Inconsistency Extraction using Advanced FP-Growth Algorithm

Pravin Gaikwad
ME (Computer Network)
Department of Computer
Engineering, SCOE, Pune-41

Jyoti Kulkarni
Assistant Professor
Department of Computer
Engineering, SCOE, Pune-41

ABSTRACT
Inconsistency or Anomaly extraction refers to the automatically finding a large set of flows observed during an anomalous time interval, the flows associated with anomalous events. It is valuable for root causes analysis, network forensics, anomaly modeling, and attack mitigation. In this paper, histogram based detectors are used which provide a meta-data which is useful for identifying suspicious flows and then apply association algorithm like Advanced FP-Growth Algorithm to summarize and find anomalous flows. Using rich traffic data from a network, Paper show that a technique efficiently finds the flows associated with anomalous events. In addition, an algorithm reduces the both in runtime and the main memory consumption. The inconsistency extraction method significantly reduces the working hours needed for anomaly detection system more practical.

Keywords
Association rules, computer network, data mining, FP-Growth, compound single linked list

1. INTRODUCTION

1.1 Motivation
Identifying the network anomalies is critical for the timely mitigation of events, like failures or attacks that can affect the performance and security of a network. An anomaly is defined as a "Deviation or abnormality from the normal or common order. While studying anomalies affecting computer networks, it considers actions that differ from normal network behavior, such as significantly increased traffic, use of new protocols and malicious attacks. Anomaly detection techniques are the last line of defence when other approaches fail to detect the security threads and other security problems. The foremost challenge in identifying and detecting anomalies is the fact that they can be caused by a vast set of events. While studying, researchers have pose number of interesting research problems like modeling, involving statistics and efficient data structure. Nevertheless researchers have not yet gain widespread adaption, as a number of challenges, like calibration and reducing number of false positive rate remain to be unsolved. In this paper, Inconsistency extraction method are interested in the problem of identifying the traffic flows correlated with an anomaly during a time interval with an alarm. Anomaly extraction reflects the goal of gaining the more information about the anomaly alarm, which, without additional meta-data, is often meaningless for the network operator. Data mining techniques are used to identify anomalous behavior. Identified anomalous flows can be used for a number of applications, like network forensics, root-cause analysis of the event causing an anomaly, anomaly modeling, and improving anomaly detection accuracy.

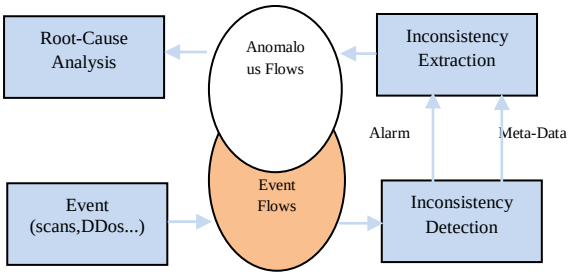

Fig. 1. High level goal of inconsistency extraction is to filter and summarize the set of anomalous flows that coincide with flows caused by network event such as DoS attack or scans.

1.2 Inconsistency or Anomaly Extraction
In Fig. 1, it presents a high level goal of anomaly or inconsistency extraction. In the bottom of the figure, events with network level footprint, like attack or network failure, trigger event flow, which after analysis by a histogram-based anomaly detector, may raise an alarm. Ideally, method would like to extract all triggered event flows. The goal of inconsistency extraction process is to find a set of anomalous flows coinciding with event flows. An inconsistency detection system may provide a meta-data relevant to an alarm that helps to find the sets of candidate anomalous flows. For example, inconsistency detection systems analyzing histograms may indicate histogram bins than an anomaly affected, e.g., range of Port numbers and IP addresses. Such meta-data can be used to decrease the candidate anomalous flows to these that have an affected port numbers and IP address. To extract an anomalous flow from a large network data, one could build a prototype describing normal flow characteristic and to identify deviating flows. To build such prototype is very challenging issue due to wide variability of flows characteristic. Similarly, one could match flows during an interval with flow from normal or past interval to the new flows or flow with significant increase or decrease in their volume to search for changes or to identify new flows that were not previously observed [9] [13]. Such approaches essentially perform anomaly detection and could be used to identify anomalous flows.

1.3 Contribution
In this paper, it take an alternative approach to identify anomalous flows that combines and consolidate information from multiple histogram-based detectors and an Advanced FP-growth algorithm is applied to speed up the system rate and decrease the false positive rate as compared to Apriori and FP-growth algorithm. Compared to other possible approaches, inconsistency extraction method does not rely on

past data for normal interval or normal models. Intuitively, each histogram-based detectors provide and additional view of network traffic.

A detector may provide a set of candidate anomalous flows. The main reason of applying an association rule is that *Anomalies typically result in many flows with similar characteristic*, e.g., common IP address or port numbers, since they have a common root-cause, like network failure or a scripted denial-of-service (DoS) attack.

1.4 Road Map

The remaining of this paper is organized as follows. In the next section paper provide Literature Survey information. In section III presents many technical details relevant to the design of an effective Inconsistency Extraction System. Finally, in Section IV, outline and evaluation results and conclude in Section V.

2. LITERATURE SURVEY

A number of studies have focused on developing volume based anomaly detection system. In [2], a good characterization of different types of anomalies and proposed wavelet based methods for change detection is provided. In [3], a general framework that aims to identify anomalies in wide network traffic data is introduced and are successful in identifying anomalies that result in traffic volume deviations. But they are failed in detecting stealth attacks, such as port scanning, that do not result in notable changes in traffic volume.

Feature-based techniques are different from volume-based techniques with respect to the measurement metric. The metrics used in a feature-based technique is extracted from packet header fields (referred to as traffic features), and commonly includes [IP Address, Port Numbers, Protocols, Packet size, Flow duration]. The motivation behind this approach is to use traffic feature distributions (TFDs) created by capturing each traffic feature during a time interval, which has proved to detect a wider range of anomalies compared to volume-based techniques. Lakhina and Diot, the idea of using traffic features to identify a wide range of anomalies is introduced [4]. Diot et al [4] proposes a method based on clustering of entropy to classify the anomalies found by Principal Component Analysis (PCA) anomaly detector. Since, this entropy residual is an internal variable of the PCA detector, the main limitation of this technique is that it only classifies anomalies that are visible by PCA on entropy [5]. And also it is lagging in the ability to output specific feature value and two completely different features have same entropy value. Kind and Dimitropoulos [6] proposes a histogram based detectors to detect different kinds of anomalies based on traffic features, but it is not focusing on the anomaly extraction problem. This work introduces such method to extract packet features. In paper [7], a tool called URCA (Unsupervised Root Cause Analysis) that searches anomalous flows by iteratively eliminating subsets of normal flows is introduced. Although, it requires to repeatedly evaluate an anomaly detector on different flow subsets, which can be costly. Compared to this work, this system show that simply computing frequent item-sets on pre-filtered data is sufficient to identify anomalous flows. Dowitcher [8] is a scalable system for worm detection in backbone network. Part of the system automatically constructs a flow filter mask from intersection of suspicious attributes provided by different detectors. But in this system, inconsistency extraction method take union of meta-data because intersection of attributes can miss anomalous flows and union of meta-data combined with association rule mining gives better result. D. Baruckhoff et al [1] introduces an automated anomaly extraction system which extract features is going to affect the anomaly. While applying an association rule mining (Apriori Algorithm) on such a large data, a computational cost and time required to find anomalous flows is high, to overcome this problem this system introduced a clever algorithm called Advanced FP-growth based on compound single linked list [10].

Association rule mining has been successfully applied to different problems on networking. The Apriori algorithm [11] is the best known previous algorithm and it uses an efficient candidate generation procedure, such that only the frequent item-sets are used to construct candidates at the next level. Although it requires multiple database scans, as many as the longest frequent item-sets. Han and Yin [12] proposes a completely different and efficient algorithm to mine frequent items without generating candidate sets: FP-growth. It just need scan database twice and mining frequent patterns in the constructed FP-tree. A study on the performance of the FP-growth method shows that it is efficient and scalable and is about an order of magnitude faster than the Apriori algorithm. However, it has its own problems, during mining frequent item-sets, mass of conditional pattern trees are generated recursively. It cost a lot of time to generate and release these trees. Besides, to overcome form these problems, system proposes an advanced FP-growth algorithm based on compound single linked list [10].The algorithm introduces the compound single linked list to improve the structure of the FP-tree.

3. IMPLEMENTATION DETAILS

3.1 Methodology

An overview of the Inconsistency extraction method is given in fig.2.An n number of histogram-based anomaly detectors supervises the network traffic and detects anomalies in an online fashion [1]. Upon detecting an anomaly, system use an different histogram-based anomaly detectors to pre-filter a set of suspicious flows. This pre-filtering process is necessary to eliminate a large amount of normal flows. A frequent item-set in the set of suspicious flows is produced by applying association rule mining [1]. The basic assumption behind applying association rules is that frequent item-sets in the pre-filtered meta-data are often related to anomalous event.

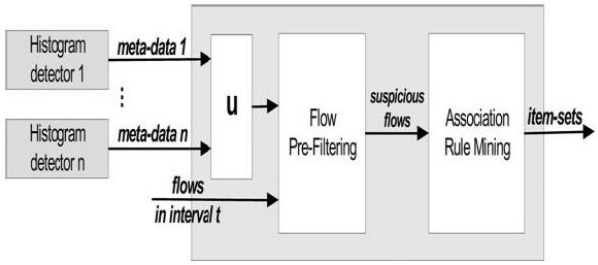

Fig 2 . Overview of our approach to the Inconsistency extraction problem

The association algorithm like Advanced FP-growth algorithm is applied which improves the algorithm both in main memory consumption and runtime, which does its work without any complex data structure [10]. The entire inconsistency extraction process is automated and can take place both in offline and online fashion.

3.1.1 Flow Prefiltering:

Pre-filtering process usually removes a large part of the normal traffic. This process is adorable for two reasons. 1] It generates a substantially smaller data dataset that results in faster in the following steps. 2] It increases the efficiency of association rule mining by removing flows that could result in false-positive item-sets. An important detail of our method is that. It keeps flows matching any of the meta-data instead of flows matching all the meta-data. Method takes union of flows matching meta-data instead of intersection of flows matching meta-data. Taking union is important because identified meta-data can be flow disjoint, meaning that metadata appear in different flows, in which case the intersection is empty. It shows that the union results in less false positive that the intersection which may miss whole anomaly.

3.1.2 Frequent Item-Set Mining:

Association rule plays an important part in the field of data mining, which is used to mine the association rules in a given data set [11]. It divides the mining process into two steps: (1) the main and most challenging part is finding frequent item-sets. (2) from frequent item-sets generate strong association rules. The main aim of applying association rule mining is that to find frequent itemsets to extract anomalous flows from large traffic data in time *t* interval. Our assumption for applying frequent item-set mining to inconsistency extraction problem is that anomalies typically result in large number of flows with similar characteristic. e.g., port numbers, IP address, since they have a common root cause analysis like scripted DoS attack or network failure. The transaction size is defined as the number of items present in a transaction. Each transaction has a set of five since each flow record has five correlated feature corresponding to its srcIP, dstIP, srcPort, dstPort, number of packets. For example the item i_1 = src_Port : 25 refers to a source port number equal to 25, while item i_2 = dst_Port : 25 refers to destination port number 25, a transaction cannot have to items same feature type.

Advance FP-Growth Algorithm Based On Compound Single Linked List:

1 The steps to construct the compound single linked list

1.1 The first scan of database is the same as the FP-tree. The scan of the database derives the set of frequent items (1-itemsets) and their support counts (frequencies). The set of frequent items is sorted in the order of descending support count. This resulting set or list is denoted L and an item header table is built.

1.2 The second scan of database is different from the FP growth. It is processing the items in each transaction in L order, and then inserting the items in each transaction into the single linked list recursively. The items‟ order in each single linked list is according to L order.

2 The realization of the pseudocode of the Advanced FP algorithm

2.1 Define storing structure:

Header Table:

Itemid	Support Count	Node-link
……	……	……
……	……	……

Frequent Itemsets:

FreItem	Pointer

Single linked list:

Item	Count	Pointer

```
for (i=1; i≤q; i++)
        do { scan number i's trasaction
           let p point to number I's first frequent itemsets;
           while (p≠Null)
        { if (p→next==Null) end;
        else { find p's pointing frequent item in header table
        traverse the single linked table that p→next points to
        during the traversal ;
if (the current node is in the corresponding single linked table)
incremented by 1
        else { generate a new node and insert it to single linked
table according to the order of L, the count is assigned by 1;
        }
        }
p= p→next
   }
}
```

The standard algorithm for generating frequent item-sets is FP-Growth and Apriori Algorithms [11] [14]. The main disadvantage of Apriori and FP-growth is that large number of candidate item-sets generation and a complex data structure. The Advanced FP-growth algorithm based on compound single linked list is introduces to improve the simplicity of the FP tree. The advanced FP-growth is mined in one direction, using the header table in the former FP-tree, storing them in a sequence table, ordering the frequent item-sets in descending sequence according to the minimum support, and then a compound single linked list is formed. Through traversing each transaction frequent item-sets stored in its compound single linked list, mining the frequent patterns directly without generating conditional FP-trees. Through the comparison between the advanced FP algorithm and the former FP-tree, it shows that the new one improves the algorithm both in runtime and the main memory consumption.

3.1.3 Histogram-Based Detectors:

Histogram-based anomaly detectors [6] [14] have been work well for detecting anomalous behaviour and changes in traffic flows distribution. Our histogram-based detector uses the KL (Kullback-Leibler) distance to detect anomalies. The KL distance has been successfully applied for anomaly detection in existing work [5] [7]. Each histogram-based detector supervises a flow feature distribution, like distribution of IP address and Port numbers. It assumes that for *n* histogram based anomaly detectors of *n* traffic features and have *m* histogram bins are constructed. During a time interval *t*, an anomaly or inconsistency detector module constructs

histogram for number of flows per traffic feature. At the end of time interval *t*, it computes for each histogram the KL distance between the distribution of reference interval and current interval distribution. The KL distance of given reference distribution *p* to discrete distribution *q* be.

$$D\,(p//q) = \sum_{i=0}^{m} p_i \log(p_i/q_i).$$

3.1.4 Histogram Cloning and Voting:

Histogram cloning is a promising technique applied to Histogram based Detections. The motivation behind it is to maintain multiple randomized histograms of same features; hence it will obtain additional views of network traffic. This technique is realized in [7] by creating *n* histogram-based detectors corresponding to *n* different traffic features. For each of the *n* features there are *m* bins per time interval, and by applying a hash function to each of the clones, one makes sure that each feature value is placed randomly into one of the m bins. This diverges from classical binning, which tends to place adjacent feature values (e.g. source ports) next to each other in a histogram. For histogram detection, KL distance is computed between every newly created distribution and a reference distribution, more specifically the distribution from the previous measurement interval.

4. RESULTS

In this section, it first describes the traces used for our experiments, and then evaluates each step of our approach for parameter settings. In particular, it evaluates the accuracy of method approach as well as the reduction in classification cost, in terms of item-sets or flows. And also it shows the comparative results of three algorithms (Apriori, FP-Growth and Advanced FP-Growth).

4.1 Dataset and Ground Truth

To validate of inconsistency extraction, it used a traffic flows trace coming from various clients connected in network. it have been collecting nonsampled and nonanonymized flows from clients. On average, system collects 15000 packets per 60sec crossing the server used for our experiments. To generate datasets for evaluating the Advanced FP-Growth algorithm, it computes the KL distance time series for the 1min of data for the following features distribution: Source IP, Destination IP, Source Port, Destination Port, Packet Size in packets. To determine the root cause of each anomaly, it extracts all traffic flows in anomalous time interval and analyased the distribution of the five features, the number of flows. From these flows system computes the set of candidate anomalous in anomalous interval using Advanced FP-growth algorithm. After applying Advanced FP, it manually analyzed the found frequent item-sets and identified true positive.

4.2 Accuracy of frequent items-sets

After meta-data has been identified by previous functions, the corresponding traffic flows are filtered and subsequently processed by the item-set mining process. The accuracy in terms of correctly identified item-sets depends on the following: the accuracy of meta-data used for prefiltering flows, the frequency of the prefiltered anomalous and normal flows and minimum support. The numbers of FP item-sets decreases with minimum support since less FP item-sets satisfy minimum support condition. Frequent item set are generated for minimum support values between 1000 and 4000 flows. If an anomaly happens to involve such a common feature value, the number of FP item-sets automatically increases even if no normal feature values are included in the meta-data. However, most of the FP item-sets can be sorted out rather easily by a network administrator.

An important question is which types of anomalies are captured with our item-set mining approach. There are two requirements for extracting an anomaly. The anomaly should: 1) be detected by causing a deviation in a traffic feature distribution and 2) trigger a large number of flows with similar characteristics. For many anomalies that originate from or are directed to a single or few IP addresses, these requirements are met. Scanning, flooding, and spamming activity, (distributed) denial-of service attacks, as well as related backscatter can be identified by frequent item-sets.

4.3 Computational Overhead

The computational cost for updating of histograms and for computing KL distance is a linear to the number of histogram bins. Frequent item-set mining is the most demanding step of our methodology both in terms of running time and memory overhead. Nevertheless, for all implementations, the computational overhead increases with the number of transactions and the number of frequent item-sets. Since both the number of transactions and the number of frequent item-sets increase as more normal flows are included in the input data set, the performance of Apriori and FP-Growth will decrease with the size of the input data set. The Advanced FP-Growth algorithm improves the scalability and efficiency of frequent item-set mining for dealing with big network data. As compared to other algorithms (Apriori and FP-growth algorithm) the advanced FP-growth reduces the runtime and the main memory consumption.

- Analysis and Comparison of different algorithms and their results

Figure 3 is the analysis that has been done on the basis of five features data sets of network packets, showing how time needed by the three different algorithms.

Data Set: Varying transactions set.

Figure 3 shows the comparative results of three algorithm w.r.t time for varying transaction sets. The graph is plotted for minimum support against time required by algorithm to run to completion. From figure 3 it is conclude that the time required by Advanced FP algorithm is comparatively low as compared to other two algorithms.

Time in seconds			
Support	Apriori	FP-Growth	Advanced FP
1000	23.082	9.789	9.047
2000	21.075	9.047	8.685
3000	17.362	8.486	7.767

Fig 3. Comparative results of three algorithm w.r.t. time

It also shows that as support increases the time required by algorithm is also decreases. Figure 4 is the analysis that has been done on five features data sets of network packets, showing how memory size is needed by the three different algorithms to generate frequents item sets.

Data Set: 10000 transactions set.

Time in seconds			
Support	Apriori	FP-Growth	Advanced FP
1000	1220	1096	240
1500	1220	1096	240
2000	620	436	136

Fig 4. Comparative results of three algorithm w.r.t. memory size

Figure 4 shows the comparative results of three algorithm w.r.t memory size for 10000 transaction sets. The graph is plotted minimum support against memory required by algorithm to run to completion. From figure 4 it conclude the size required by Advanced FP algorithm is comparatively low as compared to other two algorithms. It also shows that as support increases the memory size required by algorithm is also decreases.

5. CONCLUSION AND FUTURE SCOPE

The presented anomaly or inconsistency extraction method is homogeneous and can be used with different anomaly detectors that provide meta-data about identified anomalies. It is useful for finding the root cause of finding anomalies, which helps in network forensics, attack mitigation and anomaly modelling. The proposed methodology reduces the runtime and main memory consumption for dealing with large network traffic anomalous data and also reduces the large number of candidate set generations. A number of possible directions for future research exist. Optimizing the efficiency and scalability of frequent item-set mining for dealing with big network data including stream processing is one open problem.

6. ACKNOWLEDGMENTS

I am very thankful to my guide for guiding me and heartly thankful to IJCA to give me such a wonderful chance for publishing my paper.

7. REFERENCES

[1] D. Baruckhoff, X. Dimitropoulos, A. Wagner, and K. Salamatian, " Anomaly Extraction In Backbone Networks Using Association Rules" , *in proc.* IEEE ACM TRANSACTION ON NETWORKING, VOL 20. NO 6, DECEMBER 2012.

[2] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in *Proc. ACM SIGCOMM Internet Measurement Workshop*, Nov. 2002, pp. 71-82.

[3] Y. Zhang, Z. Ge, A. Greenberg, and M. Roughan, "Network anomography," in *Proc. ACM SIGCOMM Internet Measurement Conf.*, Oct. 2005.

[4] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *ACM SIGCOMM '05*, 2005, pp. 217-228.

[5] Y. Gu, A. McCallum, and D. Towsley, "Detecting anomalies in network traffic using maximum entropy estimation," in IMC'05: Proc. Internet Measurement Conf. 2005 Internet Measurement Conf.. Berkeley, CA, USA: USENIX Association, 2005, pp. 32-32.

[6] A. Kind, M. P. Stoecklin, and X. Dimitropoulos, "Histogram-basedtraffic anomaly detection," *IEEE Trans. Netw. Service Manage.*, vol. 6, no. 2, pp. 110–121, Jun. 2009.

[7] F. Silveira and C. Diot, "URCA: Pulling out anomalies by their root causes," in Proc. IEEE INFOCOM, Mar. 2010, pp. 1–9.

[8] S. Ranjan, S. Shah, A. Nucci, M. M. Munaf`o, R. L. Cruz, and S. M. Muthukrishnan, "Dowitcher: Effective worm detection and containment in the Internet core," in Proc. IEEE INFOCOM, 2007, pp.2541–2545.

[9] G. Cormode and S. Muthukrishnan, "What"s new: Finding significant differences in network data streams," IEEE/ACM Trans. Netw., vol. 13, no. 6, pp. 1219–1232, Dec. 2005.

[10] Ding Zhenguo, Wei Qinqin, Ding Xianhua "An Improved FP-growth Algorithm Based on Compound Single Linked List". In Proc. IEEE 2009.

[11] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proc. 20th VLDB*, Santiago de Chile, Chile, Sep. 12–15, 1994, pp. 487–499.

[12] Han,J.,Pei J. ,Yin,Y (1999).Mining Frequent Paterns Without Candidate Generation. Technical Report CMPT99-12, Schoolo f Computing Science, Simon Fraser University.

[13] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen, "Sketch-based change detection: Methods, evaluation, and applications," in Proc. 3rd ACMSIGCOMM IMC, 2003, pp. 234–247.

[14] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, and A. Lakhina, "Detection and identification of network anomalies using sketch subspaces," in Proc. 6th ACM SIGCOMM IMC, 2006, pp. 147–152 .