Amalgamation of IDS Classification with Fuzzy Techniques for Sequential Pattern Mining

Dr. Sunita Mahajan
Principal
Institute of Computer Science
M.E.T, Bandra, Mumbai

Alpa Reshamwala
Research Scholar, Assistant Professor
Computer Department, MPSTME
SVKM's NMIMS University, Mumbai

ABSTRACT
Intrusion detection system has been a powerful weapon to protect networks from attacks and has gained more and more attention. Data mining has been proven as an important method to detect intrusions. Fuzzy logic based methods together with the techniques from Artificial Intelligence have gained importance. Sequential pattern mining, which discovers frequent subsequences as patterns in a sequence database, is useful in discovering audit patterns along with time from network audit databases.
Intrusion detection system uses Boolean logic in determining whether or not an intrusion is detected and the use of fuzzy logic has been investigated as an alternative. 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 using varying degrees of truth. Because boundaries are not always clearly defined, fuzzy logic can be used to identify complex pattern or behavior variations. Accordingly, Chen *et al.* have proposed a fuzzy time-interval (FTI) sequential pattern mining algorithms, which reveals the time intervals between successive patterns [12][13]. In this paper, 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 defines the confidence of the FTI audit sequences, which is not yet defined in the previous researches.

Keywords
Data mining, fuzzy sets, sequence data, time interval, intrusion detection system.

1. INTRODUCTION
The security of computer network plays a strategic role in modern computer systems with the widespread use of network. Intrusion Detection Systems (IDS) are effective security tools, placing inside a protected network and looking for known or potential threats in network traffic and/or audit data recorded by hosts. Basically, IDS analyzes information about user's behaviors from various sources such as audit trail, system table, and network usage data. The problem of intrusion detection has been studied extensively in computer security [1][2][3], and has received a lot of attention in machine learning and data mining [4].Intrusion detection techniques can be categorized into misuse detection, which uses Patterns of well-known attacks or weak spots of the system to identify intrusions, misuse detection system, for example, IDIOT [5] and STAT [6], use patterns of well-known attacks or weak spots of the system to match and identify intrusion; and anomaly detection, which tries to determine whether deviation from the established normal usage patterns can be flagged as intrusion. Anomaly detection system, for example, IDES [7], flags observed activities that deviates significantly from the established normal (statistical based) usage profiles as anomalies (i.e., possible intrusions). Lee and Stolfo [8] discuss data mining approaches for intrusion detection. With the improvement of intrusion detection means, sometime it is difficult to judge whether an isolated sequence of audit event belongs to intrusion or not, but if a series of events are ranged in time sequencing, we may find anomalies. So the sequence pattern algorithms of data mining techniques are applied in intrusion detection and intrusion pattern rules are found by learning the frequent episodes. A frequent episode is a set of events that occur frequently within a time window. Sequential pattern mining algorithms can help us understand what (time-based) sequence of audit events are frequently encountered together. These frequent event patterns are important elements of the behavior profile of a user or program.
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, 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 [9][10][11]. An example of such a pattern in the network traffic can be, "IP sweep attack", then "ARP poisoning attack", and then "Session hijacking attack". Note that these network patterns are found to be consecutive. Once network probing signatures are detected, the above example pattern can follow sequentially. An IP Sweep attack occurs when an attacker sends ICMP echo requests (pings) to multiple destination addresses. If a target host replies, the reply reveals the target's IP address to the attacker. Address Resolution Protocol (ARP) spoofing, also known as ARP flooding, ARP poisoning or ARP Poison Routing , is a technique used to attack an Ethernet wired or wireless network. The principle of ARP spoofing is to send fake, or "spoofed", ARP messages to an Ethernet LAN. Generally, the aim is to associate the attacker's MAC address with the IP address of another node which can be default gateway. Any traffic meant for that IP address would be mistakenly sent to the attacker instead. The attacker could then choose to forward the traffic to the actual default gateway, called as passive sniffing or modify the data before forwarding it called as man-in-the-middle attack. The attacker could also launch a denial-of-service attack against a victim by associating a nonexistent MAC address to the IP address of the victim's default gateway. Session hijacking is the exploitation of a valid computer session sometimes also called a session key to gain unauthorized access to information or services in a computer system.
From the discovered sequential patterns, we can know what patterns are frequently brought together and in what order they appear. However, they cannot tell us the time gaps between successive patterns. 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 [12]. An example of time-interval sequential pattern has a form like (A, I_2, B, I_1, C), meaning that pattern B is followed by pattern A and pattern C is followed by pattern B with the predetermined time interval of I_2 and I_1 respectively. Although sequential patterns extended with time intervals can offer more information than those without time intervals, the approach may cause the sharp boundary problem. That is, when a time interval is near the boundary of two adjacent ranges, we either ignore or overemphasize it. For example, let the interval of I_1 be $5 \leq t < 10$ and that of I_2 be $10 \leq t < 20$, where t is the time gap between two successive patterns. Then if the time gap between patterns A and B is near 10, either a little larger or smaller, it is difficult to say the time gap is in I_1 or in I_2. However, according to the original definition of Chen *et al.* it can only be one hundred percent in I_1 or in I_2. This difficulty can be adequately tackled by using fuzzy techniques, for fuzzy set theory allows this time gap to be 50% in I_1 and at the same time 50% in I_2. This simple example indicates that the fuzzy sets provide a smooth transition between member and nonmember of a set. Two efficient algorithms, FTI-Apriori algorithm and the FTI-PrefixSpan algorithm, are developed by Chen *et al* for mining FTI sequential patterns [13].

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. Mining maximum frequent sequential patterns is more important for data mining FTI sequential patterns.

In this paper, an algorithm is proposed to mine frequent sequential network audit patterns using FTI sequential pattern to classify and detect intrusion.

2. 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[15][16][17][18]. 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 [15], 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 [17], 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 [12]. Chen *et al.* developed algorithms to find sequential patterns using both the approaches [12]. 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* [12], 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* [13]. 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* [20], 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.

This paper focuses on mining audit sequences, using FTI sequential pattern in network audit data to classify and detect intrusion. An algorithm is proposed to mine frequent sequential network audit patterns using FTI sequential pattern to detect high accuracy for intrusions and optimize the network intrusion classifier.

3. THEORY

In all the definitions for **n** audit patterns in **S** audit sequences with **sid** as the audit sequence-id in a network traffic pattern **T** is represented as <**sid**, **S**>.

3.1 Audit Sequence Pattern

An audit pattern-set is a non-empty set of audit patterns. An audit sequence is an ordered list of audit pattern-set. Without loss of generality, we assume that the set of audit patterns is mapped to a set of contiguous integers. We denote an audit pattern-set **a** as $(a_1a_2\ldots a_n)$, where a_j is an audit pattern. We denote an audit sequence **S** by $< s_1s_2\ldots s_n>$, where s_j is an audit pattern-set.

An audit sequence $< a_1a_2\ldots a_n >$ is contained in another audit sequence $< b_1b_2\ldots b_m >$ if there exist integers $i_1 < i_2 < \ldots < i_n$ such that $a_1 \subseteq bi_1$, $a_2 \subseteq bi_2$, …, $a_n \subseteq bi_n$. For example, the audit sequence < (3) (4 5) (8) > is contained in < (7) (3 8) (9) (4 5 6) (8) >, since (3) $\subseteq$ (3 8), (4 5) $\subseteq$ (4 5 6) and (8) $\subseteq$ (8). However, the audit sequence <(3) (5) > is not contained in < (3 5) > (and vice versa). The former represents audit patterns 3 and 5 occurred one after the other, while the latter represents audit pattern 3 and 5 occurred together.

3.2 Time interval Audit Sequence Pattern

An audit sequence **ST** is represented as $((a_1,t_1),(a_2,t_2),(a_3,t_3),\ldots,(a_n,t_n))$, where a_j is an audit pattern and t_j stands for the time at which a_j occurs, $1 \le j \le n$, and $t_{j-1} \le t_j$ for $2 \le j \le n$. In the audit sequence, if audit patterns occur at the same time, they are ordered alphabetically. The time interval values can be calculated as $ti_j = | t_{j+1} - t_j |$, where j = 1,2,…,n-1.

For example, in an audit sequence S, ((a, 4), (d, 10), (e.28)) the time interval values are 6 and 18. Here a, d and e are the audit patterns of the network traffic pattern.

3.3 Fuzzy Logic

Fuzzy logic has been a powerful tool for decision making to handle imprecise and uncertain data. In contrast to classical set, a fuzzy set is a set without crisp boundaries; the transition from "belong to a set" to "not belong to a set" is gradual. Membership function is utilized to reflect a degree of membership and indicated by a value in the range [0.0, 1.0].

A Fuzzy set can be defined as - If *U* is a collection of objects denoted generically by *x*, then a *fuzzy set A* in *U* is defined as a set of ordered pairs:

$A= \{(x, \mu_A(x))| x \in U\}$,

where, $\mu_A(x)$ is the membership function and *U* is the universe of discourse.

For example, *A one year old baby will clearly be a* member of the set, and a 100 years old person will not be a member of this set, but what about people at the age of 20, 30, or 40 years? U = {0, 20, 40, 60, 80},

A = { (0,1),(20,0.6),(40,0.1),(60,0),(80,0)}. Alternative representation can be A = 1/0 + 0.6/20 + 0.1/40 + 0/60 + 0/80. Figure 1 shows the graphical representation.

Just like an algebraic variable takes numbers as values, a Linguistic/ Fuzzy Variables *takes words* or sentences as values. The set of values that it can take is called its "*term set". Each value in the term set is a "Fuzzy variable" defined over a "Base variable".* The base variable defines the universe of discourse for all the fuzzy variables in the term set. The fuzzy variables themselves are adjectives that modify the variable (e.g. "large positive" error, "small positive" error ,"zero" error, "small negative" error, and "large negative" error). As a minimum, one could simply have "positive", "zero", and "negative" variables for each of the parameters. Additional ranges such as "very large" and "very small" could also be added to extend the responsiveness to exceptional. For example, Let x be a Linguistic/ Fuzzy Variable with the label "Age",

T = { Old, VeryOld,NotSoOld,MoreOrLessYoung, QuiteYoung, VeryYoung} or can be T = {Young, Middle Aged, Old} as depicted in the Figure 2.

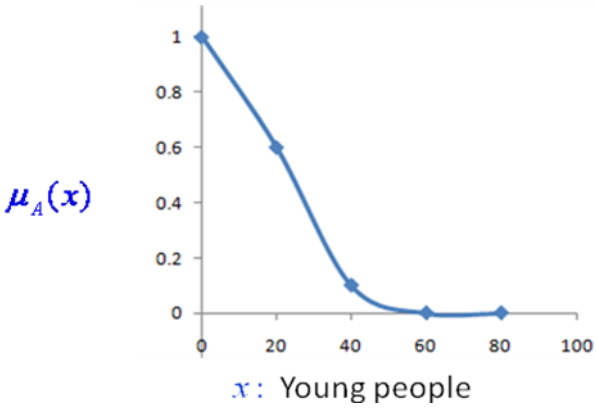

Figure 1. Graphical representation of a Fuzzy set.

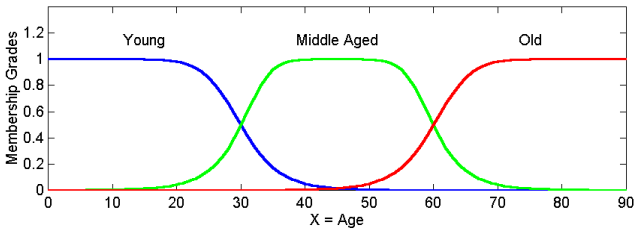

Figure 2. Membership Function for a fuzzy set.

3.4 Fuzzy Time Interval Audit Sequence Pattern

Two approaches have been used to determine linguistic terms and fuzzy membership functions. The first approach relies on domain experts to specify the functions based on their background knowledge and requirements. The second approach assumes that the functions are obtained by a preprocessing phase that learns the functions from the data, such as learning by neural-network, by GA, by clustering method, and by entropy measure. Since, in the current fuzzy mining researches, the first approach is more popular than the second one, we also adopt the same assumption that the fuzzy membership functions are as shown in Figure 3.

Suppose we want to represent a time interval by using three linguistic terms: *Short* (*S*), *Medium* (*M*), and *Long* (*L*) within a month. Consider a month having 30 days, *Short* linguistic variable can be defined as, if the time interval between first two events is less than or equal to two days then they are definitely in sequence and has the membership value as 1and if time interval is within 15 days then membership values is calculated according to the slope of the line between 2 and 15. If the time interval is more than 15 then definitely it cannot be represented by variable *Short* and hence the membership value is 0. Similarly, we can define the membership value for linguistic variable *Medium* and *Long*. Their membership functions can be represented by equations (1), (2) and (3) respectively [13].

Let $\boldsymbol{A}=\{a_1a_2\ldots a_n\}$, be the set of audit patterns and LT = { $lt_j | j= 1,2,\ldots,l\}$ be a set of all linguistic terms. An audit sequence $\alpha = (b_1, lg_1, b_2, lg_2,\ldots, b_{r-1}, lg_{r-1}, b_r)$ is a FTI audit sequence if $b_i \in \boldsymbol{A}$ and $lg_i \in$ LT for $1 \le i \le r-1$ and $b_r \in \boldsymbol{A}$.

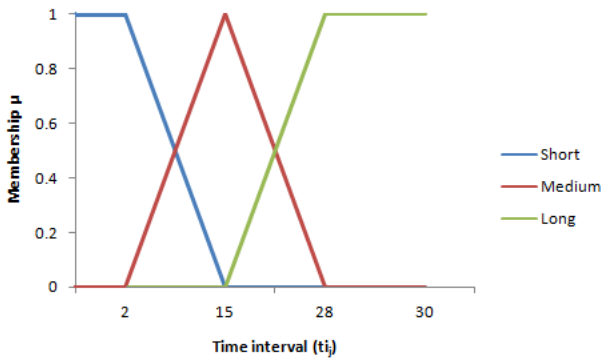

Figure 3. Membership function for time interval

$$\mu_{short}(ti_j) = = \begin{cases} 1 & ; tij \le 2 \\ \frac{15-tij}{13}; & 2 < tij < 15 \\ 0 & ; tij \ge 15 \end{cases} \quad \textbf{(1)}$$

$$\mu_{Medium}(ti_j) = = \begin{cases} 0; & 2 \ge tij \; or \; tij \ge 28 \\ \frac{tij-2}{13}; & 2 < tij \le 15 \\ \frac{28-tji}{13}; & 15 < tij < 28 \end{cases} \quad \textbf{(2)}$$

$$\mu_{Long}(ti_j) = = \begin{cases} 0 & ; tij \le 15 \\ \frac{tij-15}{13}; & 15 \le tij < 28 \\ 1 & ; tij \ge 28 \end{cases} \quad \textbf{(3)}$$

If an audit sequence α is contained in with degree γ, then we call α a FTI audit subsequence of **S** with degree γ. The total number of audit patterns in a FTI audit sequence α is referred to as the *length* of the audit sequence. A FTI audit sequence whose length is *k* is referred to as a fuzzy *k*-time-interval audit sequence.

$$\text{Support}_S(\alpha) = \sum_{(sid,s)in\,S} \frac{\gamma(\alpha,s)}{|S|} \quad \textbf{(4)}$$

Support of a FTI audit sequence α is given by equation (4). Given an audit sequence database and *min_sup*, the goal of FTI audit sequential pattern mining is to determine in the audit sequence database all the FTI audit subsequences whose supports are more than or equal to *min_sup*.

$\text{support\_count}_S(\alpha) = \{(sid,s) \mid (sid,s) \in S \square\ \alpha \text{ is contained in } s\}$ **(5)**

$$\text{Confidence}_S(\alpha_1 => \alpha_2) = \frac{\text{support\_count}(\alpha 1 \cup \alpha 2)}{\text{support\_count}(\alpha 1)} \quad \textbf{(6)}$$

Confidence of FTI audit sequence α_1 on α_2 is given by equation (6) uses the *support_count* defined in equation (5). For example, consider the audit sequence database in Table 1, with the linguistic terms defined: *Short* (*S*), *Medium* (*M*), and *Long* (*L*). For 1-sequence, L_1 is shown in Figure 4 and assume that we set *min_sup* as 0.3. Then we have L_1 = {a, b, e} as their supports are larger than *min_sup*. C_2 can be generated by joining L_1 with LT= {short, medium, long}, membership function of these are given by equation (1),(2) and (3) respectively. Here confidence of the sequences is calculated according to the equation (5). Hence we get, Confidence (a => b) = 0.625, Confidence (b => e) = 0.714 and Confidence (a => e) = 0.625. Now for 2-sequence, among the above three, only the pattern of "a then b", "b then e" and "a then e" can be generated as their confidence is larger than 0.3. Consider for example, support for "a then b" for different linguistic term in LT, using equation (4) from sid: 10,50,60,90,100.

$$\text{Support (a short b)} = \frac{(\mu\text{short (3)} + \mu\text{short (1)} + \mu\text{short (5)} + \mu\text{short (11)} + \mu\text{short (3)})}{|S|}$$

We get,

$$\text{Support (a short b)} = \frac{(0.92+1.0+0.77+0.31+0.92)}{10} = 0.392$$

$$\text{Support (a medium b)} = \frac{(0.08+0.0+0.23+0.69+0.08)}{10} = 0.108$$

$$\text{Support (a long b)} = \frac{(0.0+0.0+0.0+0.0+0.0)}{10} = 0.0$$

Similarly we get Supp (a *short* b) = 0.392, Supp (a long e) = 0.384 and Supp (b long e) = 0.408, which are larger then *min_sup*. Accordingly, the confidence ((a short b) => e) = 0.8.

After the generation of L_2, the algorithm starts to produce C_k and L_k for k>2 . Since the patterns in $L_2$are shown in Figure 4, C_3 is < a short b long e>. Support for audit sequences in C_3 can be given by using equation (4) from sid: 10,50,60,100.as,

$$\text{Support (a short b long e)} = \frac{\begin{array}{l}(\min(\mu\text{short (3)},\mu\text{long (25)}) + \min(\mu\text{short (1)},\mu\text{long (23)}) + \\ \min(\mu\text{short (5)},\mu\text{long (25)}) + \min(\mu\text{short (3)},\mu\text{long (27)}))\end{array}}{|S|}$$

We get Support (a short b long e) = 0.308, which is larger than *min_sup*.

Table 1. Audit Sequence Database

Sid	Audit Sequence
10	<(a,1),(b,4),(e,29)>
20	<(d,1),(a,2),(d,24)>
30	<(b,1),(a,11),(e,28)>
40	<(f,1), (b,5),(c,19)>
50	<(a,4),(b,5),(d,10),(e,28)>
60	<(a,0),(b,5),(e,30)>
70	<(j,2),(a,17),(h,17)>
80	<(c,3),(i,10),(f,18)>
90	<(h,4),(a,10),(b,21)>
100	<(g,0),(a,0),(b,3),(e,30)>

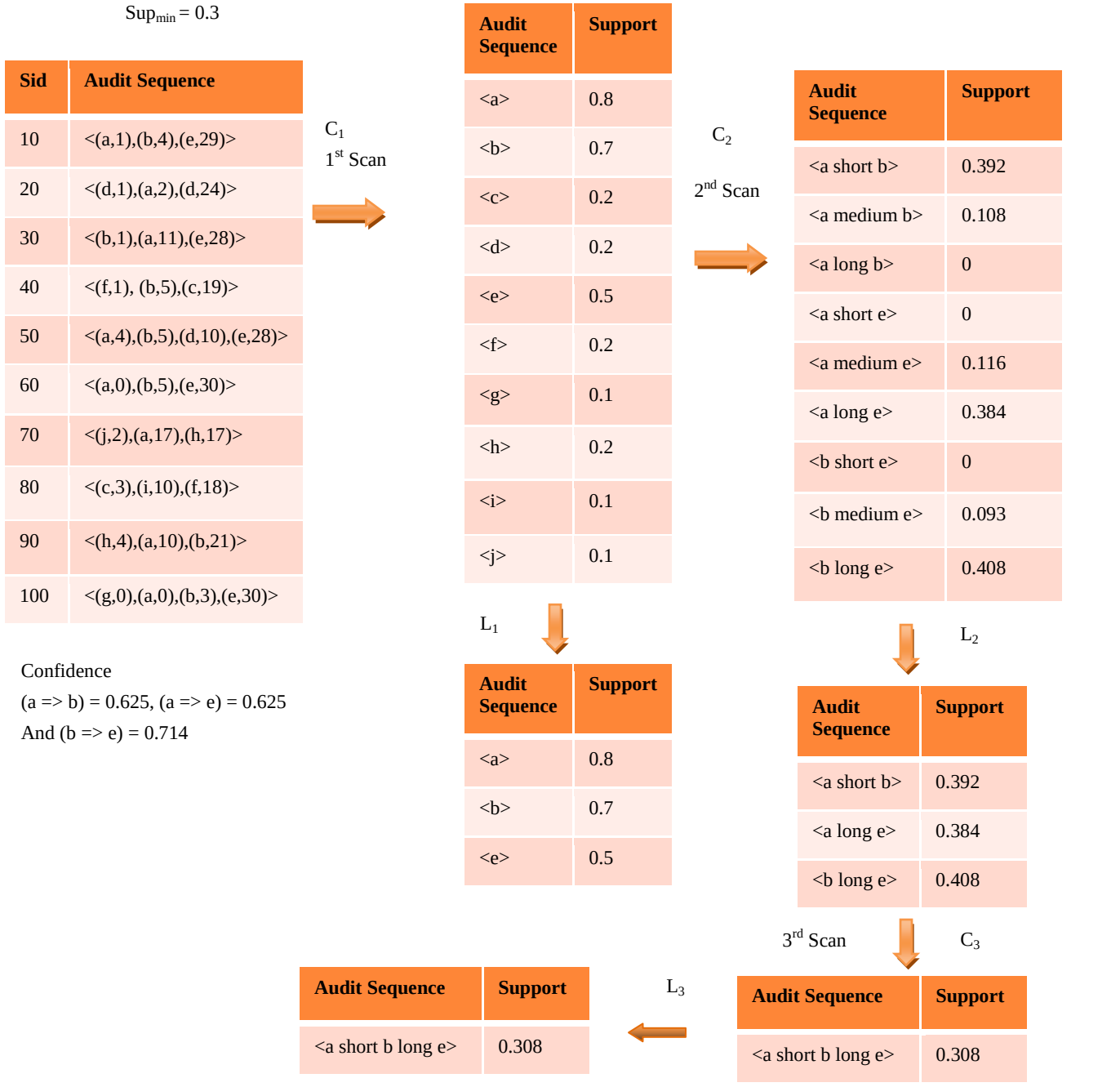

Figure 4. Large Audit sequences.

4. PROPOSED ALGORITHM

An algorithm is proposed, to implement the FTI sequential pattern mining to classify network audit data for IDS classification.

In the algorithm defined,

L_k - the set of all frequent k-FTI sequences,

C_k – the set of candidate k-FTI sequences.

The algorithm proceeds in phases, in the first phase, for k=1, L_1 is found from C_1: Clearly, C_1 can be generated by listing all distinct audit patterns in databases. For k=2, L_2 is generated from C_2, which is generated by joining L_1 with LT= {short, medium, long}.

Now for all k>2, all time interval audit sequences in C_k can be generated by joining the time interval audit sequences in $L_{k-1.}$ In the second phase, to determine the supports of all audit patterns in $C_{k,}$ a tree structure, called fuzzy candidate tree, is used as a basis. Basically, the candidate tree is similar to the prefix tree adopted in previous research [14]. The major difference lies in that the traditional approach connects each tree branch with an item name, whereas in the new approach two components are attached— an audit pattern name and a linguistic term.

Fuzzy time interval sequential pattern mining algorithm:

Input: Audit Sequence Database S, Minimum Support *min_sup*, and Linguistic Terms LT

Output: The complete set of FTI audit sequential patterns.

C_1= find all audit patterns in S.

L_1= {c Є C_1| $\left(\frac{c.count}{|S|}\right) \geq min\_sup$}

for (k=2; L_k-1≠ ∅ ; k++) {

C_k= new candidates generated from L_{k-1}

for each p_1 Є L_{k-1} {

for each p_2 Є L_{k-1} {

If (k=2) {

for each *ltd* Є LT {

c= p_1* ltd* p_2;

add c to C_k;

}

}

}

if (k>2)

Build the fuzzy candidate tree from C_k;

for each sequence s Є S

Traverse the fuzzy candidate tree and accumulate the supports;

L_k={c Є C_k | $\left(\frac{c.count}{|S|}\right) \geq min\_sup$}

}

return $\bigcup L_k$;

Thus the above algorithm mine frequent audit sequential pattern using FTI sequential pattern in network audit data to classify and detect intrusion.

5. CONCLUSION

Security audit data increased so dramatically, management and analysis of these security data become a critical and challenge issue. Sequential-pattern mining is useful in discovering audit patterns along with time from network audit databases.

In this paper, we contributed to the ongoing research on FTI sequential pattern mining by proposing an algorithm to detect and classify audit sequential patterns in network audit data. Anrong *et al* [20], addresses application of sequential pattern in intrusion detection. Their work implements PrefixSpan algorithm in the data mining module of network intrusion detection system.

IDS uses Boolean logic in determining whether or not an intrusion is detected and the use of fuzzy logic has been investigated as an alternative. 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 using varying degrees of truth. Because boundaries are not always clearly defined, fuzzy logic can be used to identify complex pattern or behavior variations. The paper defines the confidence of the FTI audit sequences, which is not yet defined in the previous researches.

6. REFERENCES

[1] Murali.A., Rao.M:A survey on intrusion detection approaches, In the First International Conference on Information and Communication Technologies .pp.233-240(2005)

[2] Nong.Y., Qiang.C., Borror.C.M: EWMA forecast of normal system activity for computer intrusion Detection. IEEE Trans, Reliab. 53(4), 557-566 (2004)

[3] Axelsson.S.: Intrusion detection systems: a survey and taxonomy. Technical report no. 99-15, Department of Computer Engineering . Chalmers University of Technology, Sewden (2000)

[4] Tian.J.F., Fu.Y., Wang.J-L: Intrusion detection combining multiple decision trees by fuzzy logic. In: Sixth International Conference on Parallel and Distributed Computing. Application an Technologies,5-8 December 2005. pp.256-258 (2005)

[5] S.kumar and E.H.Spafford, : A software architecture to support misuse intrusion detection, In *proceedings of the 18th National Information Security Conference*, pp 194-204(1995)

[6] K.Ilgun,R.A.Kemmerer,and P.A.Porras: State transition analysis: A rule-based intrusion detection approach, *IEEE Transactions on Software Engineering*, 1995,21. Pp.181-199(1995)

[7] T.Lunt,A.Tamaru, F.Gilham, R.Jagannathan, P.Neumann, H.Javitz, A.Valdes, and T.Garver: A real-time intrusion detection expert system (IDES)-final technical report, *Technical report, Computer Science Laboratory*, SRI International, Melo Park, California, February(1992)

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

[9] R. Agrawal and R. Srikant: Mining sequential patterns. In Proc. Int. Conf. Data Engineering, pp. 3–14(1995)

[10] 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)

[11] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. New York: Academic, (2001)

[12] Y. L. Chen, M. C. Chiang, and M. T. Ko: Discovering time-interval sequential patterns in sequence databases, *Expert Syst. Applicat.*, vol. 25, no. 3, pp. 343–354(2003)

[13] 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, vol.35, pp.959-972(2005)

[14] R. Agrawal and R. Srikant: Fast algorithms for mining association rules, in *Proc. Int. Conf. Very Large Data Bases*, pp. 487–499(1994)

[15] 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 (2001)

[16] 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 (2000)

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

[18] Zaki, M. J.: SPADE: An efficient algorithm for mining frequent sequences. Machine Learning Journal, 42(1/2), 31–60 (2001)

[19] Manish Saggar, Ashish Kumar Agrawal, Abhimanyu Lad: Optimization of Association Rule Mining using Improved Genetic Algorithms, IEEE International Conference on Systems, Man and Cybernetics, pp , pp 3725- 3729 (2004)

[20] 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)