A Hybrid Approach of HTTP Anomaly Detection

Yang Shi, Shupei Wang, Qinpei Zhao(✉), and Jiangfeng Li(✉)

School of Software Engineering, Tongji University, Shanghai, China
{qinpeizhao,lijf}@tongji.edu.cn

Abstract. Security technology in computer network including anomaly detection is increasingly playing an important role in the government and protection of Internet along with its popularity. Anomaly detection uses data mining techniques to detect the unknown malicious behavior. Various hybrid approaches have been proposed in order to detect outliers more accurately recently. This paper proposes a novel hybrid of clusterings and graph to detect anomaly. We introduce a new holistic approach in a common bipartite scenario of users from intranet accessing to Internet that utilizes different types of clusterings for the individual feature data to find the outliers and then a graph model to take advantage of the relational data naming network to enhance anomaly detection. The framework solution has several advantages: taking consideration of individual feature data and relational data, keeping open to extend different types of clusterings, easily appending more domain knowledge.

Keywords: Anomaly detection · Data mining · Clustering · Bipartite graph

1 Introduction

Network traffic anomalies are unusual and significant changes in terms of network traffic amount. Given a large amount of network traffic, how to precisely detect anomalies is very challenging. In particular, it is hard to get an enough amount of ground truth labels to identify whether one is normal or abnormal. Classifiers usually work with labels. Instead clusterings do not need any label data, and have been used for network anomaly detection. Indeed, the classification algorithms often can get a higher detection rate than clusterings.

To overcome the issue above, in this work, we are inspired by [9], and propose to integrate the graph based anomaly inference and cluster ensemble techniques to take full advantage of feature data and network behaviour relational data.

Specially, we first adopt clustering ensemble to combine the strength of individual clustering algorithms (such as K-means, DBSCAN, Birch and Meansshift). With the clustering ensemble of such four clustering algorithms, we can label a small amount of anomalies with high confidence. Next, we model the relations between HTTP clients and servers as a bipartite graph, and apply Markov

Random Field (MRF) on the bipartite graph to infer that the nodes are normal or abnormal. Finally, we solve the inference problem by Loopy Belief Propagation (LBP). Figure 1 illustrates the overall structure of hybrid of clustering and graph-based inference.

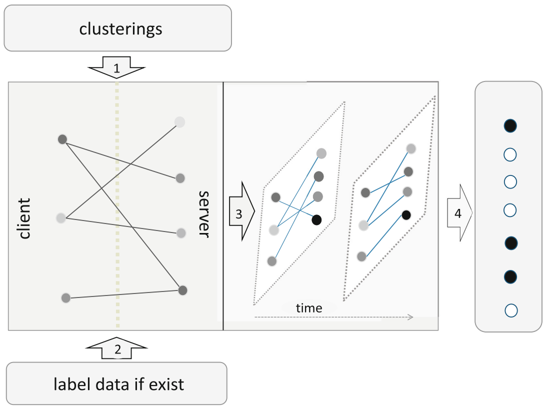


Fig. 1. Overall structure of hybrid of clusterings and graph.

Our main contributions are as follow: (1) We introduce a novel approach for network anomaly detection problem, which ties together relational data with high dimension individual information, it utilizes all of graph, (2) We recommend clustering as the predictor of prior potential, that could be applied as a general method in others unknown network and domain and keep open to extend different types of clusterings, (3) Our work is in a completely semi-supervised fashion and amendable in situation of better prior value or lack of prior.

We evaluate our method on real world data from a Internet security company. Verification on known data provided by the company in the semi-supervised fashion is more than 80% accuracy rate of malicious inference. The improvement of accuracy rate with respective to clustering algorithm is about 20 %.

2 Related Works

Different techniques in network security domain or data mining had taken part in the mission of network anomaly detection on their perspective. The signature-based approaches look for pattern that matches known signatures. For example, dos activities can be detected based on the uniformity of ip address [5]. While the traditional signature-based detection will face great challenges as they will likely be outpaced by the threats created by anomaly authors [7]. The non-signature-based approach, naming statistical techniques are applied, too. These techniques analyse the individual information. Classification and cluster algorithm can detect the network anomalies taking account of collective information of more features. The previous works [2,4] use classification algorithm to detect network anomalies. [2] choose the One Class Neighbor Machine and the recursive Kernel-based Online Anomaly Detection algorithm to detect anomalous behaviour in a distributed set of network measurements.

Some previous literatures has used clustering ensemble for anomaly detection. In [6], they apply ensemble clustering to anomaly detection, hypothesizing that multiple views of the data will improve the detection of attacks. Each clustering rates how anomalous a point is; ratings are combined by averaging or taking either the minimum, the maximum, or median score.

Using any particular algorithm alone does not yield proper results. In past few years approaches have been made by either combining or merging different algorithms together [1]. [3] concentrated on the development of performance of naive bayesian and ID3 algorithm and his hybrid algorithm was tested in Knowledge Data Discovery cup. [8] described about the ensemble approach which used decision tree and support vector machine. Besides the above cascading supervised techniques, there are number of unsupervised and supervised learning algorithm whose combinations can be made in the recent past years. [10] combine k means and ID3 for classification of anomalous and normal activities in computer address resolution protocol traffic. Similar hybrid approaches have applied in the SVM classification and k medoids clustering, k medoids clustering and naive bayes classification, one class and two class SVM, etc.

The classification and clustering algorithms, however ignore the network relational information when detecting the anomalies. Graph mining methods, which take advantage of relational information, have been successfully applied in many domain [7] from authority propagation to fraud detection. In [9], a new holistic approach that utilizes clues from all meta data as well as relational data to spot suspicious users and reviews in a online review system was proposed. In [7], they formulate the classic malware detection problem as a large-scale graph mining and inference problem, and show how domain knowledge is readily incorporated into the algorithm to identify malware.

3 Methodology

3.1 Clustering

Clustering can be defined as a division of data into group of similar objects. its advantages is able to detect anomaly without prior knowledge. Beyond the robustness provided by each algorithm, *Clustering ensemble* has been recently considered in variety of different areas used as a meta-clustering method to improve the robustness of clustering by combining the output of multiple algorithm. It can be defined as the optimization problem where, given a set of m clusterings, we want to find the clustering that minimizes the total number of disagreements with the m clusterings. There are various methods of clustering that can be applied for the anomaly detection. Intuitively, different types of clusterings or different parameters of one clustering are picked up to attain anomaly from a diverse perspective. Each chosen clustering algorithm is regarded as “anomaly voter”. Meanwhile each node is detected and classified as anomalous or normal by “anomaly voters”. Some node is detected to be anomaly by more voters, it’s assigned a higher suspicious score, while some node is detected to be anomaly by less voters, it’s assigned a lower suspicious score. The rest nodes, which are not detected by any voters, are regarded as normal with a high unsuspicious score.

$$score(node_i) = \begin{cases} a, \quad if \quad voter = 0 \\ 0.5 + 0.5 * (voter/N). \end{cases} \tag{1}$$

The Eq. 1 essentially transfers the clusterings binary result to a numerical value (the parameter a is constant and less than 0.5, N is the total number of “anomaly voter”, voter is the number of “anomaly voters” which regard the node as anomalous). The value is initial prior probability of each node’s goodness or badness.

It’s critical and skillful to balance the amount of anomaly detected by different candidate clusterings. We firstly take advantage of the high efficiency of BIRCH with fast running time, run a large number of experiments with different radius parameter, obtain the distribution of anomaly quantity and choose the proper anomaly quantity. Following the clustering result of Birch, we will tune the parameter for the rest of three clustering algorithms. For fairness, we ensure that the number of outliers by each clustering algorithm is approximately equal.

3.2 MRF Model

Review of MRF. Markov Random Field (MRF) is a class of graphical models particularly suited for solving inference problems with uncertainty in observed data. Markov network or undirected graphical model is a set of random variables having a Markov property described by an undirected graph. The Markov property in Markov random field is that the probability distribution of one node’s state depends on the nodes’ state surrounding it and not on other nodes. In a MRF, each node is in any of a finite number of states. The dependency between a node and its neighbors is represented by a *propagation matrix* (ψ), where $\psi(i, j)$ equals the probability of a node being in state j given that it has a neighbor in state i.

Problem Description. We first give the intuition of using MRF of anomaly detection. When attackers initiate network traffic attacks and cause anomaly on target nodes, the target nodes’ neighbors (and also neighbors of neighbors and etc.) could be harmed by the attacks. Thus, the network traffic behaviour of such neighbors could deviate from the majority of normal nodes. With help of MRF, we can detect how network traffic is prorogated across node edges to detect network traffic anomalies by aggregating network traffic from neighbor nodes (together with neighbors of neighbors and etc.) towards target nodes. In a nutshell, we adopt MRF to the client-server bipartite graph in Fig. (1) and formulate the anomaly detection problem as a network classification task.

An MRF model consists of an undirected graph where every node i is associated with a random variable Y_i that can be in one of a finite number of states or class represented by labels. The label of nodes is assumed to be dependent only on these nodes surrounding with edge and independent on all the other nodes in the network. The joint probability of node is written as a product of individual and pairwise factors:

$$P(y) = \frac{1}{Z} \prod_{Y_i \in V} \varphi_i(y_i) \prod_{(Y_i, Y_j) \in E} \psi_{ij}(y_i, y_j) \tag{2}$$

where y refers to an assignment of labels of all nodes, y_i denotes node i's assigned label and Z is the normalization constant which make the probability of all states of one node sum up to 1. The individual factors φ is called prior potentials and represent initial class probabilities for every node, often initialized based on prior knowledge. For the prior potentials, one can estimate them by proprietary formula or using external information. Prior potentials are obtained mainly from the clustering score.

In our whole design and implementation, one of the goal is to require the credible labels through combination of general and common techniques rather than proprietary formula in some field. The pairwise factors ψ are called compatibility potentials and capture the likelihood of a node with label y_i to be connected to node with label y_j.

Table 1. Compatibility potential matrix

	Server	
Client	Abnormal	Normal
Abnormal	$0.5 + \varepsilon$	$0.5 - \varepsilon$
Normal	$0.5 - \varepsilon$	$0.5 + \varepsilon$

The compatibility potentials are set based on several intuitions reflecting the modus-operandi of abnormal nodes. The anomalous server usually access the anomalous clients with larger probability than with normal clients, likewise the normal servers connect to the normal clients with larger probability. The compatibility potential of client and server node is as follow in Table 1 (ε is set 0.01), it's elegant form of the compatibility potentials ψ and convenient for calculating Eq. (2).

Given the model parameter (φ_i and $\psi_{i,j}$), the task is to infer the maximum likelihood assignment of states to the random variables associated with the nodes, naming to find the y that maximizes the joint probability of the network. The inference problem is an NP-hard task. The enumeration of all possible states is exponential and thus intractable for large network.

Fortunately, the *belief propagation* algorithm has been proven very successful in solving inference problem over graph in various field (e.g. image restoration, error-correcting code). In particular, the iterative message passing in BP intuitively simulate the client-server request and response behaviour and network traffic aggregation toward target nodes. Thus, iterative approximate inference algorithm such as *Loopy Belief Propagation* (LBP) is a good choice for our project. In the next section, we describe the details of LBP to solve our problem.

3.3 LBP Algorithm

LBP infers the label of nodes from prior potential of the node and compatibility potential involving with the node's neighbours through iterative message passing between all pairs of node i and j. $m_{i,}(x_j)$ denote the message sent from node i to node j. This message represents i's opinion and support about j's all states. Each node considers all the messages from his neighbors and decides the probability of all his states in one iteration. At the end, all messages come to convergence and almost stay changeless. Each node's goodness or badness is determined and estimated marginal probability and is also called belief, $b_i(x_i) \approx P(x_i)$. In a binary state, while the probability of one state is below the fair threshold of 0.5, the node is possibly inclined to the other state. At details, messages are obtained as follows. Messages associated with one edge $edge_{ij}$ are $m_{ij}(x_j)$ and $m_{ji}(x_i)$. Messages are normalized over its recipient, so that all messages point to node j sum to one in every iteration.

$$m_{ij}(x_j) \leftarrow \sum_{x_i \in X} \varphi(x_i)\psi_{ij}(x_i, x_j) \prod_{k \in N(i)-j} m_{ki}(x_i) \tag{3}$$

where node k belongs to the neighboring of node i excerpt for node j. $\varphi(x_i)$ is called the prior potential and the accumulation refers to all states of node i. Formally, $\psi_{ij}(x_i, x_j)$ equals the probability of a node i being in class x_i and its neighbor j in class x_j.

The Algorithm 1 stops if the maximum number of iteration reach or messages converge within threshold. Under most circumstances, the algorithm converges in practice, even though convergence is not guaranteed theoretically. When the algorithm ends, the final beliefs of node are determined as follow:

$$b_i(x_i) = k\varphi(x_i) \prod_{x_j \in N(i)} m_{ij}(x_i) \tag{4}$$

where k is a normalization constant making sure one node's beliefs in all state sum to 1.

Malicious server detection is a long term task, along with new graph coming in, the quantity of client nodes, server nodes and edges increase out of proportion. The incremental LBP in Algorithm 2 is the solution for continuous input and avoiding too many servers node connecting to one clients leading LBP can't work even in insufficient iteration.

Algorithm 1. LBP

```
Require:
    client-server graph $G = (V, E)$
    compatibility potential $\psi$
    node's labels L
Ensure:
    class probabilities for each node $i \in V$
 1: for $e_{ij} \in E$ do
 2:     $m_{ij} \leftarrow 1, m_{ji} \leftarrow 1$
 3: end for
 4: repeat
 5:     for $e_{ij} \in E$ do
 6:         $m_{ij}(x_j) \leftarrow \sum_{x_i \in X} \varphi(x_i)\psi_{ij}(x_i, x_j) \prod_{k \in N(i)-j} m_{ki}(x_i)$
 7:     end for
 8: until messages stop changing in threshold or reaching max iteration number
 9: for $e_{ij} \in E$ do
10:     $b_i(x_i) = k\varphi(x_i) \prod_{x_j \in N(i)} m_{ij}(x_i)$
11: end for
12: return $b_i$
```

4 Evaluation

The evaluation of this approach is dependent much on obtaining the ground truth of all the servers located all over the world and going deep into the details of actions in the network, that is almost impossible mission. Fortunately we are provided with part of near-ground-truth. The data are provided by a security company. It uses professional technique and tools to detect anomalous behavior and host, keeps a blacklist database for several years. In the following, we describe our data, comparison experiments and then performance results.

4.1 Data Description

First of all, we describe the large data set that we infer the malicious server in the internet. The raw data is captured by network devices that keep trace one intranet with an amount of six thousand of clients from July 14 2015. Every entry contains property such as time, source ip address, destination ip address, port, uri, host and so on. The bipartite graph can be constructed easily according to the source and destination ip address. We extract 7 features for clustering algorithms, such as average length of http uri, average number of parameters in uri, average number of parameters in http cookie, total number of get, total number of post, burst request, total number of requests. In the period, these clients access to the internet server from all around the world with an amount

Algorithm 2. Incremental LBP

```
Require:
    client-server graph stream $GS = (G_1, G_2, \ldots, G_n)$
    compatibility potential $\psi$
    node's labels L
Ensure:
    class probabilities for each node $i \in V$
1: $b_0 \leftarrow L$
2: for $G_i \in GS$ do
3:     $b_i \leftarrow LBP(G_i, b_{i-1}, \psi)$
4: end for
5:
   $$b_i = \begin{cases} anomaly\ if & b_i > 0.5 \\ normal & else \end{cases}$$
6: return $b_n$
```

of 280 thousand by HTTP. The client ip of raw data is hashed to LAN address for information protection and can't be mapped back. In addition, we are provided about 3 thousand of servers with verification information by the company's anomalous server database, which includes one thousand malicious server plus two thousand benign addresses.

4.2 Experiment

We first evaluate the accuracy rate of an individual clustering algorithm for network anomaly detection in Fig. 2. By tuning the weight of each single feature in calculating distance given above, we attain the minimum, maximum and average accuracy rate of different clustering algorithm.

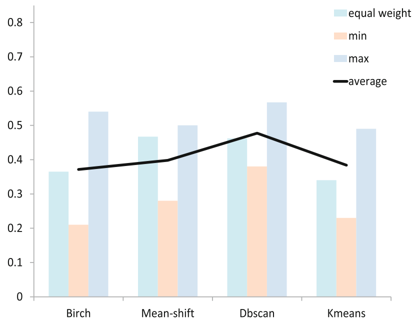
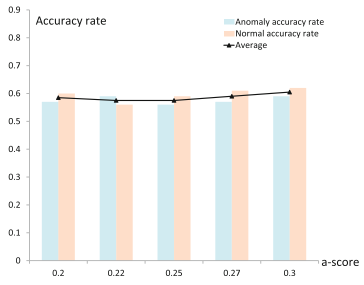
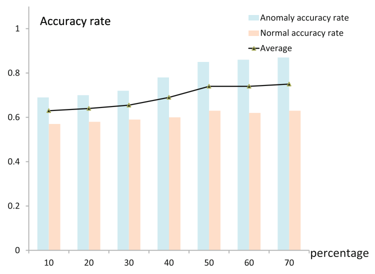


Fig. 2. Average accuracy rate of single recommended clustering algorithm.

It is not hard to find that the accuracy rate of an individual clustering algorithm is far from satisfaction. Thus, we combine the four clustering algorithms and graph, then evaluate the associated prediction accuracy. Here, we set the parameter a (used for Eq. 1 in clustering ensemble) to be 0.2, 0.22, 0.25, 0.27 and 0.3, respectively, and plot the associated accuracy in Fig. 3(a). Compared with the result in Fig. 2, the proposed approach can greatly improve the accuracy.



(a)



(b)

Fig. 3. (a) The horizontal axis refers to the experiment number mentioned above. (b) We set the known malicious server nodes with a high score of 0.95. The horizontal axis means the percentage of label data for training.

Ground-truth data are no doubt perfect choice for our algorithm in comparison with the result of cluster ensemble because they're malicious with high credibility. However, it's insufficient for the huge network and only a small quantity is available. Taking the most advantage of the known data and cluster ensemble, we combine both and design the semi-supervised experiment that should be adoptive for maximizing the accuracy in reality. We use part of blacklist database as prior labels and the rest as test data. In Fig. 3(b), we choose randomly from 10% to 70% of malicious servers as label and assign $\varphi_i(x)$ is $\{0.05, 0.95\}$, for the rest $\varphi_i(x)$ is based on Eq. 1. We choose the malicious inference accuracy and benign inference accuracy in the test data as the metric.

Acknowledgement. The authors in this work are sponsored by the Fundamental Research Funds for the Central Universities, the Youth Science and Technology of Foundation of Shanghai (15YF1412600), the Shanghai Sailing Program (17YF1420500) and the National Natural Science Foundation Committee of China under contract no. 61202382.

References

1. Agrawal, S., Agrawal, J.: Survey on anomaly detection using data mining techniques. Procedia Comput. Sci. **60**, 708–713 (2015)
2. Ahmed, T., Oreshkin, B., Coates, M.: Machine learning approaches to network anomaly detection. In: Proceedings of the 2nd USENIX Workshop on Tackling Computer Systems Problems with Machine Learning Techniques, pp. 1–6 (2007)

3. Farid, D., Harbi, N., Rahman, M.: Combining naive bayes and decision tree for adaptive intrusion detection. arXiv preprint arXiv:1005.4496 (2010)
4. Gaddam, S., Phoha, V., Balagani, K.: K-Means+ ID3: a novel method for supervised anomaly detection by cascading K-Means clustering and ID3 decision tree learning methods. TKDE **19**(3), 345–354 (2007)
5. Moore, D., Shannon, C., Brown, D., Voelker, G., Savage, S.: Inferring internet denial-of-service activity. ACM Trans. Comput. Syst. (TOCS) **24**(2), 115–139 (2006)
6. Munson, A., Caruana, R.: Cluster ensembles for network anomaly detection. Technical report, Cornell University (2006)
7. Nachenberg, C., Wilhelm, J., Wright, A., Faloutsos, C.: Polonium: tera-scale graph mining and inference for malware detection (2011)
8. Peddabachigari, S., Abraham, A., Grosan, C., Thomas, J.: Modeling intrusion detection system using hybrid intelligent systems. J. Netw. Comput. Appl. **30**(1), 114–132 (2007)
9. Rayana, S., Akoglu, L.: Collective opinion spam detection: bridging review networks and metadata. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 985–994 (2015)
10. Yasami, Y., Mozaffari, S.: A novel unsupervised classification approach for network anomaly detection by k-Means clustering and ID3 decision tree learning methods. J. Supercomput. **53**(1), 231–245 (2010)