ common out-degree connection nodes, we use the weighted common connection strength to represent the out-degree dependence relationship between node $i$ and node $j$:

$$D_{i,j} = \sum_{k=1}^{m} \sum_{1 \leq d_i \leq m} \overline{d}_{i,j}^k \times d_{i,j}^k$$

Therefore, for any two nodes, there is such a dependency relationship between them, so that we project the original computer network into a more dense network structure. In contrast, we can define the in-degree dependency relationship, $d_{i,j}^{i,j}$ represents the in-degree joint connection strength of node $i$ and $j$, similar to the above, we can get the following content:

$$d_{i,j}^{i,j} = \frac{1}{2}(d_{i,k} + d_{k,j})$$

$$\overline{d}_{i,j}^{i,j} = \ln(d_{i,j}^{i,j} + 1)$$

$$D_{i,j} = \sum_{k=1}^{m} \sum_{1 \leq d_i \leq m} \overline{d}_{i,j}^{i,j} \times d_{i,j}^{i,j}$$

By definition, we know that the adjacency matrix of the projection network is a non-negative symmetric matrix. In this way, we get an undirected graph whose adjacency matrix is real symmetric.

4. Abnormal discovery

For the modelled temporal network, we can describe it with a series of network fragments, $G_t = (V_t, E_t), t = 1, 2, \cdots, k$, and its corresponding adjacency matrix and “behavior vector” can be expressed as $\{D_1, D_2, \cdots, D_k\}$ and $\{v_1, v_2, \cdots, v_k\}$ respectively. We assume that the state of network is a relatively stable within $W$ time windows, then we fuse the “behavior vector” of $W$ time windows. Firstly, we construct the “behavior vector” matrix $U(t)$:
\[ U(t) = [v_{t}, v_{t+1}, \ldots, v_{t+w-1}] \] (11)

Then, we define the fusion vector \( r(t) \):
\[
r(t) = c \sum_{i=1}^{w} \alpha_{i} v_{t-i+1}
\] (12)

Where \( \alpha_{i} \) is the adjustment parameter, \( c \) is the normalization constant, and the fusion vector satisfies \( r^{T}r = 1 \), in order to obtain \( \alpha_{i} \), we use the following extreme value principle [4]:
\[
\alpha(t) = \arg \max_{\alpha} \left\| \sum_{i=1}^{w} \alpha_{i} v_{t-i+1} \right\|^{2}, \alpha^{T} \alpha = 1
\] (13)

Known from (11) and (12):
\[ r(t) = cU(t)\alpha(t) \] (14)

From the Lagrange equation we can get:
\[
\frac{d}{d\alpha} \left[ \alpha^{T}U(t)^{T}U(t)\alpha - \lambda \alpha^{T} \alpha \right] = 0
\] (15)
\[
U(t)^{T}U(t)\alpha = \lambda \alpha
\] (16)

From the normalization conditions, we can know that \( c = 1/\sqrt{\lambda} \) and the fusion vector \( r(t) \) is the left singular vector of \( U(t) \).

In order to evaluate the change of the vector, we use \( z(t) \) as a metric to measure the change of the vector [4], where:
\[
z(t) = 1 - r(t-1)^{T}v_{i}
\] (17)

An upper quantile of \( Z \) can be used as a threshold, when the value at the time exceeds the threshold, it means that an abnormal event occurs at that time.

5. Experiment
To verify our algorithm, we used the intrusion detection data “ISCX2012” [6] of the Canadian Cybersecurity Institute, which contains 6 days of network activity from June 11, 2010 to June 16, 2010.

5.1. "Super-active Node" Discovery

5.1.1. "Super-active node" set selection
As shown in Fig. 1, we analyzed the changes in the traffic volume of the top 50 IP addresses sorted by the number of communication records in the source IP address for all 6 days. It can be seen from the results in the figure that the source IP address has a faster convergence rate than the destination IP address, so we use the source IP address node sequence set as "super "super-active node" selection set.

5.1.2. Optimal value calculation
Fig. 2 shows the change of \( f(x), g(x), h(x) \) with the increase of the number of nodes. From the figure, it can be seen that \( f(x) \) increases rapidly. When it increases to a certain number, \( g(x) \) increases slowly, \( h(x) \) increases first and then decreases and there is a maximum value corresponding to \( x^{*} = 26 \), what we are looking for is the continuously active node \( \bigcap_{i=1}^{6} S_{i}(26) \), and the “super-active node” we are looking for is \( \bigcup_{i=1}^{6} S_{i}(26) \).
5.2. Anomaly detection results
In the experiment, we used $W = 7$ as the sliding window value to verify the experiment. On this basis, we mark the different moments after the division. If there are records marked as abnormal within a moment, then this moment will be marked as abnormal. At this time, both the source IP address and the destination IP address of this record marked as abnormal will be marked as abnormal at this time. Fig. 3 shows the value of the outliers at different times under the given sliding window value. The time marked with an “+” in the figure is the time when an abnormal record is marked in the data set.

It can be seen from the figure that after a certain threshold is given, whether an abnormality occurs at the current moment can be judged according to whether the threshold is exceeded. Table 1 shows the abnormal detection results with two kinds of projection results under the priority of precision.
Table 1. Anomaly detection results

| Abnormal type                                                                 | Alarm times | Precision | Recall ratio |
|------------------------------------------------------------------------------|-------------|-----------|--------------|
| Out-degree joint projection                                                  | 12          | 1         | 0.188        |
| Unmarked anomalies, internal penetration, HTTP denial of service,            |             |           |              |
| distributed denial of service of IRC botnet, brute force cracking of SSH      |             |           |              |
| In-degree joint projection                                                   | 7           | 1         | 0.109        |
| Internal penetration, distributed denial of service of IRC botnet             |             |           |              |
| total                                                                        | 18          | 1         | 0.28         |
| Unmarked anomalies, internal penetration, HTTP denial of service,            |             |           |              |
| distributed denial of service of IRC botnet, brute force cracking of SSH      |             |           |              |

6. Summary and conclusion

We have proposed a new method for detecting computer network anomalies. First, we modelled the computer network as a temporal network and processed the computer network according to the temporal method. Second, we proposed the concept of “super-active node” and described the discovery method of “super-active nodes” in the temporal network; again, we proposed a network projection method based on the joint connection of out-degree and in-degree. Finally, we modelled the projected network as a dynamically weighted graph method, and realized the discovery of abnormal events from the perspective of the time series of the graph, and verified the method experimentally effectiveness.

Compared with other existing network anomaly detection methods, we do not need to use large-scale data set training to build any normal or abnormal models, thus avoiding the limitations of the data set itself, and our method also reduces a lot of calculations, which makes our method more efficient, but at the same time, our method also has some limitations. For a large-scale and more drastic network, if “super-active nodes” cannot persist or abnormal events unrelated to “super-active nodes” occur, there may be difficulties in using our method for anomaly detection.

References
[1] Chandola, V., Banerjee, A., and Kumar, V. (2009) Anomaly detection: A survey. ACM Computing Surveys (CSUR), 41, 15:158.
[2] Ahmed M, Mahmood A N, Hu J. A Survey of Network Anomaly Detection Techniques[J]. Journal of Network and Computer Applications, 2015, 60:19-31.
[3] Yu W, Aggarwal CC, Ma S, Wang H. On anomalous hotspot discovery in graph streams. In: Proceedings of the 13th IEEE International Conference on Data Mining (ICDM), Dallas, TX, 2013.
[4] Ide, T. and Kashima, H., Eigenspace-Based Anomaly Detection in Computer Systems, ACM SIGKDD 2004, pp.440-449.
[5] Eslami M, Zheng G, Eramian H, et al. Anomaly detection on bipartite graphs for cyber situational awareness and threat detection[C]/ 2017 IEEE International Conference on Big Data (Big Data). IEEE, 2017.
[6] ISCIDS2012)[OL].https://www.unb.ca/cic/datasets/ids.html Canadian Institute for Cybersecurity.