

**MINING FUZZY ASSOCIATION RULES  
AND FUZZY FREQUENCY EPISODES  
FOR INTRUSION DETECTION**

**Jianxiong Luo**

**Jason@cs.msstate.edu**

**Susan M. Bridges<sup>1</sup>**

**Bridges@cs.msstate.edu**

<sup>1</sup>Corresponding author:  
Department of Computer Science  
Box 9637  
Mississippi State University  
Mississippi State, MS 39759

## **Abstract**

Lee, Stolfo, and Mok<sup>1</sup> have previously reported the use of association rules and frequency episodes for mining audit data to gain knowledge for intrusion detection. The integration of association rules and frequency episodes with fuzzy logic can produce more abstract and flexible patterns for intrusion detection, since many quantitative features are involved in intrusion detection and security itself is fuzzy. We present a modification of a previously reported algorithm for mining fuzzy association rules, define the concept of fuzzy frequency episodes, and present an original algorithm for mining fuzzy frequency episodes. We add a normalization step to the procedure for mining fuzzy association rules in order to prevent one data instance from contributing more than others. We also modify the procedure for mining frequency episodes to learn fuzzy frequency episodes. Experimental results show the utility of fuzzy association rules and fuzzy frequency episodes in intrusion detection.

## 1. INTRODUCTION

In recent years, computer security has become increasingly important and an international priority. There are two types of intrusion detection: misuse detection and anomaly detection.<sup>2</sup> Misuse detection can be applied to attacks that follow some fixed pattern and are usually constructed to examine intrusion patterns that have been recognized and reported by experts. This approach poses a problem when new types of attacks are encountered or if intruders attempt to disguise their behavior. Anomaly detection methods are designed to counter this kind of challenge by identifying patterns of normal behavior, with the assumption that an intrusion will usually include some deviation from this normal behavior. Artificial intelligence (AI) techniques have played an important role in both misuse detection and anomaly detection. AI techniques can be used for data reduction and classification tasks.<sup>3</sup> For example, many intrusion detection systems have been developed as rule-based expert systems based on the knowledge of system vulnerabilities or known attack patterns. An example is SRI's Intrusion Detection Expert System (IDES).<sup>4</sup> AI techniques have also been used to learning inductive rules. For example, sequential patterns can be learned by a system such as the Time-based Inductive Machine (TIM) for intrusion detection.<sup>5</sup> Neural networks can be used to predict future intrusions after training.<sup>6</sup> Lee, Stolfo, and Mok<sup>1</sup> have proposed data mining methods such as association rules and frequency episodes to mine normal patterns from audit data.

However, if a rule is directly dependent on audit data, “there is very little flexibility in this one-to-one (rule-to-audit record) representation.”<sup>7</sup> For example, an intrusion with a very small deviation from the patterns represented in the rules may not be matched and recognized. To improve the flexibility of an intrusion detection system, we have developed a method for integrating fuzzy logic with data mining methods for intrusion detection.

There are two main reasons to introduce fuzzy logic for intrusion detection. First, many quantitative features, both ordinal and categorical, are involved in intrusion detection<sup>8</sup> and can potentially be viewed as fuzzy variables. For instance, the CPU usage time and the connection duration are two examples of ordinal measurements. An example of a linear categorical measurement is the number of different TCP/UDP services initiated by the same source host. The second reason to introduce fuzzy logic for intrusion detection is that security itself includes fuzziness. Given a quantitative measurement, a range value or an interval can be used to denote a normal value. Then, any values falling outside the interval will be considered anomalous to the same degree regardless of their different distances to the interval. The same applies to values inside the interval, i.e., all will be viewed as normal to the same degree. Unfortunately, this causes an abrupt separation between normality and anomaly. The introduction of fuzziness to these quantitative features will help to smooth the abrupt separation. We describe an approach which integrates fuzzy logic with association rules and frequency episodes with the purpose of improving the performance of an intrusion detection system. Section 2 describes association rules and frequency episodes and how they have been

applied to intrusion detection in other work. Sections 3 and 4 describes our fuzzy association rules and fuzzy frequency episodes and their application to the intrusion detection problem.

## 2. DATA MINING METHODS FOR INTRUSION DETECTION

Data mining methods can be used to extract normal patterns from training data automatically and adaptively. Two data mining methods, association rules and serial frequency episodes, have been proposed for audit data gathering, feature selection, and off-line analysis for anomaly detection.<sup>1</sup> An association rule specifies the correlation among different features. A serial frequency episode represents a sequential pattern repeatedly occurring in the event sequence. An advantage here is that both the patterns among different features and the patterns among sequential events can be exploited.

### 2.1 Association Rules

Association rules were originally developed as a tool for analysis of retail sales. A piece of sales data usually includes information about a transaction, such as transaction date and items purchased.<sup>9</sup> Association rules can be used to find the correlation among different items in a transaction. For example, when a customer buys item A, item B will also be purchased by the customer with the probability of 90%. Agrawal and Srikant<sup>9</sup> have presented some fast algorithms to mine association rules, including algorithm Apriori. Using the notation of Agrawal and Srikant<sup>9</sup>, let  $\mathbf{D} = \{T_1, T_2, \dots, T_n\}$  be the transaction database with  $n$  transactions in total and  $\mathbf{I} = \{i_1, i_2, \dots, i_m\}$  be the set of all

the items where each  $i_j$  ( $1 \leq j \leq m$ ) represents one kind of item. Then each transaction  $T_l$

( $1 \leq l \leq n$ ) in  $D$  records the items purchased, i.e.,  $T_l \subseteq I$ . Define an itemset as a non-empty subset of  $I$ . An association rule will have the form:  $X \rightarrow Y, c, s$ , where  $X \subset I$ ,  $Y \subset I$ , and  $X \cap Y = \phi$ , i.e.,  $X$  and  $Y$  are disjoint itemsets. Here  $s$  represents the support of this association rule and  $c$  represents the confidence of this association rule. Assume the number of transactions that contains both the itemset  $X$  and the itemset  $Y$  is  $n'$ ; then  $s =$

$support(X \cup Y) = \frac{n'}{n}$  and  $c = \frac{support(X \cup Y)}{support(X)}$ . Intuitively,  $support(X)$  can be viewed as

the occurrence frequency of the itemset  $X$  in the whole transaction database  $D$ , while  $c$  indicates that when  $X$  is satisfied, there will be the certainty of  $c$  that  $Y$  is also true. Two thresholds, *minconfidence* (representing minimum confidence) and *minsupport* (representing minimum support), are used by the mining algorithm to find all association rules  $X \rightarrow Y, c, s$  such that  $c \geq minconfidence$  and  $s \geq minsupport$ . Any itemset  $X$  is called a large itemset if  $support(X) \geq minsupport$ . Usually, a mining algorithm involves two steps:

- (1) Find all the large itemsets of different lengths.
- (2) Construct association rules from every large itemset. This is actually a direct mapping process (see Agrawal and Srikant<sup>9</sup> for details).

## 2.2 Frequency Episodes

Mannila and Toivonen<sup>10</sup> have proposed an algorithm to discover simple serial frequency episodes from event sequences based on minimal occurrences. In Mannila and

Toivonen's method,<sup>10</sup> suppose  $S = \{ E_1, E_2, \dots, E_n \}$  is an event sequence with  $n$  events in total and  $A = \{ a_1, a_2, \dots, a_m \}$  is the set of all the event attributes. Each event  $E = \{ E.a_1, E.a_2, \dots, E.a_m \}$  in  $S$  consists of  $m$  values for all the event attributes.  $E$  is also associated with a timestamp denoted by  $E.T$ . Then a simple serial episode  $P(e_1, e_2, \dots, e_k)$  represents a sequential occurrence of  $k$  event variables where each  $e_i$  ( $1 \leq i \leq k$ ) is an event variable and for all  $i$  and  $j$  ( $1 \leq i < j \leq k$ ),  $e_i.T < e_j.T$ . Usually,  $k$  is much smaller than  $n$ , so  $1 \leq k \ll n$ . We use  $e^q$  to represent an event variable consisting of  $q$  event attributes, i.e.,  $e^q \{ attr_1 = v_1, attr_2 = v_2, \dots, attr_q = v_q \}$  where  $\{ e^q.attr_1, e^q.attr_2, \dots, e^q.attr_q \} \subseteq A$  and  $1 \leq q \leq m$ . In addition, each  $v_i$  ( $1 \leq i \leq q$ ) is a value from the domain of attribute  $attr_i$ . So,  $e^q$  is said to have an occurrence in an event  $E$  if for all  $i$  ( $1 \leq i \leq q$ ),  $E.(e^q.attr_i) = e^q.v_i$ .

According to Mannila and Toivonen,<sup>10</sup> given a time interval  $[t, t']$ , an episode  $P(e_1, e_2, \dots, e_k)$  is said to occur at interval  $[t, t']$  if  $t \leq e_1.T$  and  $t' \geq e_k.T$ . Define an occurrence of  $P(e_1, e_2, \dots, e_k)$  at interval  $[t, t']$  as minimal if there does not exist another of occurrence of  $P(e_1, e_2, \dots, e_k)$  at the subinterval of  $[u, u'] \subset [t, t']$ . Given a threshold of *window* (representing timestamp bounds), the frequency of  $P(e_1, e_2, \dots, e_k)$  is defined as  $frequency(P) = |\{ [t, t'] \mid (t' - t \leq window) \text{ and the occurrence of } P \text{ at interval } [t, t'] \text{ is minimal} \}|$ . The frequency of  $P(e_1, e_2, \dots, e_k)$  in the event sequence  $S$  is the total number

of minimal occurrences in any interval smaller than *window*. So, given another threshold

*minfrequency* (representing minimum frequency), an episode  $P(e_1, e_2, \dots, e_k)$  is called

frequent if  $\frac{frequency(P)}{n-k+1} \geq minfrequency$ . Since in our domain  $k \ll n$

$\frac{frequency(P)}{n-k+1} \approx \frac{frequency(P)}{n}$  will hold. Therefore, in our implementation, an episode

will be considered frequent, if  $\frac{frequency(P)}{n} \geq minfrequency$ .

Mannila and Toivonen<sup>10</sup> use an algorithm similar to Apriori except for the differences between calculating episode frequencies and calculating itemset supports.

Like association rules, episode rules can be also directly established from frequency episodes. Given a frequency episode  $P(e_1, e_2, \dots, e_k)$ , there will be  $k-1$  non-empty

ordered sub-episodes  $P_i(e_1, e_2, \dots, e_i) \subset P$  where  $1 \leq i \leq k-1$ . Then given another

threshold *minconfidence* (representing minimum confidence), a simple serial episode rule

can be constructed as  $P_i \rightarrow Q_i, c, s, w$ , where  $P_i(e_1, e_2, \dots, e_i) \subset P$ ,

$Q_i(e_{i+1}, e_{i+2}, \dots, e_k) = P - P_i \subset P$ ,  $s=frequency(P) \geq minfrequency$ ,

$c = \frac{frequency(P)}{frequency(P_i)} \geq minconfidence$ , and  $w = window$ . Here according to Lee, Stolfo, and

Mok,<sup>1</sup> both  $P_i$  and  $Q_i$  are specified in the same timestamp bound, i.e., the same

threshold  $w$ .

The last episode rule  $P_{k-1} \rightarrow Q_{k-1}, c, s, w$  is of most interest since it can be used to predict the  $k^{th}$  event given the previous  $k-1$  events. We have used this kind of episode rule in our experiments dealing with intrusion detection.

### 3. INTEGRATION OF FUZZY LOGIC WITH DATA MINING

Although association rules and frequency episodes can be mined from audit data for anomaly intrusion detection, the mined rules or episodes are at the data level. Integrating fuzzy logic with association rules and frequency episodes allows one to extract more abstract patterns at a higher level.

#### 3.1 Fuzzy Association Rules

Srikant and Agrawal<sup>12</sup> have described a very popular algorithm for mining quantitative association rules that partitions quantitative attributes into different intervals. Unfortunately, a “sharp boundary problem” results from using interval partitions.<sup>13</sup> For example, suppose  $[1, 5]$  and  $[6, 10]$  are two intervals created on a quantitative attribute as shown in Figure 1. If the minimum support threshold is set at 30%, the interval  $[6, 10]$  will not gain enough support regardless of the large support near its left boundary, as shown in Figure 1. That is to say, although the value 5 has a large support and lies near the interval  $[6, 10]$ , it will not make any contribution when counting the support of  $[6, 10]$ .

In intrusion detection, the sharp separation of intervals may raise additional problems. For example, suppose the interval  $[1, 5]$  is mined as a normal pattern for the quantitative attribute. The values 6 and 10 will both be considered abnormal regardless of the difference in their deviations from the normal pattern. Likewise, a normal behavior that varies slightly from normal may fall outside the interval representing a normal pattern and be considered an anomaly. Similarly, an intrusion with a small variance may fall inside the interval and be undetected.

To address the “sharp boundary problem”, Kuok, Fu, and Wong<sup>13</sup> have proposed to mine fuzzy association rules by using fuzzy sets to categorize a quantitative attribute. In the above example, the two intervals will be replaced by two fuzzy sets. Suppose the value 5 has membership degree of 0.9 in the first set and 0.3 in the second set. Then it will contribute 0.9 to the support of the first fuzzy set and 0.3 to the second one. However, this means that the value 5 will be more important than other values since the sum of its contributions to different fuzzy sets has become greater than 1. In our method we address this shortcoming of Kuok, Fu, and Wong’s approach by introducing an additional normalization process.

### 3.2 Mining Fuzzy Association Rules

According to Kuok, Fu, and Wong's method,<sup>13</sup> suppose we are given the complete item set  $I = \{i_1, i_2, \dots, i_m\}$  where each  $i_j$  ( $1 \leq j \leq m$ ) denotes a categorical or quantitative (fuzzy) attribute. Let  $f(i_j)$  represent the maximum number of categories (if  $i_j$  is categorical) or the maximum number of fuzzy sets (if  $i_j$  is fuzzy) and  $m_{i_j}(l, v)$

represent the membership degree of  $v$  in the  $l^{th}$  category or fuzzy set of  $i_j$ . If  $i_j$  is categorical,  $m_{i_j}(l, v) = 0$  or  $m_{i_j}(l, v) = 1$ . If  $i_j$  is fuzzy,  $0 \leq m_{i_j}(l, v) \leq 1$ . Srikant and Agrawal<sup>12</sup> introduce the idea of mapping the categories (or fuzzy sets) of an attribute to a set of consecutive integers. Then an itemset  $X^k$  ( $1 \leq k \leq m$ ) can be expressed as  $X^k \{item_1 = c_1, item_2 = c_2, \dots, item_k = c_k\}$  where  $\{X^k.item_1, X^k.item_2, \dots, X^k.item_k\} \subseteq I$  and for all  $j$  ( $1 \leq j \leq k$ ),  $1 \leq c_j \leq f(i_j)$ .

So, given a transaction  $T = \{T.i_1, T.i_2, \dots, T.i_m\}$ ,  $T.i_j$  ( $1 \leq j \leq m$ ) represents a value of the  $j^{th}$  attribute and can be mapped to  $\{(l, m_{i_j}(l, T.i_j)) \mid \text{for all } l, 1 \leq l \leq f(i_j)\}$ .

However, when using Kuok, Fu, and Wong's algorithm, if  $i_j$  is fuzzy,  $\sum_{l=1}^{f(i_j)} m_{i_j}(l, T.i_j)$

does not always equal to 1. We have developed a normalization process as follows:

$$m'_{i_j}(l, T.i_j) = \begin{cases} \frac{m_{i_j}(l, T.i_j)}{\sum_{l=1}^{f(i_j)} m_{i_j}(l, T.i_j)} & \text{if } i_j \text{ is fuzzy;} \\ m_{i_j}(l, T.i_j) & \text{if } i_j \text{ is categorical.} \end{cases}$$

Then, for an itemset  $X^k \{item_1 = c_1, item_2 = c_2, \dots, item_k = c_k\}$  where  $1 \leq k \leq m$ ,

its support contributed by  $T$  will be:

$$\prod_{j=1}^k m'_{X^k.item_j}(X^k.c_j, T.(X^k.item_j)).$$

Here we use the *product* to calculate an itemset's support because given a transaction

$T = \{T.i_1, T.i_2, \dots, T.i_m\}$  and any attribute set  $\{item_1, item_2, \dots, item_k\}$  ( $1 \leq k \leq m$ ),

$\sum_{\forall c_j \in [1, f(item_j)]} \left( \prod_{j=1}^k m'_{item_j}(c_j, T.item_j) \right) = 1$  will hold. That is to say, for any item or

any combination of items, the support from a transaction will be always 1.

The algorithm for constructing  $C_k$  from  $L_{k-1}$  ( $k \geq 2$ ) is shown in Figure 2. The rest of the algorithm for fuzzy association rules is similar to the Apriori algorithm for Boolean association rules.<sup>9</sup>

In this algorithm, normalization is introduced to ensure that every transaction is counted only one time for an item or any combination of items, either categorical or fuzzy. For example, suppose  $I = \{level, age\}$  where *level* is a categorical attribute with the domain of {freshman, sophomore, junior, senior, graduate} and *age* is a quantitative attribute with three fuzzy sets {young, medium, old}. A transaction  $T = \{\text{graduate}, 25\}$  will be mapped to  $\{(graduate, 1)\}, \{(young, 0.2), (medium, 0.9), (old, 0.1)\}$ . Without normalization, it would increase the support of itemset  $\{level = graduate, age = young\}$  by 0.2, the support of itemset  $\{level = graduate, age = medium\}$  by 0.9, and the support of itemset  $\{level = graduate, age = old\}$  by 0.1. That is to say, this transaction will be counted  $0.2+0.9+0.1=1.2$  times for the item *age*. However, it is unreasonable for one transaction to contribute more than others. In contrast, the normalization process will further transform the transaction  $T$  into  $\{(graduate, 1)\}, \{(young, 0.167), (medium, 0.75), (old, 0.083)\}$ , for a total contribution of 1.0 for the item *age*.

### 3.3 Fuzzy Frequency Episodes

In this section, we propose an idea of integrating fuzzy logic with frequency episodes. The need to develop fuzzy frequency episodes comes from the involvement of quantitative attributes in an event. That is to say, given the set of event attributes  $A = \{a_1, a_2, \dots, a_m\}$ , each attribute  $a_j$  ( $1 \leq j \leq m$ ) may be categorical or quantitative (fuzzy). Suppose  $f(a_j)$  represents the maximum number of categories (if  $a_j$  is categorical) or the maximum number of fuzzy sets (if  $a_j$  is fuzzy), and  $m_{a_j}(l, v)$  represents the membership degree of  $v$  in the  $l^{\text{th}}$  category or fuzzy set of  $a_j$ . If  $a_j$  is categorical,  $m_{a_j}(l, v) = 0$  or  $m_{a_j}(l, v) = 1$ . If  $a_j$  is fuzzy,  $0 \leq m_{a_j}(l, v) \leq 1$ . Similarly, for an event attribute, its categories or fuzzy sets can be mapped to consecutive integers. Then an event variable  $e^k$  can be expressed as  $e^k \{attr_1 = c_1, attr_2 = c_2, \dots, attr_k = c_k\}$  where  $\{e^k.attr_1, e^k.attr_2, \dots, e^k.attr_k\} \subseteq A$  and for all  $j$  ( $1 \leq j \leq k$ ),  $1 \leq c_j \leq f(a_j)$ . We define two event variables  $e^p \{attr_1 = c_1, attr_2 = c_2, \dots, attr_p = c_p\}$  and  $e^q \{attr_1' = c_1', attr_2' = c_2', \dots, attr_q' = c_q'\}$  as *homogeneous*, if  $\{e^p.attr_1, e^p.attr_2, \dots, e^p.attr_p\} = \{e^q.attr_1', e^q.attr_2', \dots, e^q.attr_q'\}$ , which also indicates that  $p = q$ . It is obvious that an event variable is homogeneous to itself.

So, given an event  $E = \{E.a_1, E.a_2, \dots, E.a_m\}$ ,  $E.a_j$  ( $1 \leq j \leq m$ ) represents a value of the  $j^{\text{th}}$  attribute and can be mapped to  $\{(l, m_{a_j}(l, E.a_j)) \mid \text{for all } l, 1 \leq l \leq f(a_j)\}$ .

However, if  $a_j$  is fuzzy,  $\sum_{l=1}^{f(a_j)} m_{a_j}(l, E.a_j)$  does not always equal to 1. A normalization

process is used as follows:

$$m'_{a_j}(l, E.a_j) = \begin{cases} \frac{m_{a_j}(l, E.a_j)}{f(a_j)} & \text{if } a_j \text{ is fuzzy;} \\ \sum_{l=1} m_{a_j}(l, E.a_j) & \\ m_{a_j}(l, E.a_j) & \text{if } a_j \text{ is categorical.} \end{cases}$$

Then, for an event variable  $e^k \{attr_1 = c_1, attr_2 = c_2, \dots, attr_k = c_k\}$  where  $1 \leq k \leq m$ , its occurrence in  $E$  is no longer counted as either 0 or 1. Instead, it is defined as:

$$occurrence(e^k, E) = \prod_{j=1}^k m'_{e^k.attr_j}(e^k.c_j, E.(e^k.attr_j)).$$

And the minimal occurrence of an episode is the *product* of the occurrences of its event variables.

That is to say, an event  $E$  may support several event variable occurrences due to the introduction of fuzzy sets. However, a side effect may arise. For example, consider the event sequence  $\{E_1, E_2, E_3\}$  within the *window* threshold.  $A, B, C$ , and  $D$  are event variables in which  $A$  and  $B$  are homogeneous but  $A \neq B$ , and  $C$  and  $D$  are homogeneous but  $C \neq D$ . Suppose  $occurrence(A, E_1) = 0.8$ ,  $occurrence(B, E_1) = 0.2$ ,  $occurrence(A, E_2) = 0.1$ ,  $occurrence(B, E_2) = 0.9$ ,  $occurrence(C, E_3) = 0.9$ , and  $occurrence(D, E_3) = 0.1$ . Then the minimal occurrence of episode  $\{A, C\}$  will become 0.09 because  $\{E_2.A, E_3.C\}$  is minimal by replacing  $\{E_1.A, E_3.C\}$  which will contribute

0.72. So, a small occurrence of an event variable may change the minimal occurrence of an episode in the event sequence.

To address this problem, we introduce another user-specified threshold *minoccurrence* to represent the minimum occurrence required for an event variable. So, given an event variable  $e^k$ , if  $occurrence(e^k, E) < minoccurrence$ , it will be claimed not to occur in  $E$ . In detail, the following normalization process will be further conducted:

$$occurrence'(e^k, E) = \begin{cases} 0 & \text{if } (occurrence(e^k, E) < minoccurrence); \\ \frac{occurrence(e^k, E)}{\sum_{e^q} occurrence(e^q, E)} & \text{if } (occurrence(e^k, E) \geq minoccurrence). \end{cases}$$

Here every  $e^q$  is homogeneous to  $e^k$  and  $occurrence(e^q, E) \geq minoccurrence$ . For instance, if  $minoccurrence = 0.2$ ,  $E_1$  will contribute 0.8 to  $A$  and 0.2 to  $B$ ,  $E_2$  will contribute 1 to  $B$ , and  $E_3$  will contribute 1 to  $C$ . As a matter of fact, if  $minoccurrence$  is set above 0.5, for any event, only one event variable will be claimed to occur in it and its occurrence will be normalized to 1. In this case, it will be the same as categorizing every quantitative attribute by intervals.

Other than the difference in calculating the frequency (or minimal occurrence) of an episode, the rest of this algorithm is similar to Mannila and Toivonen's<sup>10</sup> algorithm for mining frequency episodes.

## 4. EXPERIMENTS AND RESULTS

Association rules and frequency episodes have been proposed for feature selection, as well as audit data gathering.<sup>1</sup> On the other hand, the association rules and episode rules mined from training data that represents normal behavior can be also directly used for anomaly detection.<sup>1</sup> However, these rules are usually at the data level. For example, for the quantitative feature of “connection duration,” a rule may contain such a component as “connection duration = 5 seconds” or “5 seconds  $\leq$  connection duration  $\leq$  10 seconds”. With the integration of fuzzy logic, more general rules can be produced at a higher and more abstract level.

Another advantage resulting from the integration of fuzzy logic is that fuzzy association rules and fuzzy frequency episodes can be applied to temporal statistical measurements which are quantitative and security-related. Statistical analysis has been widely used to construct normal patterns for anomaly detection in systems such as SRI's IDES and NIDES.<sup>4</sup> Some statistical measurements have been also proven to be able to improve the accuracy of intrusion detection.<sup>11</sup> However, these statistical features are usually incorporated as additional measurements manually. By using fuzzy association rules and fuzzy frequency episodes, normal patterns for these statistical features can be automatically created and used for anomaly detection.

#### 4.1 Anomaly Detection

The first set of experiments was designed to investigate the applicability of fuzzy association rules and fuzzy frequency episodes for anomaly detection. Since a large amount of actual intrusion data is usually very hard to collect, some normal data with behavior different than that used for training can be treated as anomalous.<sup>1</sup> One of the servers in the Department of Computer Science at Mississippi State University has been monitored and its real-time network traffic data has been collected by *tcpdump*. Data preprocessing is conducted by use of *sanitize* (downloaded from <http://ita.ee.lbl.gov/html/software.html> on 1 March 1999). Porras and Valdes<sup>13</sup> and Lee and Stolfo<sup>11</sup> suggest several quantitative features of network traffic that they feel can be used for intrusion detection. Based on their suggestions, we extract the following four temporal statistical measurements from the network traffic data:

SN – the number of SYN flags appearing in TCP packet headers during last 2 seconds;  
FN – the number of FIN flags appearing in TCP packet headers during last 2 seconds;  
RN – the number of RST flags appearing in TCP packet headers during last 2 seconds;  
PN – the number of different destination ports during last 2 seconds.

Here statistical computation is done for overlapping 2 second time periods as shown in Figure 3.

Each of the above four quantitative features is viewed as a fuzzy variable and is divided into three fuzzy sets: *LOW*, *MEDIUM*, and *HIGH*. Membership function definitions have been developed for fuzzy variables representing each of the features of the network being monitored. The fuzzy association rule algorithm has been applied to mine the correlation among the first three features, and the fuzzy frequency episode algorithm has been applied to mine sequential patterns for the last feature.

The network traffic data was partitioned into different segments according to the time slots when the sets were collected (i.e., morning, afternoon, evening, and night) since different time slots likely exhibit different behavior. In this experiment, traffic data in the afternoon was used as training data. Anomaly detection was then conducted on traffic data from afternoon, evening, and night. A detailed specification of the training and test data sets is given in Table 1.

Normal patterns (represented by fuzzy association rules and fuzzy episode rules) are first established by mining the training data. An example of a fuzzy association rule mined from the training data is:  $\{ SN = LOW, FN = LOW \} \rightarrow \{ RN = LOW \}$ , 0.924, 0.49. This means the pattern  $\{ SN = LOW, FN = LOW, RN = LOW \}$  occurred in 49% of the training cases. In addition, when  $\{ SN = LOW, FN = LOW \}$  occurs, there will be 92.4% probability that  $\{ RN = LOW \}$  will also occur. An example of a fuzzy episode rule is:  $\{ PN = LOW, PN = MEDIUM \} \rightarrow \{ PN = MEDIUM \}$ , 0.854, 0.108, 10 seconds. This means that with a *window* threshold of 10 seconds, the frequency of the serial episode  $\{ PN = LOW, PN = MEDIUM, PN = MEDIUM \}$  is 10.8% and when  $\{ PN = LOW, PN = MEDIUM \}$  occurs,  $\{ PN = MEDIUM \}$  will follow with an 85.4% probability.

Then for each test case, new patterns were mined using the same algorithms and the same parameters. These new patterns were then compared to the normal patterns created from the training data. If they are similar enough, no intrusion is detected; otherwise, an anomaly will be alarmed.

The similarity function proposed by Lee, Stolfo, and Mok<sup>1,15</sup> used a user-defined threshold, e.g., 5%. Given two rules with the same LHS and RHS, if both their confidences and their supports are within 5% of each other, these two rules are considered similar. This approach exhibits Kuok, Fu, and Wong's<sup>13</sup> sharp boundary problem. For example, given a rule R which represents a normal pattern and two test rules R' and R", if both R' and R" fall inside the threshold, there will be no measurement of the difference between the similarity of R and R' and the similarity of R and R". Likewise, when both R' and R" fall outside the threshold, there is no measure of their dissimilarities with R.

Instead, we introduce a new similarity evaluation function which is continuous and monotonic. Given a normal association rule:

$$R_1: X \rightarrow Y, c, s,$$

and a new association rule:

$$R_2: X' \rightarrow Y', c', s',$$

where X, Y, X', and Y' are itemsets, define

$$similarity(R_1, R_2) = \begin{cases} 0 & \text{if } ((X \neq Y) \vee (X' \neq Y')); \\ \max\left(0, 1 - \max\left(\frac{|c - c'|}{c}, \frac{|s - s'|}{s}\right)\right) & \text{if } ((X = Y) \wedge (X' = Y')). \end{cases}$$

Given two rule sets S1 (of normal patterns) and S2 (of new patterns), define

$$s = \sum_{\substack{\forall R_1 \in S_1 \\ \forall R_2 \in S_2}} similarity(R_1, R_2).$$

Then, like the definition in Lee, Stolfo, and Mok,<sup>1</sup> we define

$$\text{similarity}(S_1, S_2) = \frac{s}{|S_1|} * \frac{s}{|S_2|},$$

where  $|S_1|$  and  $|S_2|$  are the total number of rules in  $S_1$  and  $S_2$ , respectively. Here  $\frac{s}{|S_1|}$  is

actually the percentage of normal patterns covered by the new patterns, and  $\frac{s}{|S_2|}$  is the

percentage of new patterns covered by the normal patterns. The similarity evaluation for fuzzy episode rules is almost the same as for fuzzy association rules, except that there is one more parameter  $w$  (of window length) for an episode rule. It is required that the window thresholds be identical when two episode rules are evaluated for their similarity.

The purpose of the first experiment in this set was to determine the amount of training data (duration) needed to demonstrate differences in behavior for different time periods. In this experiment, training sets of different duration (all from the same time period, i.e., afternoon) were used to mine fuzzy association rules (see Table 1 for a more detailed description of the data). The similarity of each set of rules derived from training data of different duration was compared to test data for different time periods. The results from this experiment are shown in Figure 4. These results show that the fuzzy association rules derived from test data for the same time of the day as the training data (afternoon) were very similar to the rules derived from the training data. Rules derived from evening data were less similar and rules derived from late night data were the least similar. This confirms the hypothesis that fuzzy association rules are able to distinguish different behavior. This experiment also demonstrates that there is no difference in the

similarity measures when the duration of training data is increased from 3 hours to 6 hours.

The purpose of the second experiment in this set was to further demonstrate the capability of fuzzy association rules for anomaly detection. In this experiment, 3 hours of traffic data (afternoon) was selected as the training data based on results from the first experiment. Nine test data sets from three different time periods, i.e., afternoon, evening, and late night were used (see Table 1 for a more detailed description of the data). The similarity of fuzzy association rules derived from training data was compared to each test data set. The results from this experiment are shown in Figure 5. The results show that the fuzzy association rules derived from the test data sets for the same time period as the training data (afternoon) were more similar to the rules derived from training data than any other test data set from different time periods, i.e., evening, and late night. Rules derived from any evening test data set were more similar than rules derived from any late night test data set. This further confirms the capability of fuzzy association rules in distinguishing different behavior.

The next two experiments were similar to the first two experiments, except that fuzzy episode rules were mined for anomaly detection instead of fuzzy association rules. The results are consistent with those from the first two experiments and are shown in Figure 6 and Figure 7.

Some observations can be made from these experimental results: (1) in both the fuzzy association rule training process and the fuzzy frequency episode training process, there are no significant changes in similarity when using 3 hours of training data as

opposed to 6 hours of training data. Therefore, 3 hours of training data was used for the remaining experiments; (2) similar results were obtained from both fuzzy association rules and fuzzy frequency episodes. The test cases of T1, T2, and T3 are most similar to normal patterns since all of them, as well as training data, are network traffic in the afternoon. T7, T8, and T9 are most different from normal patterns; this is to be expected since this data represents network traffic in the middle of the night when the usage of the network is lightest.

The results have also shown that, given the same training data set and test data set, their similarity as measured by mining fuzzy association rules is different from their similarity as measured by mining fuzzy episode rules. This is not unexpected, since fuzzy association rules and fuzzy episode rules use different features, which may have different effects on anomaly detection.

#### **4.2 Detecting Simulated Intrusions**

The second set of experiments was designed to further test the capability of fuzzy association rules and fuzzy frequency episodes for anomaly detection by using simulated intrusion data. Three network traffic data sets in *tcpdump* format were downloaded from <http://iris.cs.uml.edu:8080> and used for the second set of experiments. These data sets were collected by the Institute for Visualization and Perception Research at University of Massachusetts Lowell with the purpose of providing an evaluation method for different data mining techniques or some combinations of these techniques.<sup>16</sup> Among these data sets, *baseline* represents normal patterns, *network1* includes simulated IP spoofing

intrusions in which an intruder tries to access a remote host by guessing its IP sequence numbers, and *network3* includes simulated port scanning intrusions in which an intruder attempts to collect information about hosts or applications running on the network.

A program was first written to extract information about the same four temporal statistical measurements used in the previous set of experiments directly from the raw data. The data set *baseline* was segmented into two parts. The first part was used as training data and the second part was used as test data. *Network1* and *network3* were used as the other two test data sets.

The purpose of the first experiment in this set was to test the capability of fuzzy association rules for distinguishing simulated intrusions from normal behavior. The purpose of the second experiment in this set was to test the capability of fuzzy episode rules for distinguishing simulated intrusions from normal behavior. The results shown in Figures 8 and 9 provide additional evidence that anomalies can be detected by use of fuzzy association rules and fuzzy episode rules.

## 5. CONCLUSIONS

Intrusion detection is an important but complex task for a computer system. Data mining methods are capable of extracting patterns automatically and adaptively from a large amount of data. Association rules and frequency episodes have been used to mine training data to establish normal patterns for anomaly detection. However, these patterns are usually at the data level, with the result that normal behavior with a small

variation may not match a pattern and will be considered anomalous. In addition, an actual intrusion with a small deviation may match the normal patterns and thus not be detected. We have demonstrated that the integration of fuzzy logic with association rules and frequency episodes generates more abstract and flexible patterns for anomaly detection.

We have extended previous work by Lee, Stolfo, and Mok<sup>1</sup> in the areas of using association rules and frequency episodes for anomaly detection by introducing fuzzy logic. We add a normalization step to the procedure for mining fuzzy association rules by Kuok, Fu, and Wong<sup>13</sup> in order to prevent one data instance from contributing more than others. We modify the procedure of Mannila and Toivonen<sup>10</sup> for mining frequency episodes to learn fuzzy frequency episodes. We use fuzzy association rules and fuzzy frequency episodes to extract patterns for temporal statistical measurements at a higher level than the data level. We have developed a similarity evaluation function which is continuous and monotonic for the application of fuzzy association rules and fuzzy frequency episodes in anomaly detection. Our experimental results have shown the utility of fuzzy association rules and fuzzy episode rules in intrusion detection.

## REFERENCES

1. W. S. Lee, W., S. Stolfo, and K. Mok. "Mining audit data to build intrusion detection models," *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, New York, New York, August 27-31, 1998, edited by Rakesh Agrawal, and Paul Stolorz, 66-72. New York, NY: AAAI Press. (1998)
2. A. Sundaram, "An introduction to intrusion detection," (1996) (Downloaded from <http://www.cs.purdue.edu/coast/archive/data/categ24.html> on 10 March 1999.)
3. J. Frank, J. "Artificial intelligence and intrusion detection: Current and future directions," *Proceedings of the 17<sup>th</sup> National Computer Security Conference*, October, 1994.
4. T. Lunt and R. Jagannathan. "A prototype real-time intrusion-detection expert system," *Proceedings of 1988 IEEE Computer Society Symposium on Research in Security and Privacy*, Oakland, California, April 18-21, 1988, 59-66. Los Alamitos, CA: IEEE Computer Society Press.
5. H. Teng, K. Chen, and S. Lu. "Adaptive real-time anomaly detection using inductively generated sequential patterns," *Proceedings of 1990 IEEE Computer Society Symposium on Research in Security and Privacy*, Oakland, California, May 7-9, 1990, 278-84. Los Alamitos, CA: IEEE Computer Society Press.
6. H. Debar, M. Becker, and D. Siboni. "A neural network component for an intrusion detection system," In *Proceedings of 1992 IEEE Computer Society Symposium on Research in Security and Privacy*, Oakland, California, May 4-6, 1992, 240-50. Los Alamitos, CA: IEEE Computer Society Press.
7. K. Ilgun, and A. Kemmerer, "State transition analysis: A rule-based intrusion detection approach," *IEEE Transaction on Software Engineering* 21(3): 181-99 (1995).
8. T. Lunt, "Detecting intruders in computer systems," In *Proceedings of 1993 Conference on Auditing and Computer technology*. (Downloaded from <http://www2.csl.sri.com/nides/index5.html> on 3 February 1999.)
9. R. Agrawal, and R. Srikant. "Fast algorithms for mining association rules." *Proceedings of the 20<sup>th</sup> International Conference on Very Large Databases*,

- Santiago, Chile, September 12-15, 1994, 487-99. San Francisco, CA: Morgan Kaufmann.
10. H. Mannila and H. Toivonen. "Discovering generalized episodes using minimal occurrences," *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Portland, Oregon, August, 1996, 146-51.
  11. W. Lee and S. Stolfo. "Data mining approaches for intrusion detection." *Proceedings of the 7<sup>th</sup> USENIX security symposium, 1998*. (Downloaded from <http://www.cs.columbia.edu/~sal/recent-papers.html> on 10 March 1999.)
  12. R. Srikant and R. Agrawal, "Mining quantitative association rules in large relational tables," *Proceedings of ACM SIGMOD International Conference on Management of Data*, June 4-6, 1996, 1-12.
  13. C. Kuok, A. Fu, and M. Wong. "Mining fuzzy association rules in databases," *SIGMOD Record* 27(1): 41-6. (1998)
  14. P. Porras and A. Valdes. "Live traffic analysis of TCP/IP gateways," *Proceedings of the 1998 ISOC symposium on network and distributed systems security*, March, 1998.
  15. W. Lee, S. Stolfo, and K. Mok, "A data mining framework for building intrusion detection models," (1999) (Downloaded from <http://www.cs.columbia.edu/~sal/recent-paper.html> on 10 March 1999.)
  16. The Institute for Visualization and Perception Research, University of Massachusetts Lowell. (1998). *Information Exploration Shootout*. <http://iris.cs.uml.edu:8080> (Accessed 1 March 1999).

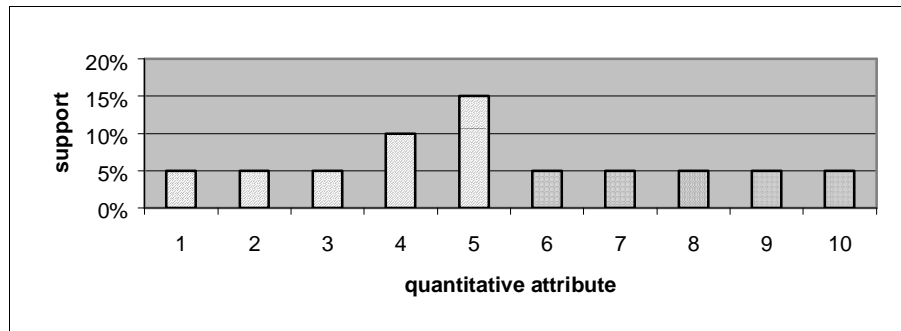


Figure 1. Example of the sharp boundary problem

```
 $C_k = \phi;$   
select  $\{X^{k-1}.item_1 = X^{k-1}.c_1, X^{k-1}.item_2 = X^{k-1}.c_2, \dots,$   
 $\dots, X^{k-1}.item_{k-1} = X^{k-1}.c_{k-1}, Y^{k-1}.item_{k-1} = Y^{k-1}.c_{k-1}\}$  into  $C_k$   
from  $L_{k-1}$   
where  $(X^{k-1} \in L_{k-1}) \wedge$   
 $(Y^{k-1} \in L_{k-1}) \wedge$   
 $(\forall j, 1 \leq j \leq k-2, (X^{k-1}.item_j = Y^{k-1}.item_j) \wedge (X^{k-1}.c_j = Y^{k-1}.c_j)) \wedge$   
 $(X^{k-1}.item_{k-1} < Y^{k-1}.item_{k-1});$   
forall itemsets  $Z^k \in C_k$  do begin  
  if (there exists a sub-itemset  $Z^{k-1} \subset Z^k$  and  $Z^{k-1} \notin L_{k-1}$ )  
    then  $C_k = C_k - \{Z^k\};$   
return  $C_k;$ 
```

Figure 2. Candidate Generation Algorithm for Fuzzy Association Rules

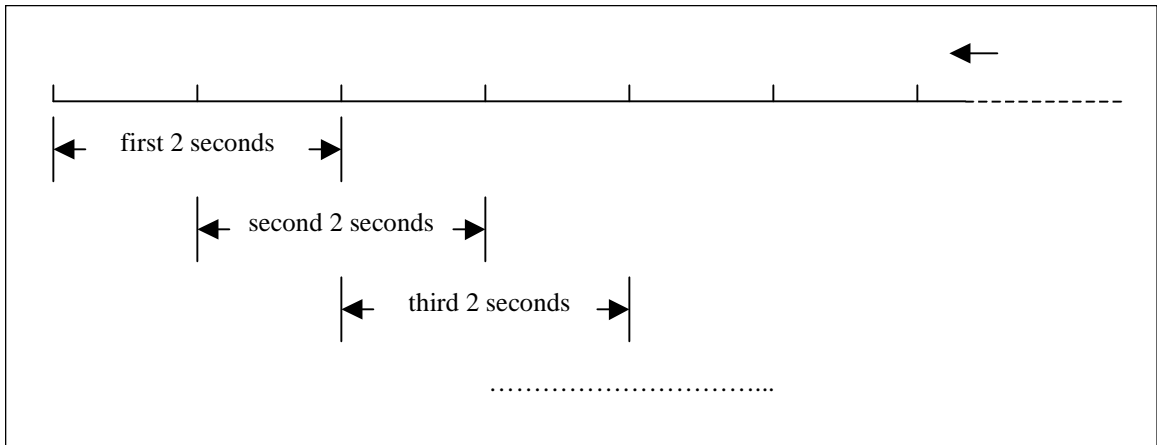


Figure 3. . Specification of Temporal Statistical Measurements Used in the Experiments

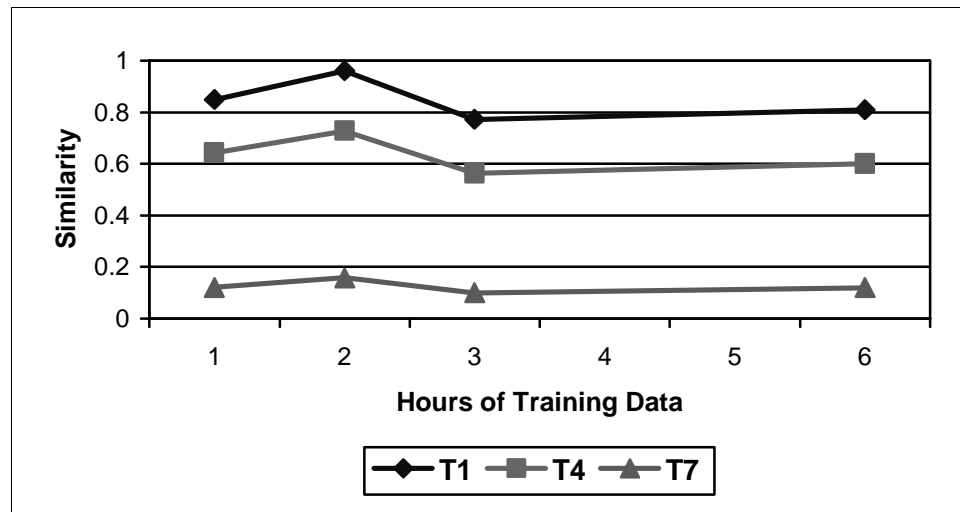


Figure 4: Comparison of Similarities Between Different Training and Test Data Sets for Fuzzy Association Rules (minconfidence=0.6; minsupport=0.1)

**Training Data Sets:** 1 hour of training data, 2 hours of training data, 3 hours of training data, and 6 hours of training data (all from the afternoon)

**Test Data Sets:** T1 (afternoon), T4 (evening), and T7 (late night)

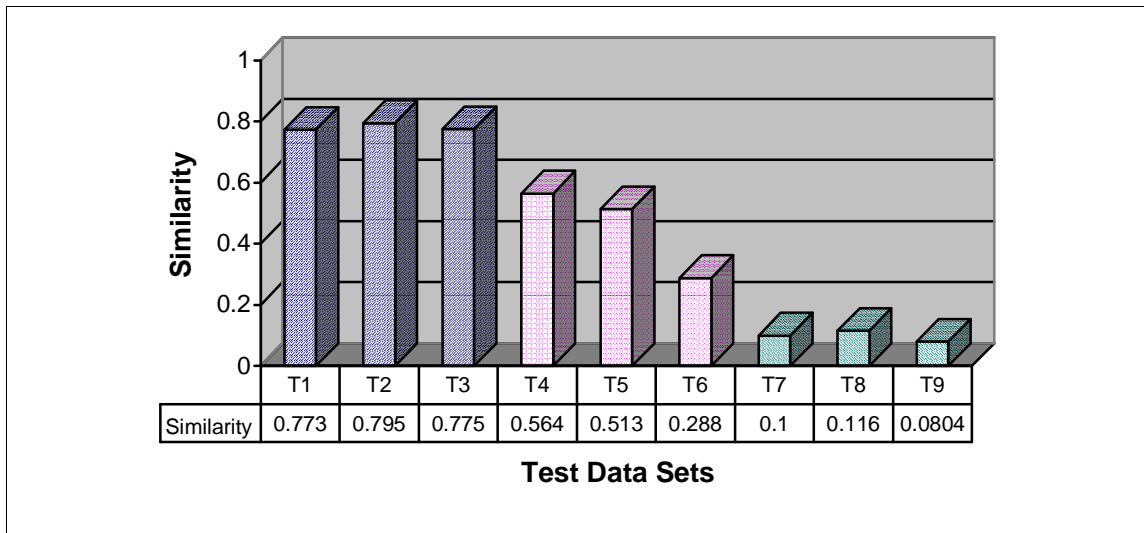


Figure 5: Comparison of Similarities Between 3 Hour Training Data Set and Different Test Data Sets for Fuzzy Association Rules (minconfidence=0.6; minsupport=0.1)

**Training Data Set:** 3 hours of training data (afternoon)

**Test Data Sets:** T1, T2, T3 (afternoon), T4, T5, T6 (evening), and T7, T8, T9 (late night)

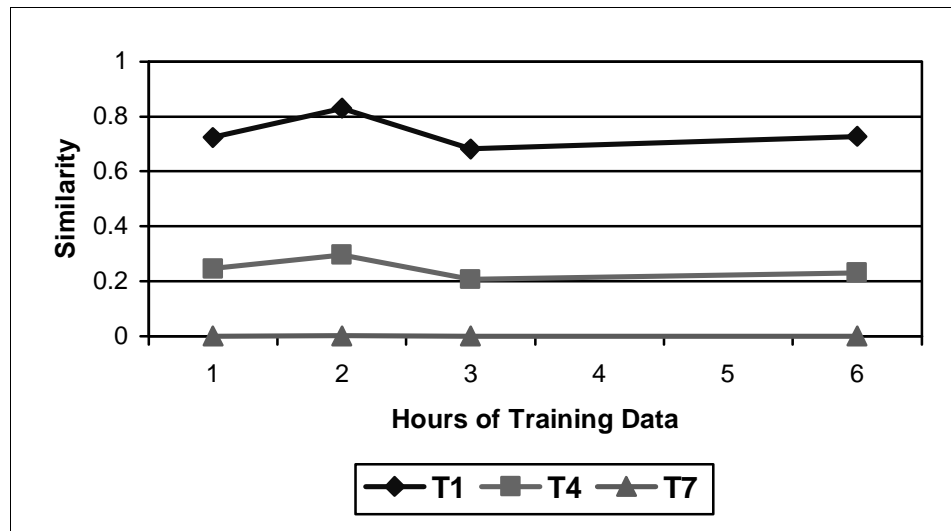


Figure 6: Comparison of Similarities Between Different Training and Test Data Sets for Fuzzy Episode Rules (minconfidence=0.6; minsupport=0.1; minoccurrence=0.3; window=10s)

**Training Data Sets:** 1 hour training data, 2 hours of training data, 3 hours of training data, and 6 hours of training data (all from the afternoon)

**Test Data Sets:** T1 (afternoon), T4 (evening), and T7 (late night)

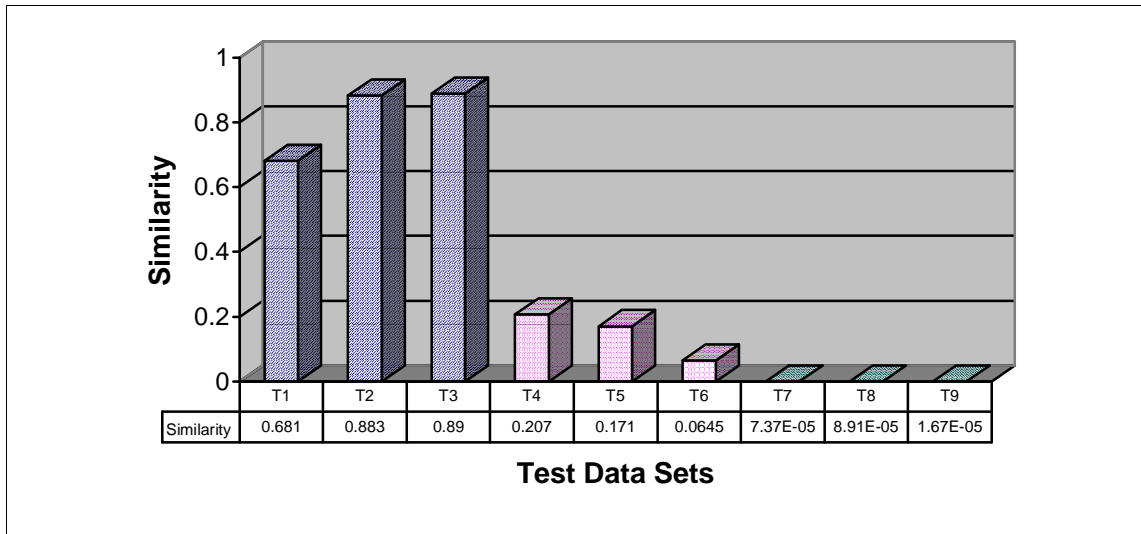


Figure 7: Comparison of Similarities Between Different Training and Test Data Sets for Fuzzy Episode Rules (minconfidence=0.6; minsupport=0.1; minoccurrence=0.3; window=10s)  
**Training Data Sets:** 1 hour training data, 2 hours of training data, 3 hours of training data, and 6 hours of training data (all from the afternoon)  
**Test Data Sets:** T1 (afternoon), T4 (evening), and T7 (late night)

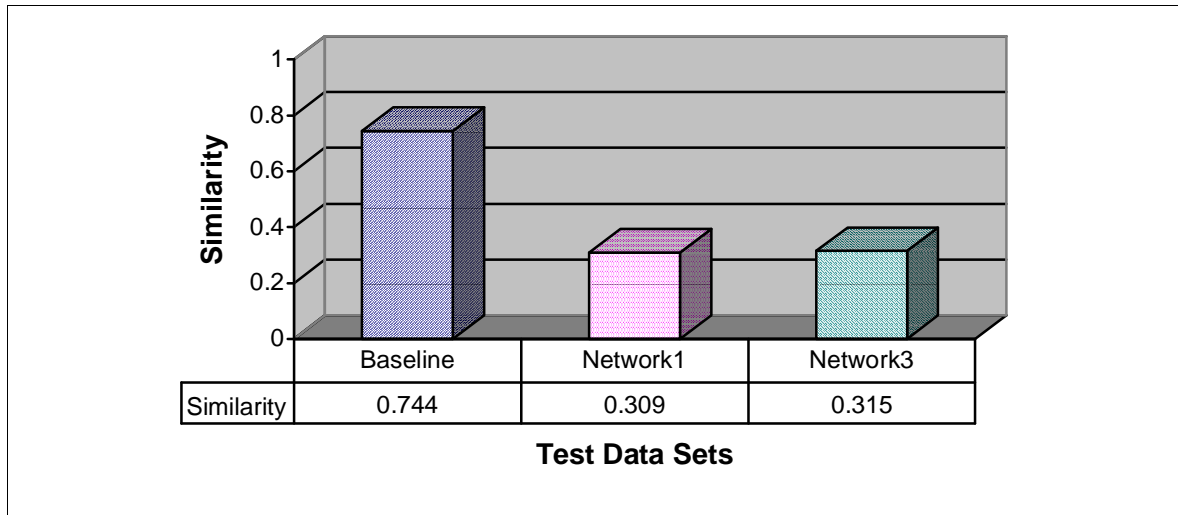


Figure 8: Comparison of Similarities Between Training Data Set and Different Test Data Sets for Fuzzy Association Rules (minconfidence=0.6; minsupport=0.1)  
**Training Data Set:** baseline (first half; representing normal behavior)  
**Test Data Sets:** baseline (second half; representing normal behavior), network1 (including simulated IP spoofing intrusions), and network3 (including simulated port scanning intrusions)

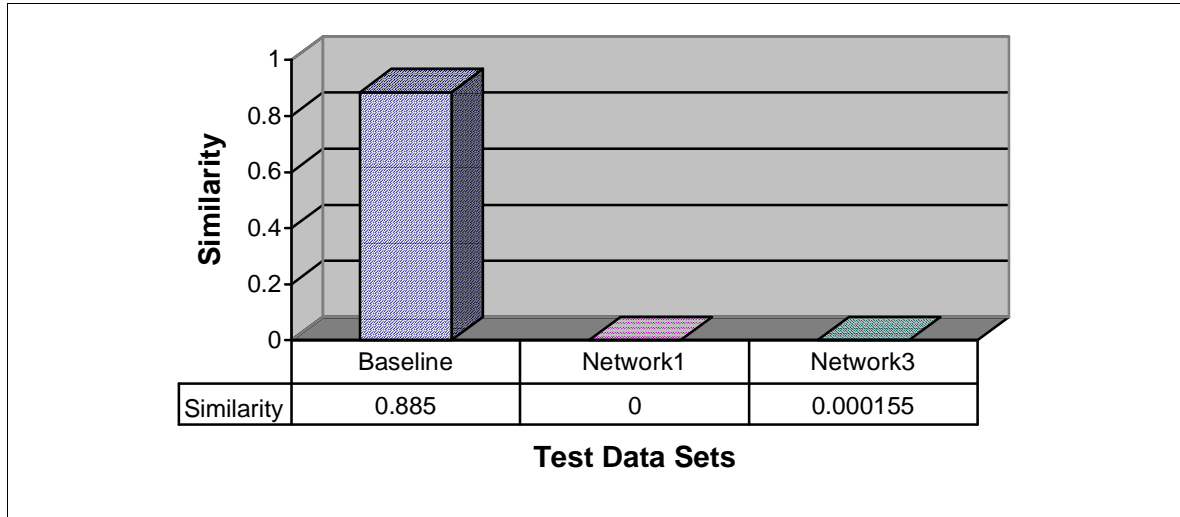


Figure 9: Comparison of Similarities Between Training Data Set and Different Test Data Sets for Fuzzy Episode Rules (minconfidence=0.6; minsupport=0.1; minoccurrence=0.3; window=10s)  
**Training Data Set:** baseline (first half; representing normal behavior)  
**Test Data Sets:** baseline (second half; representing normal behavior), network1 (including simulated IP spoofing intrusions), and network3 (including simulated port scanning intrusions)

Table 1

Specification of Training and Test Data Sets

	<b>Data Sets</b>	<b>Time Slots When Data Sets Are Collected</b>
<b>Training</b>	1 hour of training data	13:00 – 14:00, Tuesday, 23 March 1999
	2 hours of training data	13:00 – 15:00, Tuesday, 23 March 1999
	3 hours of training data	13:00 – 16:00, Tuesday, 23 March 1999
	6 hours of training data	13:00 – 16:00, Friday, 19 March 1999 & 13:00 – 16:00, Tuesday, 23 March 1999
<b>Test</b>	T1	13:00 – 14:00, Wednesday, 24 March 1999
	T2	14:00 – 15:00, Wednesday, 24 March 1999
	T3	15:00 – 16:00, Wednesday, 24 March 1999
	T4	18:00 – 19:00, Tuesday, 23 March 1999
	T5	19:00 – 20:00, Tuesday, 23 March 1999
	T6	20:00 – 21:00, Tuesday, 23 March 1999
	T7	0:00 – 1:00, Wednesday, 24 March 1999
	T8	1:00 – 2:00, Wednesday, 24 March 1999
	T9	2:00 – 3:00, Wednesday, 24 March 1999