2003

A Token-based Model for Fraud Detection and Prevention

Yunhua Lu

Leszek T. Lilien

Bharat Bhargava
Purdue University, bb@cs.purdue.edu

Report Number:
03-034

A TOKEN-BASED MODEL FOR FRAUD DETECTION AND PREVENTION

Yunhua Lu
Leszek T. Lilien
Bharat Bhargava

Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

CSD TR #03-034
November 2003

A Token-based Model for Fraud Detection and Prevention

Yunhua Lu, Leszek T. Lilien, Bharat Bhargava
Department of Computer Sciences
Purdue University
West Lafayette, IN 47907, U.S.A.
{luy, llilien, bb}@cs.purdue.edu

Abstract. A token-based model for fraud detection and prevention in information systems is presented. Due to the high false alarm rates experienced in current fraud detection systems, this model has the goal of saving overall system losses. The system raises an alarm only when the token associated with an entity reaches a negative value. It ranks actions (or transactions) by their suspicion level. The most suspicious action will have the lowest token value. This model is more appropriate in evaluating commercial fraud detection and prevention systems by conducting experiments on three types of behavior patterns, intentional cheating, smart repeated cheating, and unintended carelessness. The results show that our model can catch certain repetitive small-cost fraudulent actions which may escape other models. It has a low rate of false alarms and achieves almost optimal decision-making. This mechanism can be easily adopted by current fraud detection and prevention systems.

1. Introduction

Fraud is a deception deliberately practiced in order to secure unfair or unlawful gain [1], or as an intentional perversion of truth in order to induce another to part with something of value or to surrender a legal right [2]. Although cryptographic techniques make fraud more difficult, they cannot eliminate it. As more and more people use online transactions and rely on telecommunications, there has been an increasing interest in developing fraud management engines. For e-commerce, fraud affects less than 2% of consumers; however, it is expected to grow rapidly [3].

Several factors make fraud detection and prevention challenging. First, detection methods effective for one specific domain are high likely less effective for others [4]. Second, for fraudsters and legitimate clients, their behavior patterns change over time. The detection system needs to adapt to new fraud patterns, and it should also recognize new legitimate behaviors. Third, huge volume of involved

information makes it hard to analyze them thoroughly. Fraud needs to be detected in real time to avoid more losses, but overwhelming amount of data makes real-time discovery more difficult. Moreover, the fraud to legitimate ratio is miniscule, sometimes as low as 0.2%. False alarms are unavoidable because most systems can only correctly classify 70% to 80% instances. If every instance is classified as legitimate, in fraud detection area, we could correctly classify 90% instances or even higher. However, this is not the function of fraud detection and prevention. It is difficult to decrease false alarms and also catch more fraud.

Table 1 lists the results from a fraud detection engine. Several criteria have been used in evaluating fraud detection engines. In [5, 6, 7] researchers use Receiver Operating Characteristics (ROC). A ROC graph shows the relationship between True Positive (TP) rate and False Positive (FP) rate. A classifier is defined by its ROC for a dataset, independent of class distribution. Rosset *et al.* [8] use accuracy and fraud coverage as criteria. *Accuracy* is the number of detected fraud over the total number of classified fraud. *Fraud Coverage* is the number of detected fraud over the true number of fraud. It is impossible to know precisely the number of frauds.

Table 1. Fraud Detection Confusion Matrix

	Fraud	Legitimate
Alarm	Correct	False alarm
No-alarm	Missed	Correct

The false alarm rate and the fraud detection rate are more appropriate criteria. The *false alarm rate* is the percentage of false alarm in alarm set. The *fraud detection rate* is the loss by detected fraud over the total loss due to fraud. These two can not be obtained in real time. They may be calculated monthly or weekly in financial area in order to evaluate the performance of fraud detection and prevention engines. For other applications, each compromised state can be associated with a cost, even though patterns of fraud leading to that compromised state are unknown.

False alarms cause the loss of investigators' efforts and a potential loss of clients. A missed fraud has obvious costs. The loss caused by both is called a *system error cost*. A good fraud detection engine should reduce the system error cost.

We propose a token-based model for fraud detection and prevention. Experiments show that it can facilitate the near optimal decision making in alarm generation, and provides a more appropriate metric in evaluating fraud management systems. Proposed design is similar to the earlier proposals [9, 10] in some respects. It has additional advantages of being able to detect the repetitive small-amount frauds that are not caught by the earlier designs and has a lower false alarm rate.

The rest of this paper is organized as follows. Section 2 introduces a state-of-art fraud detection system, followed by token-based fraud detection models in Section 3. Section 4 identifies three types of user behavior. Section 5 describes the experiments and their results. Conclusion is given in Section 6.

2. A State-of-the-Art Fraud Detection System

Fraud detection and prevention systems are widely used in telecommunications [11], online transaction processing [12, 13], and intrusion detection in computers or networks [14, 15]. These systems share the feature that actions are recorded. Based on these recorded data, methods such as data mining [10], machine learning [9, 11] can be applied for fraud detection. Preprocessing is done to select and format the data before using them. Preprocessing can hide private or sensitive information visible in the raw data.

Currently, most fraud detection and prevention systems consist of a profiling engine and a decision-making component.

Profiling Engine. A profiling engine normally involves three major subcomponents: rule generation, user profiling, and online detection. *Rule generation* subcomponent [8, 10, 16, 17] mines massive amounts of database records to get association rules and estimates accuracies of these rules. Both user-level and behavior-level information can be used in mining. Normally, a large volume of fraud rules will be generated.

In the *user profiling* subcomponent, the first step is to select variables that can be used to characterize the range of normal behaviors. The variables should be sensitive to abnormalities and comparable over time. If a fraud happens, at least one

of the patterns for these variables should show an abnormality. The variables can be identified in a number of ways. One approach is to derive them from the fraud rule generation process. The variables will be a subset of those used in the rule set.

Pattern normalization follows variable selection and involves only legitimate behavior data. Patterns include group patterns and individual patterns. *Group pattern normalization* groups clients with similar backgrounds within the period of investigation. It is possible that clients are moved from one group to another over time. Patterns with obvious abnormalities are eliminated from the group pattern normalization.

The normalization process makes a normal distribution of the data for a variable during a short period, such as per twenty four hours. These variable patterns are stored as the user *profile histories* and will be retrieved for reference during online fraud detection. Another set of data is called *current use behavior patterns*. It is similar to the previous set, except that there is no elimination of data. New users have their profiles initiated based on their expected behaviors. For example, it can be based on their user group classification.

The third subcomponent of a profiling engine is *online detection*. When a new action or transaction starts, the detection engine retrieves the related rules from the user profiling subcomponent for the user running the action or transaction. It may need to retrieve user's profile history and his or her current behavior patterns. Each rule is checked. If the action or transaction does not match any fraud rule, the output is zero. Otherwise, the maximum accuracy of the rules that match this action or transaction can be used as the output. This is called *fraud indicator* and denoted "FI".

Decision-making Component. Based on the fraud indicator from the profiling engine, the decision-making component decides if an alarm needs to be generated or not, depending on its decision algorithm. For example, in a cost-based model, an alarm is generated only when the loss from an action or transaction is higher than the cost of investigation.

3. Token-based Fraud Detection Models

The token-based models combine both absolute and differential fraud analysis. *Absolute analysis* uses predefined and static domain-specific rules for fraud

detection. If an action or a transaction matches any of these rules, the system can have 100% confidence that a fraud has happened. For example, in the telecommunications domain, detecting a call collision means that phone is cloned. *Differential analysis* uses generated association rules as described in Section 2. Each rule may have different thresholds for different users. This type of analysis is attractive because it is domain-independent. It can be used to find new fraud patterns. The rule generation subcomponent operates continuously to generate or update rules from newly found frauds and disputable cases, thus producing new sets of rules and their accuracies. The user profiling subcomponent updates the current user behavior patterns and user-level information. The online detection rules are dynamically updated either periodically, or when the rules generated by the rule generation subcomponent have reached a certain predefined level of change defined by the administration. Thus, the token-based model can be easily adopted by current fraud detection and prevention systems. The online detection will adapt rapidly as the other two subcomponents keep adjusting to new patterns.

At the beginning, a system administrator assigns tokens for each participating user. The total number of tokens is system-dependent. The output FI from a profiling engine is used by the decision-making component. FI can be interpreted as the estimate of the probability of fraud given by the profiling engine.

The *risk of loss* R can be defined as

$$R = FI - r \ (0 < r < 50\%)$$

where r is called *risk adjustment parameter*. The higher the sensitivity of a transaction, the lower the risk adjustment parameter. For example, exchanging a good of value $10,000 has higher risk than exchanging a good of value $1 under the same procedure. So the first transaction will have a lower r to make the risk of loss R higher. As another example, intrusion puts the computer system into a higher risk state than Denial of Service, so a low r can make the first action have a higher risk of loss.

Suppose that the expected total benefit from a transaction is B. The token value for the user committing this transaction is updated by the risk of loss R associated with this transaction:

$$token = \begin{cases} token - b \times B \times R & (b < 1) \quad if\ R \le 0 \\ token - d \times B \times R & (d > 1) \quad if\ R > 0 \end{cases}$$

where b and d are called *benefit adjustment parameter and damage adjustment parameter*, respectively. If the risk of loss is not positive, then the token value will increase. The amount of increase depends on the benefit B of this transaction. As the benefit adjustment parameter is less than 1, the increase in token is less than the benefit. If the risk of loss is positive, the token value is decreased. The damage adjustment parameter can be varied, depending on the expected security level of the system. By awarding token conservatively with good behaviors and taking away token aggressively with bad behaviors, a system can achieve nearly optimal decision making.

If the token owned by an account becomes negative, the system generates an alarm and outputs the negative token value. As the output is ordered, the higher the fraud cost, the lower the token value. According to this order, investigators can give priorities to their investigations.

4. Types of User Behavior

Based on three categories on fraud indicators (figure 1), we define three types of user behaviors [18] and use them as the input in the experiments. They are: intentional cheating, smart repeated cheating, and unintended carelessness. In each category, fraud indicator has a normal distribution, that is FI ~ N (μ, σ^2) with mean of μ and standard deviation of σ.

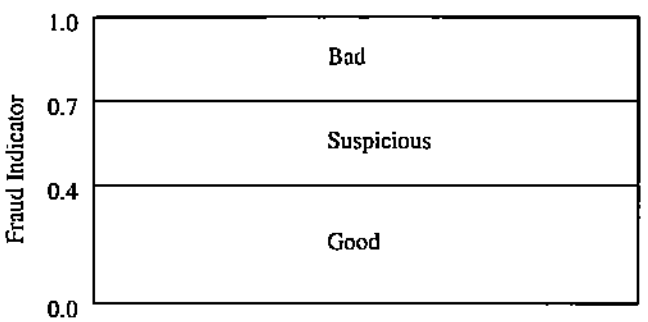


Figure 1. Categorization based on fraud indicator.

Intended cheating means that a user makes an attempt to cheat. The value of fraud indicator fluctuates in the good domain without causing the system alarm, and then suddenly moves toward bad domain. Figure 2 is an example behavior for this model. The fraud detection engine shows the following value of fraud indicator for the user of these 120 transactions:

$$Fraud\ Indicator = \begin{cases} 0.85 & if\ sequence\ number\ is\ 30 \\ 0.9 & if\ sequence\ number\ is\ 70 \\ 0.78 & if\ sequence\ number\ is\ 100 \\ N(0.2, 0.05^2) & otherwise \end{cases}$$

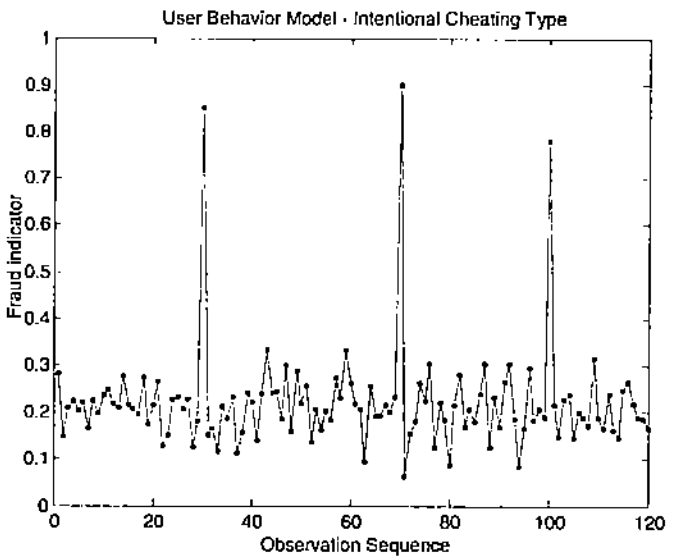


Figure 2. Intentional cheating.

Smart repeated cheating is the behavior model for a user that has repetitive small-cost cheating actions with the intension of avoiding being caught by the fraud detection system. The user constrains his or her actions in suspicious area and commits no bad behaviors. Figure 3 shows an example of smart repeated cheating type. The system shows the fraud indicator of the user behaviors has a mean of 55% and a standard deviation of 2%.

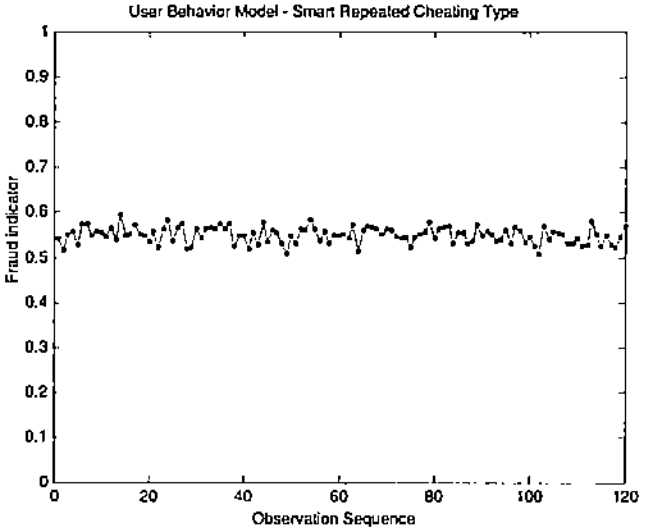


Figure 3. Smart repeated cheating.

Unintended carelessness is the behavior model for a user that does not intend to cheat. But occasionally either due to the careless of an entity or because of limitations of the expert system, the system may indicate a little high possibility of fraud. Figure 4 shows an example of unintended carelessness. For the 120 sequential transactions, the fraud detection engine indicates that the fraud indicators follow:

$$Fraud\ Indicator = \begin{cases} 0.65 & if\ sequence\ number\ is\ 31 \\ 0.55 & if\ sequence\ number\ is\ 62 \\ 0.63 & if\ sequence\ number\ is\ 93 \\ N(0.2, 0.05^2) & otherwise \end{cases}$$

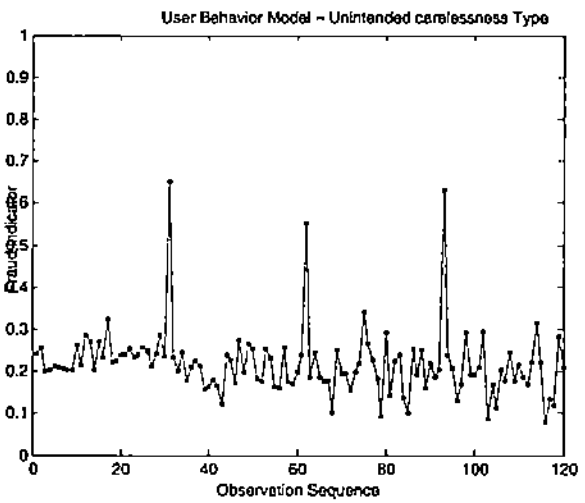


Figure 4. Unintended carelessness.

5. Experimental Results and Evaluation

We assume that the threshold for cost-based model is 1. When a cost is above the threshold, an alarm will be generated in this model. For simplicity, we assume that each transaction has the same benefit B, which may not be true in reality.

Table 2 describes the input simulation parameters and their values. We run the experiment using Matlab.

Table 2 Simulation Parameters

Benefits	b	d	r	Token
1.6	0.01	1.5	0.5	0.5

Figure 5 shows the comparison of the results from a cost-based decision making model and a token-based model for the intentional cheating behavior models of Section 3. Both models catch these anomalous behaviors and trigger alarms. There is no time delay in our model comparing with the other.

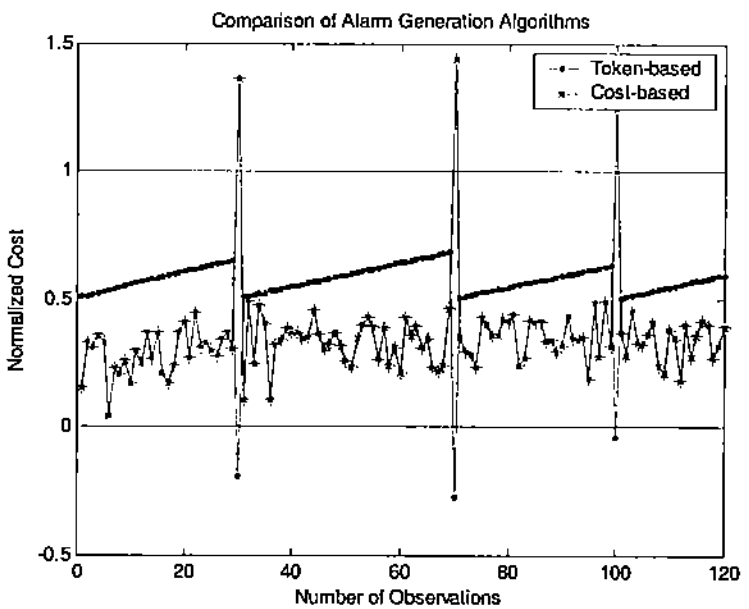


Figure 5. Decision making model comparison for intentional cheating.

If someone figures out the threshold of triggering an alarm, he or she may make smart small-cost repeated cheatings and can successfully avoid being caught under the cost-based model. But the token-based model will detect it, as the token will go down the threshold and an alarm will be generated. Figure 6 shows the results of two models for the smart repeated cheating behavior of Figure 3.

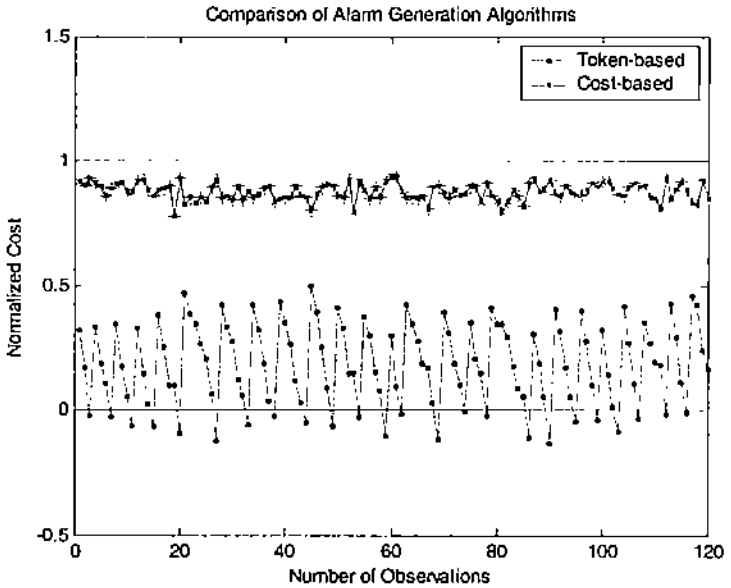


Figure 6. Decision making model comparison for smart repeated cheating.

Figure 7 shows token-based model can reduce false alarms under certain conditions. For the instance of unintended carelessness type in figure 4, we will use the entity's token to offset that entity's carelessness or shortcomings of profiling engine. If the adjusted token value is still above threshold, an alarm will not be generated. Cost-based model triggers two times in figure 7.

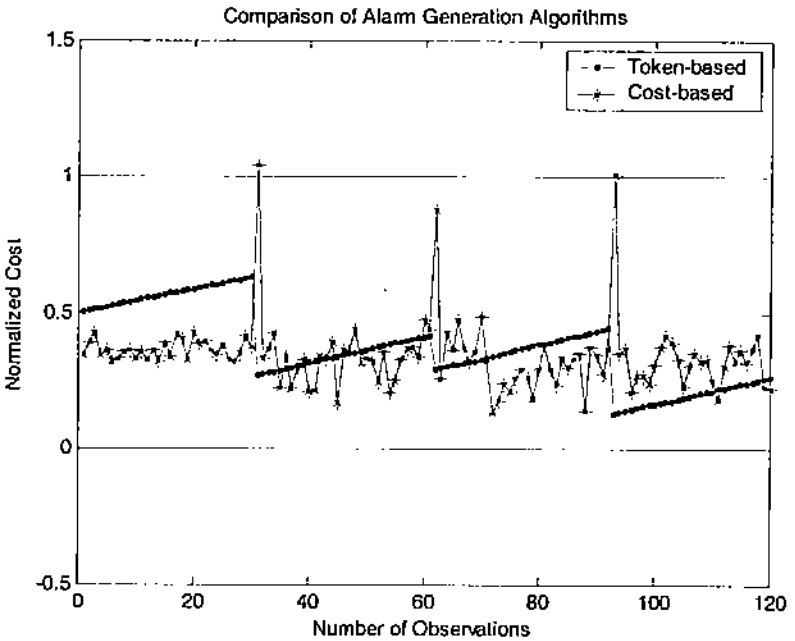


Figure 7. Decision making model comparison for unintended carelessness.

Next, we simulate each type on 1000 observation sequences. For Intentional cheating, 97% time the fraud indicator is within good area, with 3% time in bad area. For smart repeated cheating, the fraud indicator is within suspicious area. For unintended-carelessness, 97% time the fraud indicator is within good area, 3% time in suspicious area. The other parameter is the same in Table 2. We use the standard deviation of 5%. For smart repeated cheating, as the fraudster tries to control his or her activity, the standard deviation is less than this. We set it as 2%. Figure 8-10 shows token-based model generate a little small alarm number than cost-based model for intentional cheating type and unintended carelessness type. However, it catches much more in smart repeated cheating type.

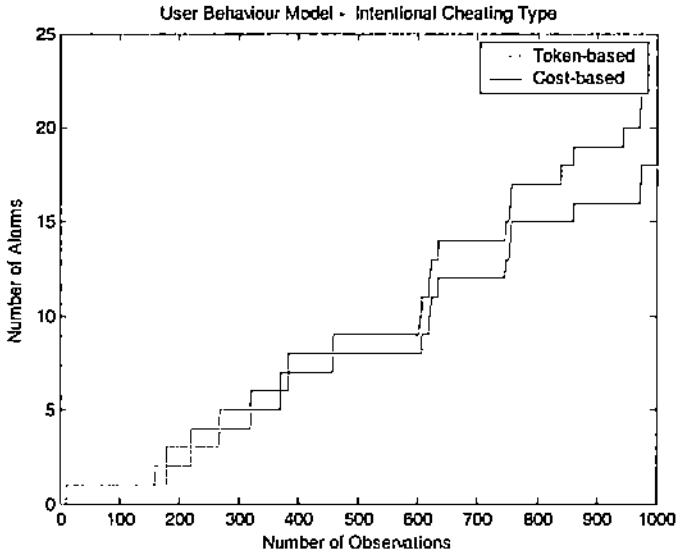


Figure 8. False alarm number in intentional cheating type.

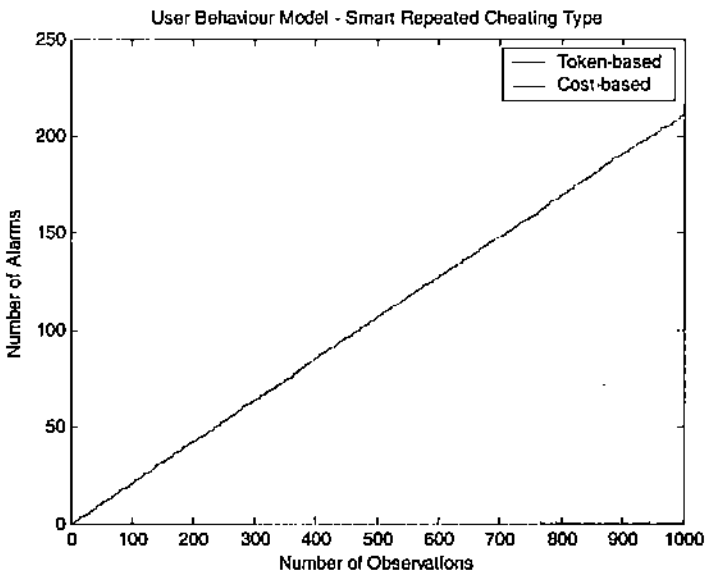


Figure 9. False alarm number in smart repeated cheating type.

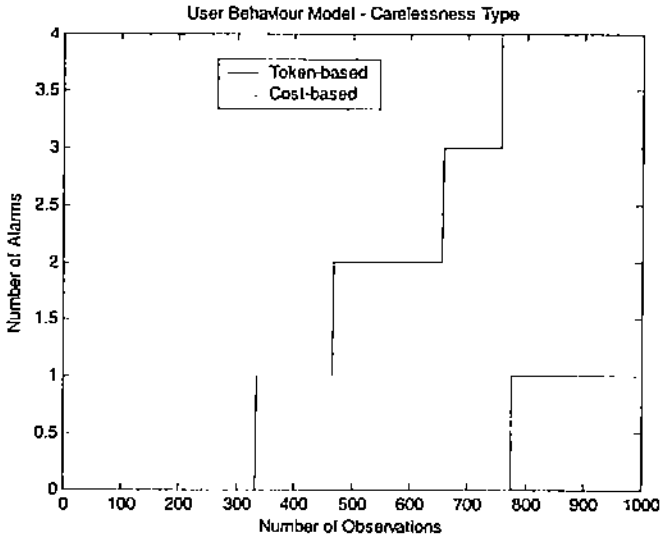
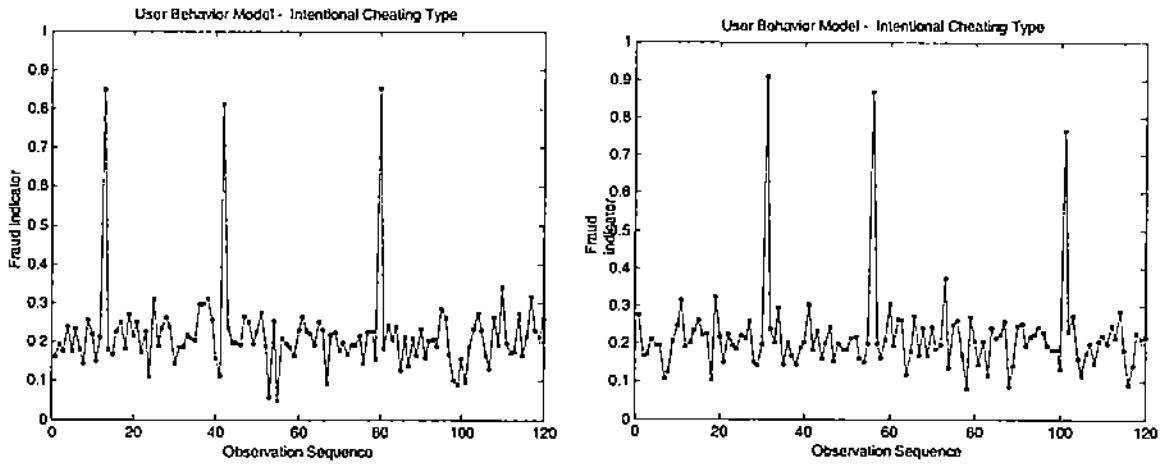


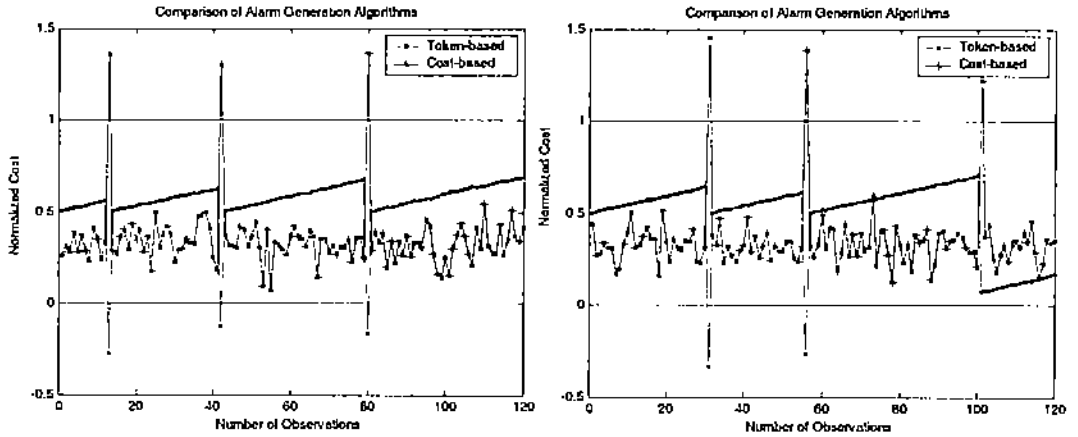
Figure 10. False alarm number in unintended carelessness type.

Figure 12 shows the results of two instances in Figure 11 for intentional cheating type. The most common one is instance a, it is similar as Figure 2. Missing alarm is still possible in token-based model, as shows in instance of b. However, this will not affect the functions of the fraud detection system. As the overall benefit from missed-alarm entities is still positive, the overall system benefit is positive. Also, as most entities are well behaviors, missed alarms from cheating entities are quiet small.



(a) Intentional cheating instance (b) Intentional cheating instance

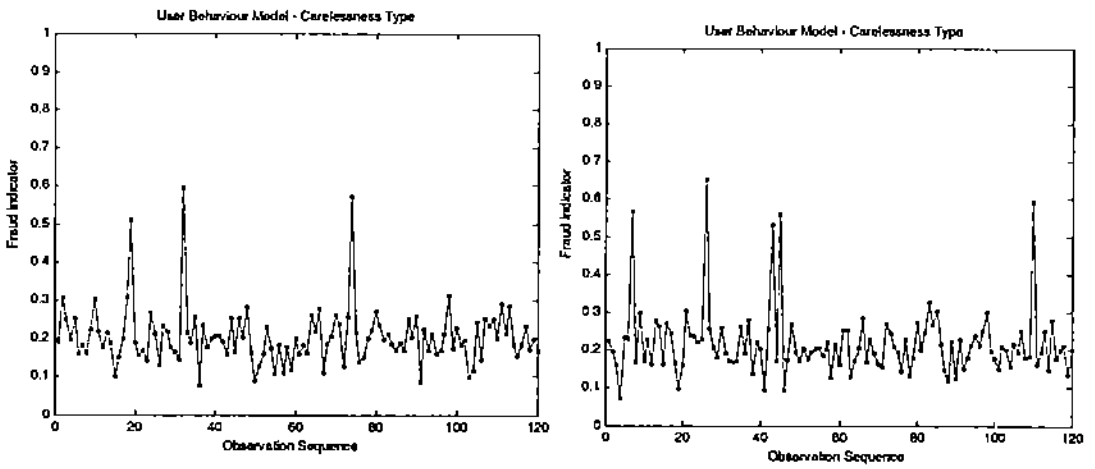
Figure 11. Intentional cheating.



(a) Result for instance a (b) Result for instance b

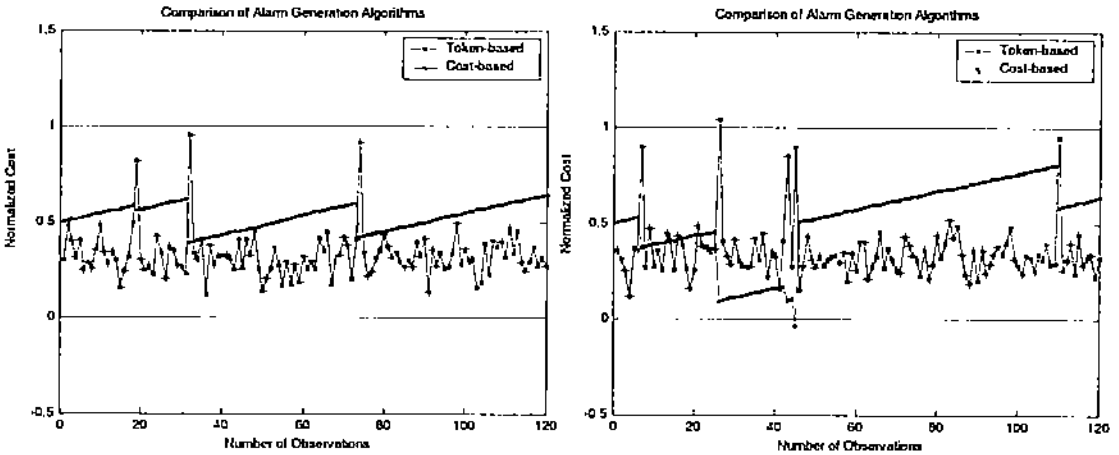
Figure 12. Results of intentional cheating.

There is no difference in smart repeated cheating type as previous result. Besides the instance as we show before, there are two additional instances we observed for unintended carelessness type, as show in Figure 13. Figure 14 is the results for them. In instance a, both models tolerate suspicious activities. In instance b, the token-based generates alarm when several suspicious activities observed.



(a) Unintended carelessness instance (b) Unintended carelessness instance

Figure 13. Unintended carelessness.



(a) Result for instance a (b) Result for instance b

Figure 14. Result of unintended carelessness.

6. Conclusion

We present a token-based model for fraud detection and prevention in real-time. By combining absolute analysis and differential analysis, we propose to use tokens in order to save overall system cost and reduce false alarm rate without comprising the function of fraud detection system. This model achieves almost optimal decision-making in alarm generation. This mechanism can also catch small-cost repetitive fraudulent activities. The system raises alarm only when the token associated with an entity reaches a negative value. Our experiments demonstrate that token-based model is more appropriate for fraud detection and prevention than cost-based models. This model can be easily adopted by current fraud detection systems.

7. References

[1] The American Heritage Dictionary of the English Language, Fourth Edition, Houghton Mifflin, 2000.

[2] Merriam-Webster's Collegiate Dictionary, Eleventh Edition, 2003.

[3] Clara Centeno. (April, 2002). Building Security and Consumer Trust in Internet Payments/The potential of "soft" measures. Available: http://epso.jrc.es/Docs/Backgrnd-7.pdf

[4] J. Hollmen, "User profiling and classification for fraud detection in mobile communications networks," Ph.D. thesis, Dec. 2000.

[5] Y. Moreau, H. Verrelst and J. Vandewalle, "Detection of mobile phone fraud using supervised neural networks: a first prototype," in International Conference on Artificial Neural Networks, 1997.

[6] M. Taniguchi, M. Haft, J. Hollmén, and V. Tresp, "Fraud detection in communications networks using neural and probabilistic methods," in *Proc.* of the

1998 IEEE International Conference in Acoustics, Speech and Signal Processing, 1998.

[7] J. Hollmén and V. Tresp, "Call-based fraud detection in mobile communication networks using a hierarchical regime-switching model," in *Proc.* of Advances in Neural Information Processing Systems 11(NIPS'11), 1998.

[8] S. Rosset, U. Murad, E. Neumann, Y. Idan and G. Pinkas, "Discovery of fraud rules for telecommunications - challenges and solutions," in *Proc.* of the 5th ACM SIGKDD International Conference of Knowledge Discovery and Data Mining, 1999.

[9] S. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. Chan, "Cost-based modeling for fraud and intrusion detection: results from the JAM project," in *Proc.* of the 2000 DARPA Information Survivability Conference and Exposition (DISCEX '00), 2000.

[10] P. Chan, W. Fan, A. Prodromidis, and S. Stolfo, "Distributed data mining in credit card fraud detection," in IEEE Intelligent Systems, pp 67-74, 1999.

[11] P. Burge and J. Shawe-Taylor, "Detecting cellular fraud using adaptive prototypes," in *Proc.* of AAAI-97 Workshop on AI Approaches to Fraud Detection and Risk Management, 1997.

[12] Bolton, R. J. and D. J. Hand, "Unsupervised profiling methods for fraud detection," Credit Scoring and Credit Control VII, Edinburgh, UK, 2001.

[13] ClearCommerce Corporation. (April, 2001). Fraud Prevention Guide for Online Merchants. Available: www.clearcommerce.com/pdf/whitepapers/ClearCommerce_Fraud_Prevention_White_Paper.pdf.

[14] M. Cahill, F. Chen, D. Lambert, J. Pinheiro and D. Sun, "Detecting fraud in the real world," in *Handbook of Massive Datasets*, Klewer Academic Publishers, pp. 911- 930, 2002.

[15] Bolton, R. J. and D. J. Hand, "Statistical Fraud Detection: A Review (with discussion)," *Statistical Science*, pp 235-255, 2002.

[16] T. Fawcett and F. Provost, "Adaptive fraud detection," in *Data Mining and Knowledge Discovery*, pp 291- 316, 1997.

[17] R. Brause, T. Langsdorf and M. Hepp, "Neural data mining for credit card fraud detection," in *Proc.* of the 11th IEEE International Conference on Tools with Artificial Intelligence, 1999.

[18] B. Bhargava, Y. Zhong, and Y. Lu, "Fraud formalization and detection," in DaWak 2003.