ISSN XXXX XXXX © 2018 IJESC

Research Article Volume 8 Issue No.4

An Efficient Predictive Network Anamoly Detection and Visualization

G.K.Prabhu[1], S. M. Jagatheesan
Research Scholar[1], Assistant Professor[2]
Department of Computer Science
Gobi Arts and Science College, Gobichettipalayam, India

Abstract:
Various approaches have been developed for quantifying and displaying network traffic information for determining network status and in detecting anomalies. Although many of these methods are effective, they rely on the collection of long-term network statistics. Here, we present an approach that uses short-term observations of network features and their respective time averaged entropies. Acute changes are localized in network feature space using adaptive Wiener filtering and auto-regressive moving average modeling. The color-enhanced datagram is designed to allow a network engineer to quickly capture and visually comprehend at a glance the statistical characteristics of a network anomaly. First, average entropy for each feature is calculated for every second of observation. Then, the resultant short-term measurement is subjected to first- and second-order time averaging statistics. These measurements are the basis of a novel approach to anomaly estimation based on the well-known Fisher linear discriminant (FLD). Average port, high port, server ports, and peered ports are some of the network features used for stochastic clustering and filtering. We empirically determine that these network features obey Gaussian-like distributions. The proposed algorithm is tested on real-time network traffic data from Ohio University's main Internet connection. Experimentation has shown that the presented FLD-based scheme is accurate in identifying anomalies in network feature space, in localizing anomalies in network traffic flow, and in helping network engineers to prevent potential hazards. Furthermore, its performance is highly effective in providing a colorized visualization chart to network analysts in the presence of bursty network traffic

I. INTRODUCTION

Intrusion detection devices are the primary way of protecting today's modern enterprise networks from a host of network anomalies such as viruses, worms, scanners, and denial of service (DoS) from botnets. Their defense mechanism relies on detection of attacks after they have begun affecting the targeted network. Existing methods are able to identify specific packets which match a known pattern or originate from a specified location. However, these signature-based systems fail to detect unknown anomalies. An anomaly might be an old attack that has changed in some way to avoid detection, or it could be a completely new form of attack. To alleviate these shortcomings, significant research has been devoted to the task of identifying network anomalies using methods from statistical signal analysis and pattern recognition theory.

A. ROUTING BASICS

This chapter introduces the underlying concepts widely used in routing protocols. Topics summarized here include routing protocol components and algorithms. In addition, the role of routing protocols is briefly contrast with the role of routed or network protocols. *Routing* is the act of moving information across an internetwork from a source to a destination. Along the way, at least one intermediate node typically is encountered. Routing is often contrasted with bridging, which might seem to accomplish precisely the same thing to the casual observer. The primary difference between the two is that bridging occurs at Layer 2 (the link layer) of the OSI reference model, whereas routing occurs at Layer 3 (the network layer). This distinction provides routing and bridging with different information to use in the process of moving information from source to destination, so the two functions accomplish their tasks in different ways. The topic of routing has been covered in computer science literature for more than two decades, but routing achieved commercial popularity as late as the mid-1980s. The primary reason for this time lag is that networks in the 1970s were simple, homogeneous environments.

B. ALGORITHM TYPES

Routing algorithms can be classified by type. Key differentiators include these:

- Static versus dynamic
- Single-path versus multipath
- Flat versus hierarchical
- Host-intelligent versus router-intelligent
- Intradomain versus interdomain
- Link-state versus distance vector

1) STATIC VERSUS DYNAMIC

Static routing algorithms are hardly algorithms at all, but are table mappings established by the network administrator before the beginning of routing. These mappings do not change unless the network administrator alters them. Algorithms that use static routes are simple to design and work well in environments where network traffic is relatively predictable and where network design is relatively simple. Because static routing systems cannot react to network changes, they generally are considered unsuitable for today's large, constantly changing networks. Most of the dominant routing algorithms today are dynamic routing algorithms, which adjust to changing network circumstances by analyzing incoming routing update messages. If the message indicates that a network change has occurred, the routing software recalculates routes and sends out new routing update messages. These messages permeate the network, stimulating routers to rerun their algorithms and change their routing tables accordingly. Dynamic routing algorithms can be supplemented with static routes where appropriate. A router of last resort (a router to which all unroutable packets are sent), for example, can be designated to act as a repository for all unroutable packets, ensuring that all messages are at least handled in some way.

2) SINGLE-PATH VERSUS MULTIPATH

Some sophisticated routing protocols support multiple paths to the same destination. Unlike single-path algorithms, these multipath algorithms permit traffic multiplexing over multiple lines. The advantages of multipath algorithms are obvious: They can provide substantially better throughput and reliability. This is generally called load sharing.

3) FLAT VERSUS HIERARCHICAL

Some routing algorithms operate in a flat space, while others use routing hierarchies. In a flat routing system, the routers are peers of all others. In a hierarchical routing system, some routers form what amounts to a routing backbone. Packets from no backbone routers travel to the backbone routers, where they are sent through the backbone until they reach the general area of the destination. At this point, they travel from the last backbone router through one or more no backbone routers to the final destination. Routing systems often designate logical groups of nodes, called domains, autonomous systems, or areas. In hierarchical systems, some routers in a domain can communicate with routers in other domains, while others can communicate only with routers within their domain. In very large networks, additional hierarchical levels may exist, with routers at the highest hierarchical level forming the routing backbone. The primary advantage of hierarchical routing is that it mimics the organization of most companies and therefore supports their traffic patterns well. Most network communication occurs within small company groups (domains). Because intradomain routers need to know only about other routers within their domain, their routing algorithms can be simplified and depending on the routing algorithm being used, routing update traffic can be reduced accordingly.

4) HOST-INTELLIGENT VERSUS ROUTER-INTELLIGENT

Some routing algorithms assume that the source end node will determine the entire route. This is usually referred to as source routing. In source-routing systems, routers merely act as store-and-forward devices, mindlessly sending the packet to the next stop. Other algorithms assume that hosts know nothing about routes. In these algorithms, routers determine the path through the inter-network based on their own calculations. In the first system, the hosts have the routing intelligence. In the latter system, routers have the routing intelligence.

5) INTRADOMAIN VERSUS INTERDOMAIN

Some routing algorithms work only within domains; others work within and between domains. The nature of these two algorithm types is different. It stands to reason, therefore, that an optimal intradomain-routing algorithm would not necessarily be an optimal interdomain-routing algorithm.

6) LINK-STATE VERSUS DISTANCE VECTOR

Link-state algorithms (also known as shortest path first algorithms) flood routing information to all nodes in the inter-network. Each router, however, sends only the portion of the routing table that describes the state of its own links. In link-state algorithms, each router builds a picture of the entire network in its routing tables. Distance vector algorithms (also known as Bellman-Ford algorithms) call for each router to send all or some portion of its routing table, but only to its neighbors. In essence, link-state algorithms send small updates everywhere, while distance vector algorithms send larger updates only to neighboring routers. Distance vector algorithms know only about their neighbors. Because they converge more quickly, link-state algorithms are somewhat less prone to routing loops than distance vector algorithms. On the other hand, link-state algorithms require more CPU power and memory than distance vector algorithms. Link-state algorithms, therefore, can be more expensive to implement and support. Link-state protocols are generally more scalable than distance vector protocols.

II. REVIEW OF LITERATURE

One of the most important tasks of data mining is to find patterns in data[7]. The best known data analysis technique in this area is association rule mining. An association rule is an expression of the form X => Y, where X and Y are sets of items. Given a database V consisting of transactions *T* (where each transaction is a set of items), a rule of the form X =>Y expresses that whenever a transaction *T* contains X, and then it is likely to contain *Y*. The degree of likeliness is expressed by the confidence of the rule which is defined as the number of transactions containing both *X* and *Y* divided by the total number of transactions containing *X*. That is, the rule confidence is understood as the conditional probability *p{Y* C *T\X* C T). Association rule mining originated from the analysis of market-basket data where rules like "A customer who buys milk and eggs will also buy bread with high probability" are found. The association rule mining problem can be formally stated as follows Let J = {'ii,'^2, • • • •*,'im]* be a set of literals, called items. Let P be a set of transactions, where each transaction T is a set of items such that *T C X*. associated with each transaction is a unique identifier, called its *TID*. We say that a transaction T *contains* X, a set of some items in X, if X C T. An *association rule* is an implication of the form X => Y, where X C X, *Y d* , and X D F == 0. The rule *X* => *Y* holds in the transaction set *V* with **confidence** c if c% of transactions in *V* that contain X also contain *Y*. The rule *X* =^*Y* has **support** s in Z> if s% of the transactions in *V* contain *XUY*. Rules are deemed interesting when they occur in many transactions (high support), and where transactions that contain the left hand side are likely to contain the right hand side (high confidence). The association rule mining problem is to find all such rules, i.e., all *A*, J5, and *C* that form rules with the desired confidence and support. The key component is to find sets of items that occur frequently, from these the rules can be determined. The canonical example of association rule mining is that of the original motivators - supermarkets. Any supermarket chain employs market-basket Analysis.

III. TYPES OF ALGORITHMS

There are different types of algorithms in association rules namely:

- Apriori algorithm
- Partition algorithm
- Pincher-search algorithm
- Dynamic item set counting algorithm
- FP-TREE growth algorithm.

C. APRIORI ALGORITHM

Apriori algorithm is the most popular algorithm to find all the frequent sets. It makes use of the downward closure property. Apriori algorithm is a bottom-up search, moving upward levelwise in the lattice. Before reading the database at every level it graciously prunes many of the sets which are unlikely to be frequent sets. The apriori frequent item set discovery algorithm uses the two functions namely candidate generation and pruning at every iteration. It moves upward in the lattice starting from level I till level k, where no candidate set remains after pruning. [6]

D. PARTITION ALGORITHM

The partition algorithm is based on the premise that the size of the global candidate set is considerably smaller than the set of all possible itemsets. As a result if we partition the set of transactions to smaller segments such that each segment can be accommodated in the main memory, then we can compute the set of frequent sets of each of these partitions. It is assumed that these sets contain reasonably a small number of itemsets.

E. PINCER-SEARCH ALGORITHM

The pincer-search computation algorithm starts from the smallest set of frequent itemsets and moves upward till it reaches the largest frequent itemset. The pincer –search algorithm is based on the principle of finding frequent item sets in a bottom-up manner but, at the same time, it maintains a list of maximal frequent item sets.

F. DYNAMIC ITEMSET COUNTING ALGORITHM

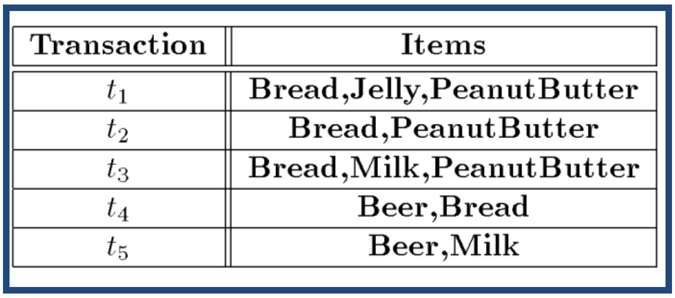
The rationale of dynamic set counting algorithm is that it works like a train pruning over the data, with stops at intervals M between transactions. When the train reaches the end of transaction file, it has made one pass over the data, and it starts all over again from the beginning for the next pass.

G. FP-TREE GROWTH ALGORITHM

The main idea of the algorithm is to maintain a Frequent Pattern Tree of the database. It is an extended prefix-tree structure, storing crucial and quantitative information about frequent sets. In this the tree nodes are the frequent items and are arranged in such a way that more frequently occurring nodes will have better chances of sharing nodes than the less frequently occurring ones. The method starts from frequent1-itemsets as an initial suffix pattern and examines only its conditional pattern base, which consists of the set of frequent items co-occurring with the suffix pattern.

H. FREQUENT ITEMSET MINING IN ASSOCIATION RULE

The items which are frequently purchased together are called as frequent item set mining. (Example: Bread=>Peanut Button.) Consider a set of items $I=\{I_1,I_2,\ldots,I_m\}$ a Transaction $D=\{t_1,t_2, \ldots, t_n\}$, $t_j \subseteq$ I and Item set: $\{I_{i1},I_{i2}, \ldots, I_{ik}\} \subseteq I$. In this the support of an item set is the percentage of transactions which contain that item set. The large frequent item sets is the number of occurrences in a threshold.



Transaction	Items
t_1	Bread,Jelly,PeanutButter
t_2	Bread,PeanutButter
t_3	Bread,Milk,PeanutButter
t_4	Beer,Bread
t_5	Beer,Milk

I = {Beer, Bread, Jelly, Milk, Peanut Butter}

Support of {Bread, Peanut Butter} is 60%

Association Rule (AR): implication $X \Rightarrow Y$ where $X,Y \subseteq I$ and $X \cap Y$;

Support of AR (s) $X \Rightarrow Y$: Percentage of transactions that contain $X \cup Y$

Confidence of AR (a) $X \Rightarrow Y$: Ratio of number of transactions that contain $X \cup Y$ to the number that contain X.

1) Association Rule Problem And Techniques In Frequent Itemset Mining

Given a set of items $I=\{I_1,I_2,\ldots,I_m\}$ and a database of transactions $D=\{t_1,t_2, \ldots, t_n\}$ where $t_i=\{I_{i1},I_{i2}, \ldots, I_{ik}\}$ and $I_{ij} \in I$, the Association Rule Problem is to identify all association rules $X \Rightarrow Y$ with a minimum support and confidence. Link Analysis Support of $X \Rightarrow Y$ is same as support of $X \cup Y$. There are two techniques in association rule mining with frequent item set namely.

1. To Find Large Item sets.

2. Generate rules from frequent item sets.

Large item set property: Any subset of a large item set is large.

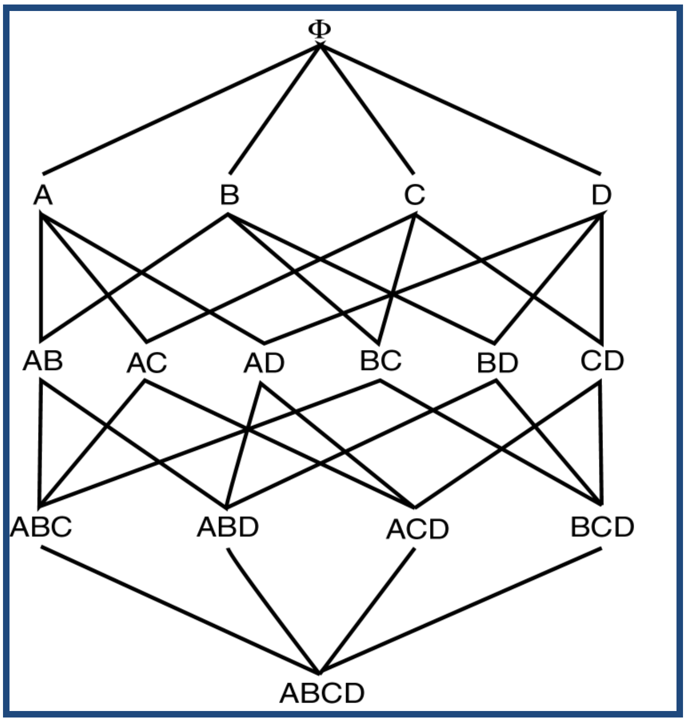


Figure.1. Large item set property

IV. BLOOMS TAXONOMY

Blooms taxonomy is founded by Benjamin Bloom (1913-1999). Blooms taxonomy categorizes the level of learning into six levels in the cognitive domain, namely-knowledge comprehension, application, analysis, synthesis and evaluation [2]. The levels are thought to build on one another to express quality of different kinds of thinking and to get acquainted to more actual teaching practices.

A. DOCUMENT TYPES WITH BLOOMS TAXONOMY

As this work is focused towards identifying the type of document automatically [8], table 1 shows the categorization of six types of document and their characteristics mapped with blooms taxonomy

Table.1.

Document Type	**Definition**	**Blooms level**	**Example**
Introduction	Documents which contain introduction of concepts	Knowledge	A document introducing a concept of a network or definition of a network
Explanation	Documents which contain explanation of concepts	Comprehension	A document describing or comparing different type of network topology
Application	Documents which give applications of any concept in practical situations	Application	A document that deals with application of network in e-mail, browsing , chatting etc.
Experiment	Documents which give experimental instructions and discussions	Analysis	A document that deals with configuration of network for a LAN
Compound	Documents that combine different concepts to form a whole	Synthesis	A document that deals about network and its hardware, software and peripherals
Exercise	Documents containing questions and exercises	Evaluation	Documents containing questions, numerical problems to find the data rate, bandwidth of a transmitted signal in a network

B. INTEGRATION WITH E-LEARNING

This approach specifies education and training impose challenges to e-learning systems to generate content according to the level of the learner. Identification of documents to particular level of this taxonomy enables e-learning systems to match learner needs. The evaluation of E-Learning is implemented by integrating the bloom's taxonomy using Association rule. The conversion of raw information into blooms model of learning and it is stored in database. Data mining can be used to extract knowledge from e-learning systems through the analysis of the information available in the form of data generated by their users.

V. CONCLUSIONS

Adoption of blooms taxonomy for document classification enables to effectively locate learning materials to the required level of the learner. The documents can be given to the users based on most viewed materials to individual users using association rule. This work is proposed for future research of storing information into the database automatically based on blooms taxonomy.

VI. REFERENCES

[1]. Agrawal.R, Mannila. H Srikanth. R., Toivinen H., and Verkamo A.I. Fast discovery of association rule. Advances in knowledge and data mining, AAAI/MIT press 1995.

[2]. Mannila, H. and Toivonen, H. 1997. Levelwise search and borders of theories in knowledge discovery. Data Mining and Knowledge Discovery 1, 3, 241{258. Oliveira, S. and Zaiane, O. 2002. Privacy preserving frequent itemset mining. In Proc. ICDM.

[3]. Anderson, L.W. & Krathwohl, D.R. (Eds.) (2001). A taxonomy for Learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives. New York: Addison Wesley Longman.

[4]. Bloom, B.S. (Ed.), Engelhart, M.D., Furst, E.J., Hill, W.H., & Krathwohl, D.R. (1956). Taxonomy of educational objectives: Handbook I: Cognitive domain. New York: David McKay.

[5]. Juan M.Ale, Gustavo H. Rossi, 2000. An Approach to Discovering Temporal Association Rules. In: 2000 ACM Symposium on Applied Computing (SAC'00). Como, Italy, pp.294~300

[6]. Sridhar Ramaswamy, Sameer Mahajan, Avi Silberschatz,1998.On the Discovery of Interesting Patterns in Association Rules.In: International Very Large Databases Conference, New York, America, pp.368~379

[7]. C. Romero, S. Ventura "Educational data mining: A survey from 1995 to 2005" Expert systems with Applications, Volume 33, Issue 1, July 2007, Pages 135–146.

[8]. A.Abdollahzadeh Barfourosh, H.R.Motahary Nezhad, M.L.Anderson, D.Perlis "Information Retrieval on the World Wide Web and Active Logic: A Survey and Problem Definition".