2012 International Conference on Communication, Information & Computing Technology (ICCICT), Oct. 19-20, Mumbai, India

Prediction of DoS attack Sequences

Alpa Reshamwala
Assistant Professor, Computer Engineering Department
MPSTME, SVKM's NMIMS University
Mumbai, India
alpa.reshamwala@nmims.edu

Dr. Sunita Mahajan
Principal, Institute of Computer Science
M.E.T, Bandra
Mumbai, India
sunitam_ics@met.edu

Abstract— **A denial of service attack (DOS) is any type of attack on a networking structure to disable a server from servicing its clients. Attacks range from sending millions of requests to a server in an attempt to slow it down, flooding a server with large packets of invalid data, to sending requests with an invalid or spoofed IP address. Sequential pattern mining is an important data mining problem with broad applications. Sequential Pattern Mining is to discover the frequent sequential pattern in the sequential event dataset. Intrusion detection using sequential pattern mining is a research focusing on the field of information security. In this paper, we have implemented Apriori a candidate generation algorithm and PrefixSpan a pattern growth algorithm on a network intrusion dataset from KDD Cup 1999, 10 percent of training dataset, which is the annual Data Mining and Knowledge Discovery competition organized by ACM Special Interest Group on Knowledge Discovery and Data Mining, the leading professional organization of data miners. To address the absence of timestamp in the dataset, we considered two approaches to generate the sequence database from the dataset. One is by taking service as reference attribute and the other one by taking a timestamp window of size one day (86400 seconds). We found that experimental results of PrefixSpan for predicting DoS attacks sequences on KDD cup 99 training dataset are efficient. These results are then compared with SPAM (Sequential Pattern Mining) algorithm which uses vertical bitmap data layout allowing for simple, efficient counting.**

Keywords- Data mining, fuzzy sets, sequence data, time interval, intrusion detection system, DoS attacks

I. INTRODUCTION

Data mining extracts implicit, previously unknown and potentially useful information from databases. The discovered information and knowledge are useful for various applications, including market analysis, decision support, music recommendation, fraud detection, intrusion detection and business management. Many approaches have been proposed to extract information, and mining sequential patterns is one of the most important ones [1][2][3].

Sequential Pattern Mining finds interesting sequential patterns among the large database. It finds out frequent subsequences as patterns from a sequence database. With massive amounts of data continuously being collected and stored, many industries are becoming interested in mining sequential patterns from their database. Sequential pattern mining is one of the most well-known methods and has broad applications including web-log analysis, customer purchase behavior analysis and medical record analysis. In the retailing business, sequential patterns can be mined from the transaction records of customers. For example, having bought a notebook, a customer comes back to buy a PDA and a WLAN card next time. The retailer can use such information for analyzing the behavior of the customers, to understand their interests, to satisfy their demands, and above all, to predict their needs. In the medical field, sequential patterns of symptoms and diseases exhibited by patients identify strong symptom/disease correlations that can be a valuable source of information for medical diagnosis and preventive medicine. In Web log analysis, the exploring behavior of a user can be extracted from member records or log files. For example, having viewed a web page on "Data Mining", user will return to explore "Business Intelligence" for new information next time. These sequential patterns yield huge benefits, when acted upon, increases customer royalty.

However, from these discovered sequential patterns, the time gaps between successive patterns cannot be determined. Accordingly, Chen *et al.* have proposed a generalization of sequential patterns, called time-interval sequential patterns, which reveals not only the order of patterns, but also the time intervals between successive patterns [4]. Two efficient algorithms, FTI-Apriori algorithm and the FTI-PrefixSpan algorithm, are developed by Chen *et al* for mining FTI sequential patterns [5]. In [6], we contributed to the ongoing research on FTI sequential pattern mining by proposing an algorithm to detect and classify audit sequential patterns in network traffic data. The paper also defines the confidence of the FTI audit sequences, which is not yet defined in the previous researches. In [7], we have proposed an algorithm which uses a fuzzy genetic approach to discover optimized sequences in the network traffic data to classify and detect intrusion.

It has been a great challenge to improve the efficiency of Apriori algorithm. Since all the frequent sequential patterns are included in the maximum frequent sequential patterns, the task of mining frequent sequential patterns can be converted as mining maximum frequent sequential patterns. PrefixSpan [8] is not a refinement of the apriori-like, candidate generation-and-test approach, rather a divide-and-conquer approach, called pattern-growth approach, which is an extension of FP-growth [9], an efficient pattern-growth algorithm for mining frequent patterns without candidate generation. SPAM [10] algorithm is based on the lexicographic sequence tree and can make either depth or width first traversal.

978-1-4577-2078-9/12/$26.00©2011 IEEE

In this paper, all these algorithms are implemented to mine frequent sequential patterns on KDD Cup 1999 dataset to predict DoS attack sequences on network traffic data.

II. Related Work

The problem of mining sequential patterns was first introduced by Agarwal and Srikant [1] which discovers patterns that occur frequently in a sequence database. A sequence database is formed by a set of data sequences. Each data sequence includes a series of transactions, ordered by transaction times. After mid 1990's, following Agrawal and Srikant [1], many scholar provided more efficient algorithms [8][11][12][13]. Besides these, works have been done to extend the mining of sequential patterns to other time-related patterns. Existing approaches to find appropriate sequential patterns in time related data are mainly classified into two approaches. In the first approach developed by Agarwal and Srikant [14], the algorithm extends the well-known Apriori algorithm. This type of algorithms is based on the characteristic of Apriori—that any subpattern of a frequent pattern is also frequent [1]. The later, uses a pattern growth approach [8], employs the same idea used by the Prefix-Span algorithm.

This algorithm divides the original database into smaller subdatabases and solve them recursively. Previous research addresses time intervals in two typical ways, first by the time-window approach, and second by completely ignoring the time interval. First, the time window approach requires the length of the time window to be specified in advance. A sequential pattern mined from the database is thus a sequence of windows, each of which includes a set of patterns. Patterns in the same time window are bought in the same time period. In the algorithm [12], Shrikant and Agrawal, specified the maximum interval (max-interval), the minimum interval (min-interval) and the sliding time window size (windowsize). Moreover, they cannot find a pattern whose interval between any two sequences is not in the range of the window-size. Agrawal and Srikant [1], introduced mining traditional sequential mining, by ignoring the time interval and including only the temporal order of the patterns.

To address the intervals between successive patterns in sequence database, Chen *et al.* have proposed a generalization of sequential patterns, called time-interval sequential patterns, which reveals not only the order of patterns, but also the time intervals between successive patterns [4]. Chen *et al.* developed algorithms to find sequential patterns using both the approaches [4]. Their work, by assuming the partition of time interval as fixed, developed two efficient algorithms -I-Apriori and I- PrefixSpan. The first algorithm is based on the conventional Apriori algorithm, while the second one is based on the PrefixSpan algorithm. An extension of the algorithm developed by Chen *et al* [4], to solve the problem of sharp boundaries to provide a smooth transition between members and non-members of a set, is addressed in Chen *et al* [5]. The sharp boundary problems can be solved by the concept of fuzzy sets. Two efficient algorithms, the FTI-Apriori algorithm and the FTI-PrefixSpan algorithm, were developed for mining FTI sequential patterns. There are several other reasons that support the use of FTI in place of crisp interval. First, the human knowledge can be easily represented by fuzzy logic. Second, it is widely recognized that many real world situations are intrinsically fuzzy, and the partition of time interval is one of them. Third, FTI is simple and easy for users.

Anrong *et al* [15], addresses application of sequential pattern in intrusion detection by refining the pattern rules and reducing redundant rules. Their work implements PrefixSpan algorithm in the data mining module of network intrusion detection system (NIDS). Fuzzy logic addresses the formal principles of approximate reasoning. It provides a sound foundation to handle imprecision and vagueness as well as mature inference mechanisms by varying degrees of truth. As boundaries are not always clearly defined, fuzzy logic can be used to identify complex pattern or behavior variations. And it can be accomplished by building an intrusion detection system that combines fuzzy logic rules with an expert system in charge of evaluating rule truthfulness. In [6], we contributed to the ongoing research on FTI sequential pattern mining by proposing an algorithm to detect and classify audit sequential patterns in network traffic data. The paper also defines the confidence of the FTI audit sequences, which is not yet defined in the previous researches. In [7], we have proposed an algorithm which uses a fuzzy genetic approach to discover optimized sequences in the network traffic data to classify and detect intrusion.

III. Sequence Database Generation

Data acquired from the dataset may be not sequential. A sequence is an ordered list of items [1]. Now, consider KDD cup 99 training dataset which is approximately 4,900,000 single connection vectors, each of which contains 41 features and is labelled as either normal or an attack, with exactly one specific attack type. For example, a network connection can be uniquely identified by the combination of its *timestamp* (start time), *src host* (source host), *src port* (source port), *dst host* (destination host), and *service* (destination port). These are the essential attributes when describing network data [16]. To address the absence of timestamp in the dataset, for generating a sequence database we have considered two approaches namely, one by taking service as reference attribute and the other by taking a timestamp window of size 1 day (86400 seconds). These approaches are applied on the dataset of different classified attacks to predict the sequence of attacks. For experimental purpose, we consider DoS attacks, which are approximately 392000 single connection vectors each labelled as either normal or a DoS attack. From these, we can find the sequence of DoS attacks such as Pod is followed by Teardrop, where pod and teardrop are the two types of Dos attacks. Table I shows the sample of DoS attack of KDD cup 99 training dataset with service (ex: telnet, ftp), protocol type (ex: tcp, udp) and the classified attack class as the significant attributes to detect intrusion. The format of the sequences will be ($<t_i>$ (a_1 a_2 . . . a_n)). Now by applying the first approach, with service as the reference attribute on which attack sequences are generated is as shown in Table II.

TABLE I. DOS ATTACK TCPDUMP OF KDD CUP 99 TRAINING DATASET

SequenceID	Day	Protocol_type	Service_ID	Service	Attack_ID	Class
82871	2	Icmp	13	ecr_i	8	pod.
82872	2	Icmp	13	ecr_i	8	pod.
82873	2	Icmp	13	ecr_i	8	pod.
86545	2	Udp	39	private	10	teardrop.
86546	2	Udp	39	private	10	teardrop.
86547	2	Udp	39	private	10	teardrop.
143311	3	Icmp	13	ecr_i	8	pod.
143312	3	Icmp	13	ecr_i	8	pod.
149209	3	Udp	39	private	10	teardrop.
149210	3	Udp	39	private	10	teardrop.

TABLE II. SEQUENCE DOS ATTACK OF KDD CUP 99 APPROACH 1

Service_ID	Attack Sequence
13	(<82871> (8) <143311> (8))
39	(<86545> (10) <149209> (10))

When taking the second approach, a timestamp window of 1 day, we get the timestamp sequence database as shown in Table III. Here the sequence is divided by a timestamp window of 1 day or 86400 seconds.

TABLE III. SEQUENCE DOS ATTACK OF KDD CUP 99 APPROACH 2

Day	Attack Sequence
2	(<82871>(8) <86545> 10)
3	(< 143311>(8) <149209> 10)

IV. ALGORITHMS

Prediction of DoS attack sequences on KDD Cup 1999 dataset using, the three algorithms: AprioriAll, SPAM, PrefixSpan. All of which find sequential patterns without time intervals using traditional approach.

AprioriAll[1] is based on Apriori algorithm, in each pass we use the large sequences from the previous pass to generate the candidate sequences and then measure their support by making a pass over the database. SPAM [10] algorithm is based on the lexicographic sequence tree and can make either depth or width first traversal. An additional advantage is its property of online outputting sequential patterns of different length compared to a breadth first search strategy that outputs al patterns of length one, then all patterns of length two and so on. SPAM also uses vertical bitmap data layout allowing for simple, efficient counting. PrefixSpan is not a refinement of the apriori-like, candidate generation-and-test approach, rather a divide-and-conquer approach, called pattern-growth approach, which is an extension of FP-growth [9], an efficient pattern-growth algorithm for mining frequent patterns without candidate generation. PrefixSpan recursively projects a sequence database into a set of smaller projected sequence databases and grows sequential patterns in each projected database by exploring only locally frequent fragments. It mines the complete set of sequential patterns and substantially reduces the efforts of candidate subsequence generation. A pseudo projection technique is used for PrefixSpan to reduce the number of physical projected databases to be generated.

V. RESULTS AND DISCUSSION

In this section we perform a simulation study to compare the performances of the algorithms: the Apriori [1], PrefixSpan [8] and SPAM [10], which find sequential patterns without time intervals.

These algorithms were implemented in Sun Java language and tested on an Intel Core Duo Processor, 2.10 GHz with 2GB main memory under Windows XP operating system.

The first comparison is based on the KDD cup 99 dataset (Approach 1 and Approach 2) where the minimum support threshold is varied 10 % to 60%. Figure 1 and 3 summarizes those results. All the results show that SPAM is fastest followed by PrefixSpan and then Apriori. SPAM performs well for large datasets is due to the bitmap representation of the data for efficient counting [10].

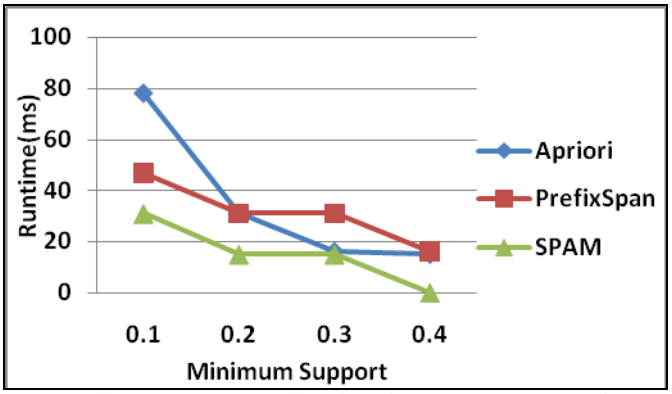

Figure 1. Execution Times for KDD cup 99 Approach 1

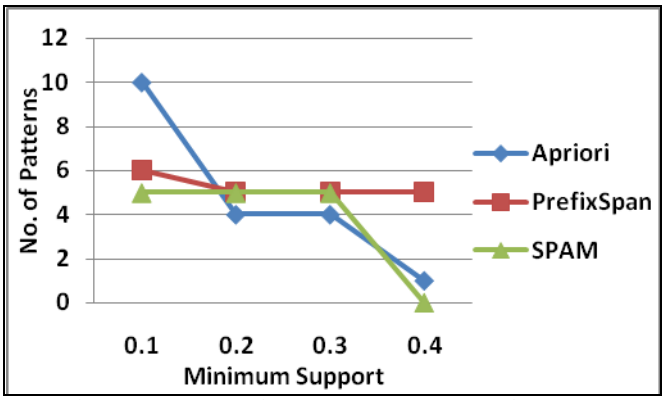

Figure 2. No. of patterns verses pattern type for KDD cup 99 Approach 1

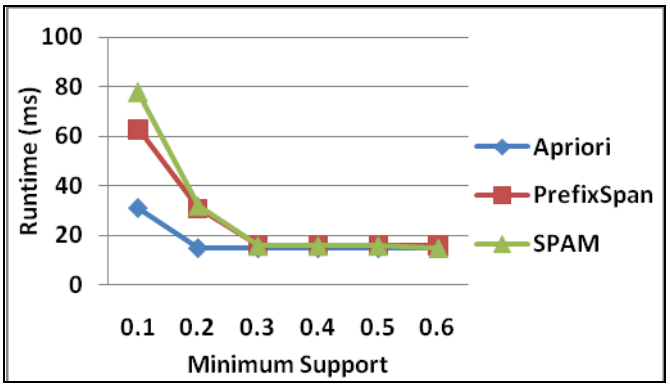

Figure 3. Execution Times for KDD cup 99 Approach 2

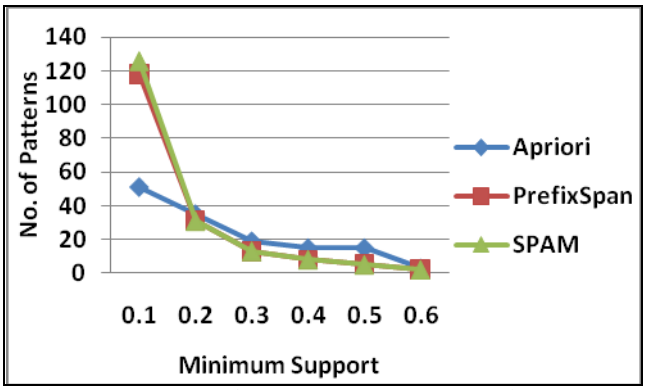

Figure 4. No. of patterns verses pattern type for KDD cup 99 Approach 2

The second comparison is done on the number of frequent sequence patterns found executing these algorithms with the varying minimum support threshold. From the results in Figure 2 and 4, it is shown that PrefixSpan gives the efficient number of sequential patterns as it examines only the prefix subsequences and project only their corresponding postfix subsequences into projected database and its pseudo-projection has the best performance[8]. Apriori gives the maximum number of all sequential patterns due to its property of counting non-maximal sequences.

As shown in Figure 4, SPAM and PrefixSpan results more than 100 efficient frequent sequential patterns. Divide and conquer method of PrefixSpan and vertical bitmap data layout of SPAM allowing for simple, efficient counting makes Approach 2 results to be efficient. Also, from the data in Table II and III, we can say that, sequences in Table III are efficient as Pod is followed by Teardrop, where pod and teardrop are the two types of Dos attacks.

VI. Conclusions and Future Work

In this paper we have implemented Apriori a candidate generation algorithm and PrefixSpan a pattern growth algorithm on KDD Cup 1999 dataset to predict DoS attack sequences on network traffic data. Experimental results have shown that PrefixSpan has generated efficient sequential patterns. These results are then compared with SPAM (Sequential Pattern Mining) algorithm. The results in Figure 1, also shows that SPAM performs well for large datasets like KDD Cup 1999 dataset is due to the bitmap representation of the data for efficient counting[10].

The second comparison is done on the number of frequent sequence patterns found executing these algorithms with the varying minimum support threshold and from the results, it is shown that the PrefixSpan gives the least number of efficient sequential patterns by recursively generating pseudo projected databases[8].

Also from Figure 4, SPAM and PrefixSpan results more than 100 efficient frequent sequential patterns. Divide and conquer method of PrefixSpan and vertical bitmap data layout of SPAM allowing for simple, efficient counting. From the data in Table II and III, we can conclude that, sequences in Table III are efficient as Pod is followed by Teardrop, where pod and teardrop are the two types of Dos attacks. Hence we can conclude that, Approach 2 results are more efficient.

In future work, as in these experiments we have found sequence patterns, by ignoring the time interval and including only the temporal order of the patterns. To address the intervals between successive patterns in sequence database, Chen *et al.* have proposed a generalization of sequential patterns, called time-interval sequential patterns, which reveals not only the order of patterns, but also the time intervals between successive patterns [4]. An extension of the algorithm developed by Chen *et al* [4], to solve the problem of sharp boundaries to provide a smooth transition between members and non-members of a set, is addressed in Chen *et al* [5] can also be implemented. Also as proposed in [7], the use of fuzzy genetic approach to discover optimized sequences in the network traffic data to classify and detect intrusion can also be implemented.

References

[1] R. Agrawal and R. Srikant, "Mining sequential patterns", In Proc. Int. Conf. Data Engineering, 1995, pp. 3–14.

[2] Y. L. Chen, S. S. Chen, and P. Y. Hsu, "Mining hybrid sequential patterns and sequential rules", Inf. Syst., vol. 27, no. 5, pp. 345–362, 2002.

[3] J. Han and M. Kamber, Data Mining: Concepts and Techniques, New York: Academic, 2001.

[4] Y. L. Chen, M. C. Chiang, and M. T. Ko, "Discovering time-interval sequential patterns in sequence databases", Expert Systems with Applications, Volume 25, Issue 3, October 2003, pp 343−354.

[5] Yen-Liang, Tony Cheng-Kui Huang, "Discovering Fuzzy Time-Interval Sequential Patterns in Sequence Databases", IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 2005, vol.35, pp.959-972.

[6] Sunita Mahajan and Alpa Reshamwala, "Amalgamation of IDS Classification with Fuzzy techniques for Sequential pattern mining ",IJCA Proceedings on International Conference on Technology Systems and Management - ICTSM 2011, Number 3 - Article 7, pp 9–14.

[7] Sunita Mahajan and Alpa Reshamwala, "An Approach to Optimize Fuzzy Time-Interval Sequential Patterns Using Multi-objective Genetic Algorithm", ICTSM 2011, CCIS 145, pp. 115–120, 2011, Springer-Verlag Berlin Heidelberg 2011.

[8] Pei, J., Han, J., Pinto, H., Chen, Q., Dayal, U., & Hsu, M.-C., "PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth", Proceedings of 2001 International Conference on Data Engineering, pp. 215–224.

[9] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," Proc. 2000 ACM-SIGMOD Int'l Conf. Management of Data (SIGMOD '00), pp. 1-12, May 2000.

[10] J. Ayres, J. Gehrke, T. Yiu, and J. Flannick, "Sequential PAttern Mining using A Bitmap Representation", In Proceedings of ACM SIGKDD on Knowledge discovery and data mining, pp. 429-435, 2002.

[11] Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., & Hsu, M.-C., "FreeSpan: Frequent pattern-projected sequential pattern mining", Proceedings of 2000 International Conference on Knowledge Discovery and Data Mining, pp. 355–359.

[12] Srikant, R., & Agrawal, R., "Mining sequential patterns: Generalizations and performance improvements", Proceedings of the 5th International Conference on Extending Database Technology,1996, pp. 3–17.

[13] Zaki, M. J., "SPADE: An efficient algorithm for mining frequent sequences", volume 42 Issue 1-2, January-February 2001, pp 31–60.

[14] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules", Proceedings of 20th VLDB Conference Santiago, Chile, 1994, pp. 487–499.

[15] XUE Anrong, HONG Shijie, JU Shiguang, CHEN Weihe, "Application of Sequential Patterns Based on User's Interest in Intrusion Detection", Proceedings of 2008 IEEE International Symposium on IT in Medicine and Education, pp 1089- 1093, 2008.

[16] Lee W and Stolfo S J, "Data mining approaches for intrusion detection", Proceedings of the 7th USENIX Security Symposium, :26-29, 1998.