Signature-based Approach for Intrusion Detection

Bon K. Sy

Queens College/CUNY
Computer Science Department
Flushing NY 11367
U.S.A.
bon@bunny.cs.qc.edu

Abstract. This research presents a data mining technique for discovering masquerader intrusion. User/system access data are used as a basis for deriving statistically significant event patterns. These patterns could be considered as a user/system access signature. Signature-based approach employs a model discovery technique to derive a reference ground model accounting for the user/system access data. A unique characteristic of this reference ground model is that it captures the statistical characteristics of the access signature, thus providing a basis for reasoning the existence of a security intrusion based on comparing real time access signature with that embedded in the reference ground model. The effectiveness of this approach will be evaluated based on comparative performance using a publicly available data set that contains user masquerade.

1 Introduction

Different kinds of security intrusion could occur in a networked computing environment [1]. For example, network intrusion could be launched via a denial of service attack, while system intrusion in the application layer (or layer 7) could occur through user masquerade. Intrusion prevention involves IT security professions to define security policy rules that can be translated into event patterns that, through real time monitoring, could trigger an alert for a potential intrusion [2, 3].

The challenge for intrusion detection is to develop scalable, extensible data mining techniques that can efficiently examine the audit trials in real time to accurately pinpoint the occurrence of an intrusion. Instead of relying on event patterns that attempt to capture an intrusion, we propose to rely on event patterns that attempt to capture what is expected to be the normal behavior of users and systems. In other words, our research is focused on developing models that signify the access signature as opposed to the intrusion signature. The rationale behind this shift in paradigm is that data are readily available to derive the statistical information about the event patterns, and thus the access signature. On the other hand, significant statistical information from sporadic intrusion activities may hardly be available, if any.

In this research we propose a signature-based approach for discovering masquerader intrusion. Masquerader intrusion refers to an intruder who executes system commands or requests system services under the identity of someone else, often other legitimate user. In our proposed signature-based approach, statistically significant event patterns that characterize the user/system access behavior will be identified based on the concept of association patterns discussed in our previous research [9]. These statistically significant event patterns will be used to define a unique signature about the user/system access behavior. A probability model, referred to as a reference model, preserving the statistical information embedded in the unique signature will then be derived [10]. In the production cycle, statistically significant event patterns will be derived using windowed sequential real time user/system access data, and these event patterns of a windowed sequential data block defines a transitional signature. Inference about the existence of an intrusion will then be based on the degree of statistical deviation as measured by comparing the transitional signature with that embedded in the reference model.

2 Background discussions

The signature based approach for intrusion detection presented in this paper could be considered as a behavior-based approach for statistical anomaly detection; where the essence of the signature based approach is to capture normal behavior ─ as opposed to unusual behavior ─ in terms of signature patterns.

Many different well-known techniques have been proposed for statistical anomaly detection. Many such techniques rely on detecting change point or outlier. One common approach towards change point or outlier detection is to determine how much an observed event (in question) is deviated from some reference “normal event set” using a distance measurement such as L-norm, Hamming distance, Manhattan distance, or vector cosine measure. Another common approach [13] is to determine whether an observed event (in question) appears in the low density regions of the probability distribution characterizing the “normal event set.” More recently, novel approach based on the use of n-gram matching rule [14] for positive and negative detection, as well as hybrid Markov model chain and rarity index model (based on extending the STIDE model) were also proposed [6,7].

In comparison to the existing techniques for statistical anomaly detection, signature based approach presented in this paper is unique in two regards. First, “normal event set” is characterized by a set of statistically significant association patterns referred to as a signature. These statistically significant association patterns bear an important information-theoretic characteristic; namely, frequently co-occurred events in a pattern do not just happen by chance as measured by mutual information criterion. Second, the distance measurement is then conducted under a two-way mutual comparison as opposed to a one-way comparison as typical in standard posterior probability measure. In a one-way comparison, observed event sample is compared against normal event observation. In a two-way comparison, access signature is compared against the observed model (of possible intrusion), and the signature of observed events (of possible intrusion) is compared against the access model to arrive a composite measurement. These are the distinctions of the signature-based approach in comparison to other approaches such as rarity criterion or posteriori probability based matching rule of n-gram samples.

3 Deriving statistically significant patterns

In this research, the type of intrusion we focus on is masquerader intrusion in a Unix/Linux environment; i.e., an intruder injects operating system commands into the shell environment for a command execution under someone else identity. In a Unix/Linux system, "praudit" utility can be installed to keep track of the command execution history of a user [4]. Consider the following example of the command execution history of a user since a successful login session is established: pine, emacs, netscape, ssh, chmod, sftp, javac, java, ….

> *The problem of discovering masquerader intrusion is to determine from the command execution history such as the one shown above whether some command(s) in the command execution history is/are injected by an intruder but not issued by the user who owns the successful login session.*

While it is conceivable to define sequential intrusion patterns as a basis for intrusion detection, there are three fundamental challenges of this approach [5,6,7]. First, the size of the security policy rule set and the corresponding intrusion patterns will grow over time as new intrusion methodologies are discovered. Second, real world intrusion seldom occurs frequent enough to accumulate statistical evidence for timely intrusion detection. Third, it may not be possible to always define security policy rules without causing conflict to what may be an expected acceptable activities. Consider a general security policy: "Change of the file access privilege on the password table should be trapped and interpreted as a potential intrusion," this may be translated to an event trigger defined by "chmod 770 /etc/passwd". Yet such a policy will cause interference on backup/recovery during the regular maintenance process.

To address the limitations just mentioned, we propose a signature concept that attempts to capture the unique characteristic of a legitimate user. The premise of applying the concept of signature is that there exist some unique access patterns of a legitimate user. Imagine in an extreme case where each legitimate user always performs the same activity upon establishing a successful login session; e.g., checking email (using pine), launching emacs to write a report to the supervisor, launching netscape to check company news events, … etc. The likelihood of having two or more users with identical command execution sequence would be very small. Therefore, one may consider the entire command execution sequence of a user as an access signature. This is similar to the idea of *uniqueness* for intrusion detection discussed elsewhere [7]. Obviously defining an access signature based on the entire command execution sequence is unlikely to be computationally manageable. In addition, no user will have the execution sequence completely identical upon different successful login sessions ─ even certain commands or command sequence may always co-occur and appear as association patterns. An alternative approach is to consider categorizing the commands into few categories and to focus on low order association patterns [8] as an access signature. In doing so, it would be relatively more computationally manageable while we try to "optimize" the uniqueness of the access signature of a user.

In this research every Unix/Linux command is categorized into one of the following five groups: (1) Networking, (2) OS/System application/shell script, (3) File access, (4) Security, and (5) Communication.

The example of the command execution history shown earlier "pine, emacs, netscape, ssh, chmod, sftp, javac, java, …." can then be translated into a category sequence “5 3 2 5 4 3 2 2…” Furthermore, the category sequence can be shifted and aligned when considering the low order association event patterns. In this example, shift and alignment for considering 4th order association patterns that accounts for the 4-tuple patterns (x1 x2 x3 x4) of the command execution history will be (x1:5 x2:3 x3:2 x4:5), (x1:3 x2:2 x3:5 x4:4), (x1:2 x2:5 x3:4 x4:3), … etc. In this research, an access signature is defined as the collection of the statistically significant association patterns of 4th order (x1 x2 x3 x4) using the criteria below [9]:

Support measure Pr(x1, x2, x3, x4) ≥ some predefined threshold (1), and

$$MI(x1,x2,x3,x4) \to \left(\frac{1}{\Pr(x1,x2,x3,x4)}\right)\left(\frac{\chi^2}{2N}\right)^{\left(\frac{\hat{E}}{E'}\right)^{0/2}} \qquad (2)$$

where $MI(x1,x2,x3,x4) = Log_2 Pr(x1\ x2\ x3\ x4)/Pr(x1)Pr(x2)Pr(x3)Pr(x4)$

N = sample population size

χ^2 = Pearson chi-square test statistic defined as $(oi - ei)^2/ei$

with *oi* = observed count = *N Pr(x1 x2 x3 x4)*

ei = expected count under the assumption of independence

$\widehat{E}$ = Expected entropy measure of estimated probability model

E’ = Maximum possible entropy of estimated probability model

O = order of the association pattern (i.e., *4* in this case)

The choice of the 4th order association patterns is ad-hoc but under a careful consideration on balancing the representational and computational complexities. Further details about statistically significant patterns could be found in our previous paper [9]. Note that the above two criteria guarantee that any pattern considered statistically significant would have appeared frequently, and the co-occurrence of the associated events in a pattern does not just happen independently and by chance [10].

Since there are only 625 4th order association patterns for five command categories, one could argue that an intruder just has to run commands that belong to the same group as the legitimate user to reduce the chances of detection. This is true under the assumption that the intruder has the prior knowledge about the behavior of the legitimate user. If this is the case, no behavior-based intrusion detection will succeed because the intruder and legitimate user will no longer be distinguishable. And if the intruder is trying to guess the command sequence of the patterns that represent the access signature, there are *C(625,k)* combinations; where *k* is the number of patterns defining the access signature. In this case, we will want to define the time period within which the legitimate user must reveal the access signature, while the likelihood of guessing the correct set of patterns defining the access signature is low.

4 Identifying probability reference model

Referring to the example in the previous section, there are 5^4=625 possible association patterns for *(x1 x2 x3 x4)*. Let’s assume three statistically significant association patterns are found: *(x1:3 x2:2 x3:5 x4:4) (x1:4 x2:3 x3:2 x4:2) (x1:5 x2:4 x3:3 x4:2)*. Let’s further assume the following probability information related to the three significant patterns just shown is available as below:

Pr(x1:3 x2:2 x3:5 x4:4) = 0.03 *Pr (x1:4 x2:3 x3:2 x4:2) = 0.05*

Pr (x1:5 x2:4 x3:3 x4:2)=0.07 Pr(x1:3)=0.15 Pr(x1:4)= 0.37 Pr(x1:5) = 0.23
Pr(x2:2) = 0.13 Pr(x2:3) = 0.35 Pr(x2:4) = 0.14 Pr(x3:2) = 0.27
Pr(x3:3) = 0.17 Pr(x3:5) = 0.3 Pr(x4:2) = 0.45 Pr(x4:4) = 0.12

Note that the degree of freedom of a joint probability model *Pr(x1 x2 x3 x4)* is 5^4 (=625) - 1 – 14 (# of constraints) = 610. Therefore, there are multiple probability models that can satisfy the conditions. The process of model discovery is beyond the scope of this paper. Readers interested in further details are referred to chapter 9 of our book [10]. Nonetheless, we show one such probability model that is locally optimized to minimize the bias to unknown information:

Pr(x1:1 x2:1 x3:2 x4:2) = 0.057272747 Pr(x1:1 x2:2 x3:5 x4:1) = 0.057272717
Pr(x1:1 x2:3 x3:1 x4:1) = 0.12454547 Pr(x1:1 x2:4 x3:5 x4:4) = 0.010909086
Pr(x1:3 x2:1 x3:2 x4:1) = 0.12 Pr(x1:3 x2:2 x3:5 x4:4) = 0.03
Pr(x1:4 x2:1 x3:1 x4:2) = 0.09727271 Pr(x1:4 x2:1 x3:5 x4:1) = 0.026363678
Pr(x1:4 x2:3 x3:2 x4:2) = 0.05 Pr(x1:4 x2:3 x3:5 x4:2) = 0.1754545
Pr(x1:4 x2:4 x3:3 x4:1) = 0.020909093 Pr(x1:5 x2:1 x3:3 x4:4) = 0.079090910
Pr(x1:5 x2:2 x3:2 x4:1) = 0.042727260 Pr(x1:5 x2:4 x3:1 x4:1) = 0.038181823
Pr(x1:5 x2:4 x3:3 x4:2) = 0.07

Where the remaining Pr(x1 x2 x3 x4)s equal to 0.

The significance of an optimal probability model just shown is that it preserves the statistical properties of the significant association patterns while minimizing bias. In other words, the probability information of the model will reveal the statistically significant association patterns that define an access signature. This optimal probability model will be referred to as a <u>reference model</u> for a user.

5 Chi-square goodness of fit for intrusion detection

To determine masquerader intrusion, the command execution history will be examined in a regular time interval. If the command execution history of a user within some time interval could not produce a matching access signature with sufficient confidence level, then it will serve as a basis to suspect the existence of a masquerader intrusion. In the statistical inference framework, Chi-square test statistic λ^2 is used to determine the goodness of fit between the access signature revealed in the command execution history and that in the reference model. Specifically, we test the following null hypothesis versus the alternative hypothesis:

<u>Null Hypothesis:</u>

Masquerader intrusion exists if $\lambda^2 = \sum_{i=1}^{k} (o_i - e_i)^2/e_i > \chi^2_{(1-\alpha,d)}$ where

k is the number of significant patterns $\{ssp_i: i=1 \,..k\}$ revealed in the source model,
N is the size of the command execution history within some given time interval,
$e_i = N\cdot Pr_{source}(ssp_i)$ is the expected count of the i^{th} pattern ssp_i derived from source model,
$o_i = N\cdot Pr_{target}(ssp_i)$ is the observed count of the i^{th} pattern ssp_i derived from target model,
$\chi^2_{(1-\alpha,d)}$ is the value of the Chi-square random variable with a degree of freedom $d = k -1$; where $0 <\alpha < 1$ is the significance level.

<u>Alternative Hypothesis:</u>

Masquerader intrusion does not exist if $\lambda^2 = \sum_{i=1}^{k} (o_i - e_i)^2/e_i \le \chi^2_{(1-\alpha,d)}$

In the formulation just shown, if ssp_is are statistically significant patterns revealed in the reference model, then the source model is the reference model described in the previous section. The target model is then the probability distribution estimated by observing the actual frequency count of the occurrence of the ssp_is in the command execution history within the given time interval. On the other hand, if ssp_is are significant patterns revealed in the data block pertaining to the command execution history, then the source model is the probability distribution estimated from the data block while the target model is the reference model described in the previous section.

In other words, the goodness of fit statistical test will be conducted twice. First, it will be from the perspective of the reference model by comparing the access signature of the data block against its own signature. Second it will be from the perspective of the access data block by comparing the access signature in the reference model against its own signature. By combining the result of the statistical inference tests, a strong conclusion on the existence of masquerader intrusion could be drawn if the null hypothesis survives in both tests, or a weak conclusion on the existence of masquerader intrusion could be drawn if the null hypothesis survives in at least one test.

6 Experimental study and result discussion

The proposed signature-based approach for masquerader intrusion is evaluated using the masquerade data available in the public domain for comparing various intrusion detection methods [11]. The masquerade data consists of 50 groups ─ each group is the access data of one user. Each data group for a user contains 15000 commands. The first 5000 commands within each data group do not contain any masqueraders and are used for the training purposes. The remaining 10000 commands are divided into 100 blocks. Within each block there are 100 commands that are used for testing purposes. The data blocks for testing are seeded with user masquerade that simulates masquerader intrusion. The location of masquerader intrusion is also provided in the masquerade data for evaluation purposes. About 5% (231 out of 5000) of the test data contain masquerades.

The masquerade data also provide additional information. At any given block after the initial 5000 commands a masquerade starts with a probability of 1%. If the previous block was a masquerade, the next block will also be a masquerade with a probability of 80%. However, this information is not used in this experimental study. The following procedure is applied to each one of the 50 users in the study:

Step 1: Reproduce the data column of the first 5000 commands by shifting the data by one time unit. Repeat it 4 times to get a 4-column data table in which each row is a sequential pattern of 4 Unix commands appearing in the command execution history.

Step 2: Extract the first 5000 rows of the four-column data. Apply the function made available elsewhere [10] that implements the two criteria described in section 3 to discover statistically significant patterns. This set of statistically significant patterns defines the access signature of a user.

Step 3: Based on the set of the statistically significant patterns that defines the access signature, define the constraint set similar to that described in section 4.

Step 4: Apply the model discovery utility made available elsewhere [10] to derive the probability reference model for a user. This probability reference model, Pr_{ref}, reveals the access signature and has the (local) minimum biased information.

Step 5: For each one of the 100 test data block that has been enumerated into four-column data, apply the following procedure:

Step 5a: Reproduce the data column of the 100 Unix commands in the block by shifting the data by one time unit. Repeat it four times to get a four-column data table similar to that described in step 1.

Step 5b: Apply the function made available elsewhere [10] that implements the two criteria described in section 3 to discover statistically significant patterns. This set of statistically significant patterns defines the access signature of the test data.

Step 5c: For each statistically significant pattern of a target user, derive the observed count o_i. Likewise, derive the expected count e_i as described in section 5.

Step 5d: Apply statistical inference based on Chi-square goodness of fit as described in section 4 to determine whether masquerader intrusion exists.

Step 6: Derive the correct detection rate, the false positive rate, and the false negative rate based on the result of the 100 test data blocks in step 5.

In order to determine the effectiveness of the approach, Receiver Operating Characteristic (ROC) curve [12] analysis is used to evaluate the result. The followings are the parameters used in a ROC curve analysis:

AP = Actual total positive counts in the test data (masquerader intrusion)
AN = Actual total negative counts in the test data (no masquerader intrusion)
PP = Number of predicted true positive counts
PF = Number of predicted false positive counts
FP = False positive rate = PF/AN
TP = True positive rate = PP/AP

An ROC curve is a graphical plot of FP (X-axis) against TP (Y-axis). Note that both FP and TP are between 0 and 1. An ideal intrusion detector will have a performance where TP = 1 and FP = 0; i.e., every masquerader intrusion is accurately captured with no false alarm. When TP = 1 and FP = 0, it also implies that there is no false negative (since TP = 1) and all negative counts in the test data are correctly concluded by the detection system as no intrusion.

Referring to the threshold value $\chi^2_{(1-\alpha,d)}$ defined in section 4, FP and TP will vary with different choices of α. An ROC curve shows the changes of TP vs. FP as different threshold values are applied. An intrusion/anomaly detector is optimized if its threshold value $\chi^2_{(1-\alpha,d)}$ yields a point (FP, TP) that has the shortest distance to (0,1) in an ROC curve.

An ROC curve is derived for every single user based on the six steps described previously. An ROC curve using all the data ─ referred to as overall ROC ─ is also derived to illustrate the overall performance. In the case of overall ROC, all 5000 blocks (100 blocks for each of the 50 users) are used as testing data. Again, an overall ROC curve is obtained by varying the threshold $\chi^2_{(1-\alpha,d)}$.

Referring to section 4, a Chi-square test statistic $\lambda 1^2$ could be derived by using the training data as the source, and the testing data as the target. Likewise, another Chi-square test statistic $\lambda 2^2$ could be derived by using the testing data as the source, and the training data as the target. We then derive an overall Chi-square test statistic λ^2

based on the linear combination of $\lambda 1^2$ and $\lambda 2^2$; i.e., $\lambda^2 = (1\text{-}w) \cdot \lambda 1^2 + w \cdot \lambda 2^2$. The choice of w varies from 0 to 1 with an increment of 0.1. In applying the statistical inference described in section 4 using the test statistic $\lambda^2 = (1\text{-}w) \cdot \lambda 1^2 + w \cdot \lambda 2^2$, the optimal setting for w is 0.1. Using the test statistic λ^2, the overall ROC curve and the ROC curves for the 50 users (but skipping those with a testing data set that has no intrusion) are shown in Fig.1.

Fig. 2 shows the ROC band envelope that encloses all the ROC curves, the overall ROC curve, and the *estimated* ROC curve. The *estimated* ROC curve is based on "averaging" all ROC curves. Fig. 2 also shows the ROC curves that are one and two standard deviation away from the estimated ROC curve. In Fig. 2, one could note that the ROC band is wide due to a wide variation across all 50 users. Consequently, it is no surprise that the estimated ROC matches closely to the overall ROC curve only partially at FP < 0.2 or FP > 0.8.

Fig. 3 and Fig. 4 show the ROC curves of different selected users. Fig. 5 shows the ROC curve for six different approaches reported elsewhere [7]. Fig. 5 is reproduced for gaining insights into achievable performance. An interesting observation in comparing Fig. 1 and Fig. 3 is that the overall optimal performance for 50 users is better than that for 8 selected users as shown in the corresponding ROC curve. But by comparing Fig. 3 and Fig. 4, the optimal performance for 6 selected users is better than that of all 50 users and 8 selected users. In other words, one must be mindful that performance comparison is only meaningful when the ROC curves generated for different methods are based on the same population of sample users.

One final note about the experimental result is that only normal event/behavior instances are available in the training data set for deriving access signature and reference model. If we are willing to reduce the size of the testing data, it is conceivable to include some of the masquerade intrusion test data as training data to explore the idea of incorporating both access signature and intrusion signature in a reference model. To extend the signature-based approach to incorporate intrusion signature, we only need to modify the statistical hypothesis test by introducing two additional test statistics in additional to $\lambda 1^2$ and $\lambda 2^2$ described earlier to account for the consideration of known intrusion patterns. This additional study will be included in our next report.

7 Conclusion

A signature-based approach is presented for discovering masquerader intrusion. In this proposed approach we introduce the concept of an access signature, which is a collection of statistically significant association patterns. The concept of an access signature is appealing because it allows one to derive a probability model that captures the uniqueness of the access behavior of a user while taking into the consideration of the intra-usage variation. Equally important, the derived probability model provides a basis for detecting masquerader intrusion efficiently. As shown in this paper, efficient detection on masquerader intrusion is simply a process of matching the real time online access signature against the one in the probability reference model based on Chi-square statistical test for goodness of fit. The experimental study also shows an encouraging result in the comparative evaluation. Although we focus on this paper only the masquerader intrusion, the signature-based approach is extensible for incorporating intrusion signatures, as well as for discovering other kinds of intrusion; e.g., network intrusion. This will be the focus of our future research.

Acknowledgement: The author is indebted to the anonymous reviewers who offered many valuable comments that resulted in the improvement of this manuscript. This work is supported in part by a NSF DUE CCLI grant #0088778, and a PSC-CUNY Research Award. Students in the author's Data Mining class of Fall/2004 semester have contributed significantly to the experimental study.

References

1. Sandeep Kumar, *Classification and Detection of Computer Intrusions*, Ph.D. thesis, Purdue University, August 1995.
2. Wenke Lee and Salvaor Srolfo, "Data Mining Approaches for Intrusion Detection," *Proc. of the 7th USENIX Security Symposium*, San Antonio, Texas, Jan., 1998.
3. *etrust Audit: Policy Management Guide 1.5*, Computer Associates, 2003.
4. Sun Microsystems. *SunShield Basic Security Module Guide*.
5. Jeremy Frank, "Artificial Intelligence and Intrusion Detection: Current and Future Directions," June 9, 1994
6. W.-H. Ju and Y. Vardi, "A Hybrid High-order Markov Chain Model for Computer Intrusion Detection," *J. of Computational & Graphical Statistics*, V. 10(2), 2001.
7. M. Schonlau, W. Dumouchel, W.-H., Ju, A.F. Karr, M. Theus, Y. Vardi, "Computer Intrusion: Detecting Masquerades," *Statistical Science*, V. 16, #1, 58-74, 2001.
8. R. Agrawal, T. Imielinski, A. Swami, "Mining Association Rules between Sets of Items in large Databases," *Proc. ACM SIGMOD Conf.*, Washington DC, May 1993.
9. Bon Sy, "Discovering Association Patterns based on Mutual Information," Machine Learning and Data Mining in Pattern Recognition (editor: Petra Perner), Lecture Notes in Artificial Intelligence, Springer-Verlag, July 2003.
10. Bon Sy and Arjun Gupta, *Information-statistical Data Mining: Warehouse Integration with Examples of Oracle Basics*, ISBN 1-4020-7650-9, 2004.
11. http://www.schonlau.net/intrusion.html
12. T. Fawcett, "ROC Graphs: Notes and Practical Considerations for Data Mining Researchers," *Technical Report HPL-2003-4*, Intelligent Enterprise Technologies Laboratory, HP Laboratories Palo Alto, Jan 7, 2003.
13. E. Eskin, "Anomaly Detection over Noisy Data Using Learned Probability Distributions," Proc. of the 17th International Conference on Machine Learning, 2000, pp 255-262, Morgan Kaufmann, San Francisco, CA.
14. F. Esponda, S. Forrest, and P. Helman, "A Formal Framework for Positive and Negative Detection," *IEEE Transactions on Systems, Man and Cybernetics.* 34:1 pp. 357-373 (2004).

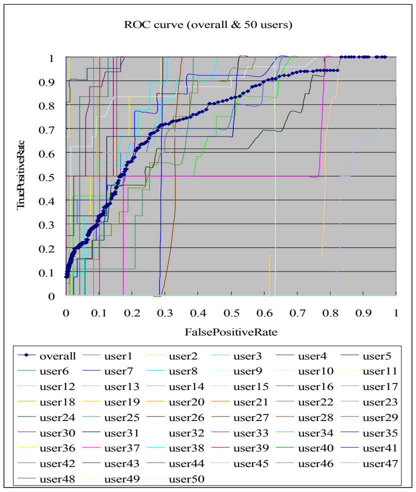

Fig. 1.:ROC curves of all 50 users

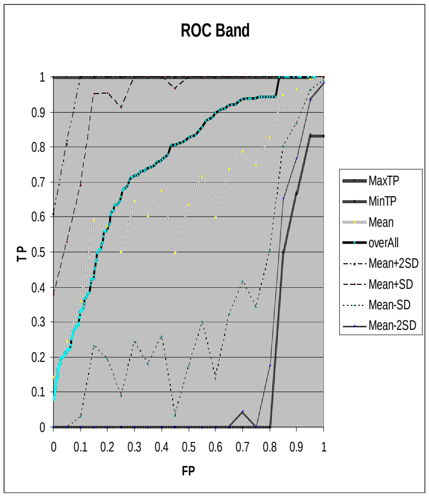

Fig. 2. ROC band and estimated ROC curve

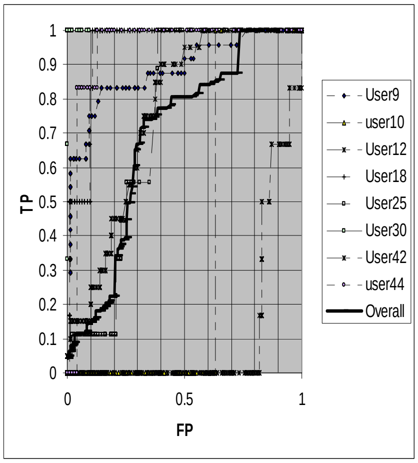

Fig. 3. ROC curve of 8 selected users

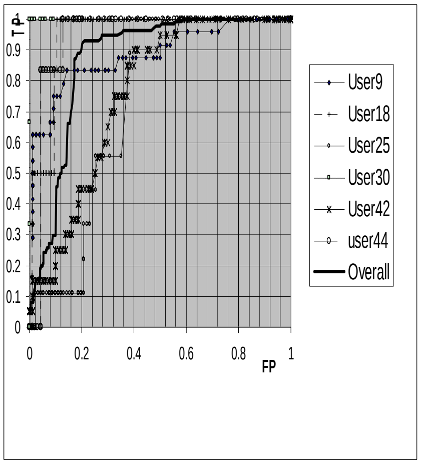

Fig. 4. ROC curve of 6 selected users

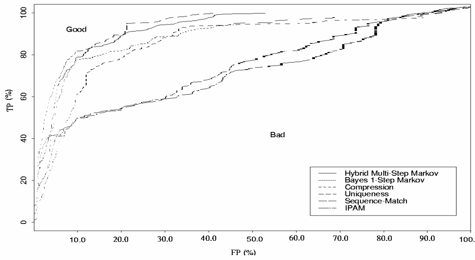

Fig. 5. ROC curves for six different approaches