2003

Fraud Formalization and Detection

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

Yuhui Zhong

Yuhua Lu

Report Number:
03-008

FRAUD FORMALIZATION AND DETECTION

Bharat Bhargava
Yuhui Zhong
Yuhua Lu

Computer Science Department
Purdue University
West Lafayette, IN 47907

CSD TR #03-008
April 2003

Fraud Formalization and Detection[*]

Bharat Bhargava, Yuhui Zhong, Yuhua Lu

Center for Education and Research in Information Assurance and Security
And
Department of Computer Sciences
Purdue University
West Lafayette, IN 47907, U.S.A
{bb, zhong, luy}@cs.purdue.edu

Abstract. A fraudster can be an impersonator or a swindler. An impersonator is an illegitimate user who steals resources from the victims by "taking over" their accounts. A swindler is a legitimate user who intentionally harms the system or other users by deception. Previous research efforts in fraud detection concentrate on identifying frauds caused by impersonators. Detecting frauds conducted by swindlers is a challenging issue. In this paper, three types of deceiving intentions, namely uncovered deceiving intention, trapping intention, and illusive intention, are defined. We propose an architecture that integrates deceiving intention prediction with fraud detection to catch swindlers. It consists of four components: profile-based anomaly detector, state transition analysis, deceiving intention predictor, and decision-making component. Profile-based anomaly detector outputs fraud confidence indicating the possibility of fraud when there is a sharp deviation from usual patterns. State transition analysis provides state description to users when an activity results in entering a danger state leading to fraud. Deceiving intention predictor discovers malicious intentions. DI-confidence is used to characterize the belief that a target entity has such intentions. An algorithm is developed to evaluate DI-confidence by analyzing an entity's behaviors. Its effectiveness is investigated via experimental study. A user-configurable risk evaluation function is designed for decision-making component. The decision-making component raises a fraud alarm when expected risk is greater than fraud-investigating cost.

1.Introduction

Fraudsters can be classified into two categories: impersonators and swindlers. An impersonator is an illegitimate user who steals resources from the victims by "taking over" their accounts. Taking superimposed fraud in telecommunication [9] as an example, impersonators gain access to the accounts of legitimate users by stealing their Mobile Identification Numbers (MIN) and Equipment Serial Numbers (ESN) and producing cloned phones. The abnormal usage is imposed on the victims. All subsequent bills will go to their accounts. A swindler, on the other hand, is a legitimate user who intentionally harms the system or other users by deception. For instance, subscription fraud conducted by swindlers is recognized as one of the major frauds in telecommunication today. The fraudsters obtain legitimate accounts and use the services without intention to pay the bills.

Impersonators can be forestalled by utilizing cryptograph technologies that provide strong protection to users' authentication information. The idea of separation of duty may be applied to reduce the impact of a swindler. The essence of separation of duty is to restrict the power an entity (e.g., a transaction partner) can have to prevent him from abusing it. An empirical example of this idea is that laws are set, enforced and interpreted by different parties. Separation of duty can be implemented by using access control mechanisms such as role based access control mechanism, lattice-based access control model [13]. Duties are separated statically or dynamically. Suppose a set of operations {O1, O2, ...On} need to be executed in order to complete a transaction. Static separation of duty assigns the execution privileges to entities in such a way that nobody is allowed to execute all operations. Dynamic separation of duty imposes no constraint on execution privileges. An entity is allowed to carry out any potential operation. However, the possibility of executing some (conflict) operations is automatically ruled out when an operation is performed. The

[*] This research is supported by NSF grant IIS-0209059

Chinese wall policy arising in the commercial sector of consulting services is an example of dynamic separation of duty.

Separation of duty policy and other mechanisms like dual-log bookkeeping prevent frauds but cannot eliminate them. For example, for online auctions, such as eBay, sellers or buyers have restricted knowledge about the other side. Although eBay, as a trusted third party, has authentication services to check the information of sellers and buyers (e.g. phone number, credit card number etc), it is impossible to verify all of them due to the high quantities of online transactions. Fraud is a persistent issue under such an environment.

In this paper, we concentrate on swindler detection. Three approaches are considered: (1) detecting an entity's activities that deviate from normal patterns; (2) Constructing state transition graphs for existing fraud scenarios and detecting frauds similar to the known ones (3) discovering an entity's intention based on his past behaviours. The first two approaches are also used to detect frauds conducted by impersonators. The last one is only applicable for swindler detection. An architecture utilizing all three approaches is proposed.

The rest of this paper is organized as the follows. Section 2 introduces the related work. Formal definitions related to fraud and deceiving intentions are presented in section 3. An architecture for swindler detection is proposed in section 4. It consists of four components: profile-based anomaly detector, state transition analysis, deceiving intention predictor, and decision-making. The functionalities and design considerations for each component are discussed. The algorithm for deceiving intention prediction and experiments testing it are presented. The conclusion and future work is in section 5.

2.Related work

Fraud detection systems are widely used in telecommunication, online transactions, computer and network security, and insurance. The majority of research efforts addresses superimposed fraud detection in telecommunications [2][3][10][11]. When a phone call is made, a toll ticket is generated to record information, such as the time, duration and destination, from the call. Data mining [4][5][7], machine learning [1][2] and statistical methods [3][11] have been developed to detect fraud on recorded data. Effective fraud detection uses both fraud rules and pattern analysis.

Fawcett and Provost [4] presented an adaptive rule-based detection framework for superimposition fraud. First, fraud patterns are encoded in the form of classification rules. Each account is profiled during non-fraud periods to determine normal user behavior patterns. After the profiling step, online monitors check each transaction against established patterns to indicate anomalies. The system combines the outputs of monitors to decide if it will raise an alarm.

Burge and Shawe-Taylor [3] developed a neural network technique for fraud detection. The probability distributions of current behavior profiles and behavior profile histories are compared using Hellinger distances. Larger distances mean more suspicion of fraud.

Rosset et al. [9] pointed out that standard classification and rule generation is not appropriate for fraud detection. In rule-based fraud detection, the generation and selection rule set should combine both user-level and behavior-level attributes. The rule set should have high coverage of fraud cases, be accurate, have quick triggering, and the set size should be small.

Chan et al. [8] proposed two approaches solving incompatible schema problem caused by distributed database sharing during fraud detection.

Due to the skewed distribution of fraud, one challenge in fraud detection is a very high false alarm rate. Several criteria exist to evaluate the performance of fraud detection engines. ROC [1][11][12]is a common one. Rosset et al. [9] uses accuracy and fraud coverage as criteria. Accuracy is the number of detected instances of fraud over the total number of classified frauds. Fraud coverage is the number of detected frauds over the total number of frauds. It is difficult to know exactly the total number of frauds. Stolfo et al. [6] use a cost-based metric in commercial fraud detection systems. They proposed that if a fraud causes loss less than the cost for the investigation, then this fraud should be ignored. However, if such fraud happens frequently enough, the accumulated loss will be significant, and the cost-based model can't detect it.

Our work focuses on detecting fraud from swindlers. In the rest of the paper, fraud refers to the cheating actions from swindlers.

3. Formal definitions

Fraud occurs when two or more entities cooperate to fulfil a task. When a user subscribes the telecommunication services from a specific company, the user and the company establish a cooperative relationship. A transaction between buyer and seller on eBay is another example of cooperation. Each entity commits to his partner to satisfy certain terms, conditions, integrity constraints and assumptions. A swindler is an entity that has no intention to keep one's commitment.

- *Commitment:* The integrity constraints, assumptions and conditions an entity promises to satisfy in cooperation. Commitment is described by using a set conjunction of expressions. An expression could be (a) an equality between a variable and constant, or (b) a user-defined predicate. For the first case, the variable in an equality represents a feature used in the commitment while the constant representing the expected values of the feature. For the second case, a user-defined predicate represents certain complex constraints, assumptions and conditions. An associated user-defined boolean function checks whether the constraints etc. hold.

  Example: A commitment of a seller for selling a vase is

  $(\text{Received\_by} = 04/01) \wedge (\text{Prize} = \$1000) \wedge (\text{Quality} = A) \wedge \text{ReturnIfAnyQualityProblem}$

  This commitment says that the seller promises to send out one "A" quality vase at the price of $1000. The vase should be received by Apr. 01. If there is quality problem, the buyer can return the vase.

- *Outcome:* The actual results after execution. For each expression in a commitment, there is an expression in an outcome corresponding to it. For an equality expression, the constant indicates the actual value of the feature. For a predicate expression in a commitment, if the use-define function is evaluated to be true, the predicate itself is included in the outcome. Otherwise, the negation of the predicator is included.

  Example: An outcome corresponds to the above commitment can be

  $(\text{Received\_by} = 04/05) \wedge (\text{Prize} = \$1000) \wedge (\text{Quality} = B) \wedge \neg\text{ReturnIfAnyQualityProblem}$

  This outcome shows that the vase was received at Apr. 05. However, the quality was B. The return request is refused. In this case, we may conclude that the seller is a swindler.

Predicates and feature variables play different roles in detecting whether an entity is a swindler or not. We define two properties, namely *intention-testifying* and *intention-dependent*.

Predicator:

- *Intention testifying predicate:* A predicate P is intention-testifying if we can conclude a cooperation partner is a swindler given the fact that $\neg P$ appears in an outcome.
- *Intention dependent predicate:* A predicate P is intention-dependent if it is possible that a cooperation partner is a swindler given the fact that $\neg P$ appears in an outcome.

In the above example, ReturnIfAnyQualityProblem can be an intention testifying or an intention dependent predicate. The decision is up to the user.

Feature variable:

Suppose the expected value of a feature variable V is E in a commitment. Its actual value is A in an outcome. The domain of the feature variable is totally ordered by a *dominant* relationship. If α *dominate* β, α is more desirable than β.

- *Intention testifying feature variable:* The feature variable V is intention-testifying if we can conclude a cooperation partner is a swindler given the fact that E *dominate* A.
- *Intention dependent feature variable:* The feature variable V is intention-dependent if it is possible that a cooperation partner is a swindler given the fact that that E *dominate* A.

In the above example, *prize* is an intention testifying feature variable. The *dominant* relationship is defined by natural "<". If the seller charges more money, we conclude that he is a swindler. *Quality* and *received_by* can be defined as intention dependent feature variables considering the fact that a seller himself may not have full control on them.

An intention testifying feature variable/predicate is intention dependent. The opposite direction is not necessarily true: An intention dependent feature variable/predicate may not be intention testifying.

3.1 Deceiving intentions

The intention testifying is so strong that few such variables/predicates are available in real applications. Usually, variables/predicates are specified as intention dependent. A conclusion that a partner is a swindler cannot be drawn with certainty based on one intention dependent variable/predicate in one outcome. Two approaches can be used to increase the confidence: (1) consider multiple variables/predicates in one outcome; (2) consider one variable/predicate in multiple outcomes. The second approach is used in this paper.

We assume that a user is given a satisfied rating ranging from 0 to 1 for the actual value of an intention dependent variable in an outcome. The higher the rating is, the more satisfied the user is. The value of 0 indicates totally unacceptable while the value of 1 indicates that actual value is not worse than the expected value. For example, if the quality of received vase is B, the rating can be 0.5. If the quality is C, the rating may drop to 0.2. For an intention dependent predicator, the rating is either 0 or 1. In this section, three types of deceiving intentions are identified.

A satisfied rating is related to an entity's deceiving intention as well as some unpredictable factors. It is modelled by using random variables with normal distribution. The mean function f_m (n) determines the mean value of the normal distribution at the n^{th} rating.

- *Uncovered deceiving intention:* For a swindler with uncovered deceiving intention, the satisfied ratings associated with him are stably low. The ratings vary in a small range over time. We can think that all ratings obey the same normal distribution. The mean function is defined as f_m (n) = m, where m is a constant. Figure 1 shows satisfied ratings with f_m (t)=0.8. The fluctuation of ratings results from the unpredictable factors.
- *Trapping intention*: The rating sequence can be divided into two phases: preparing and trapping. An entity behaves well in the preparing phase to achieve a trustworthy image. Then, he conducts fraud. The mean function can be defined as:

$$f_m(n)=\begin{cases} m_{high} & n \le n_0 \\ m_{low} & otherwise \end{cases}$$ Where n_0 is the turning point.

Figure 2 shows satisfied ratings for a swindler with trapping intention. For the first 50 interactions, f_m (n) is 0.8 and 0.2 afterwards.

- *Illusive intention*: For a smart swindler with illusive intention, instead of misbehaving continuously, he attempts to cover the bad effects by intentionally doing something good after misbehaviours. He repeats the process of preparing and trapping. Fm (t) is a periodic function. For simplicity, we assume the period is N, the mean function is defined as:

$$f_m(n)=\begin{cases} m_{high} & (n \bmod N) < n_0 \\ m_{low} & otherwise \end{cases}$$

Figure 3 shows satisfied ratings with period of 20. In the first 15 interactions of each period, f_m (t) is 0.8. For last 5 interactions, f_m (t) drops to 0.2.

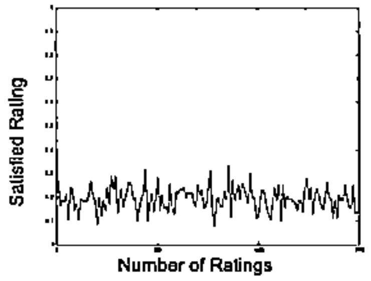

Fig. 1. Uncovered deceiving intention

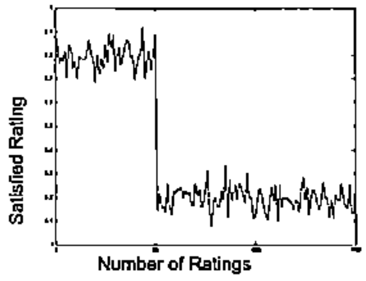

Fig. 2. Trapping intention

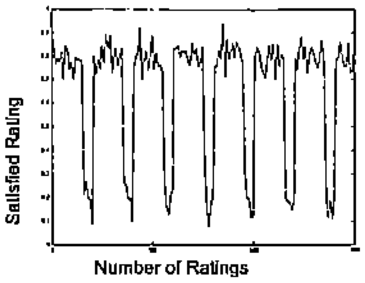

Fig. 3. Illusive intention

4. Architecture for swindler detection

The major functionality of swindler detection is to react to a suspicious cooperation that may lead to a fraud. The design of the architecture is based on the following considerations:

- Deviation from the usual pattern of an entity may imply the existence of a fraud, e.g., a sharp increase in sales income may result from accounting fraud.
- The similarity between an entity's current activity and a known fraud scenario indicates the same fraud may occur again.
- Analysis of an entity's behaviours in a relatively long period may reveal his real intentions that are covered by good activities.

Swindler detection consists of four components: profile-based anomaly detector, state transition analysis, deceiving intention predictor, and decision-making.

Profile-based anomaly detector monitors suspicious actions based upon the established patterns of an entity. It outputs fraud confidence indicating the possibility of a fraud. State transition analysis builds a state transition graph that provides state description to users when an activity results in entering a danger state that may lead to a fraud. Deceiving intention predictor discovers deceiving intention of the cooperation partner based on his history and satisfied ratings. DI-confidence is used to measure a deceiving intention, which characterizes the belief that the target entity has such an intention. It is a real number ranging over [0,1]. The higher the value is, the greater the belief is.

Anomaly detector, state transition analysis, and deceiving intention predictor work together to detect a potential swindler. The former two analyze the entity's activity in current cooperation while the last one investigates his past behaviours. They provide fraud confidence, states, and DI-confidence as inputs to the decision-making component, which assists users to reach decisions based on predefined policies.

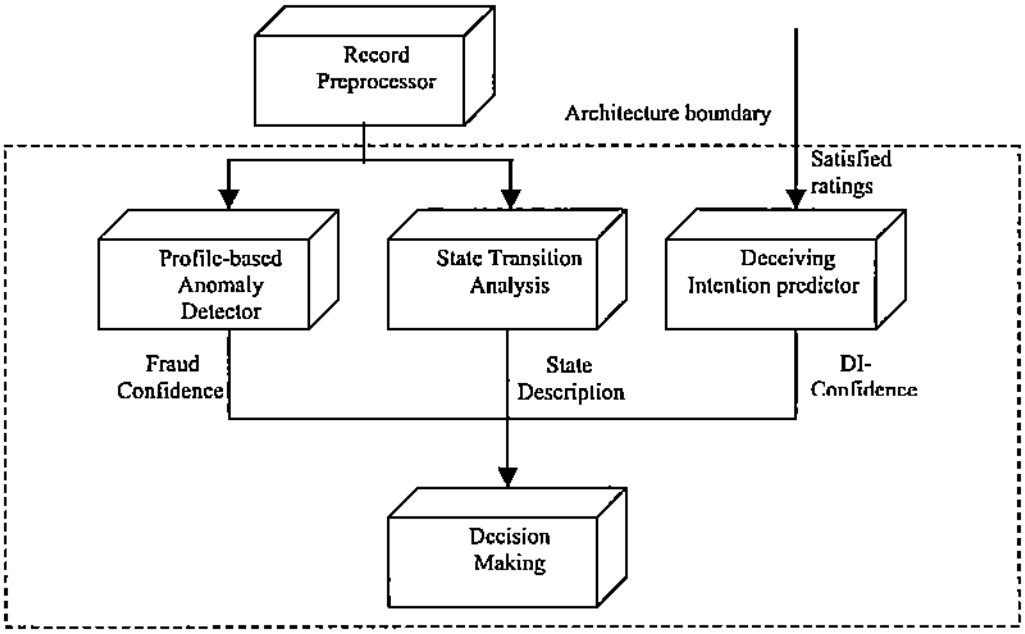

Fig. 4. Architecture for swindler detection

4.1 Profile-based anomaly detector

Profile-based anomaly detector monitors for a target entity's activities that deviate from established patterns. As illustrated in figure 5, the detector consists of three major components: rule generation and weighting, user profiling, and online detection.

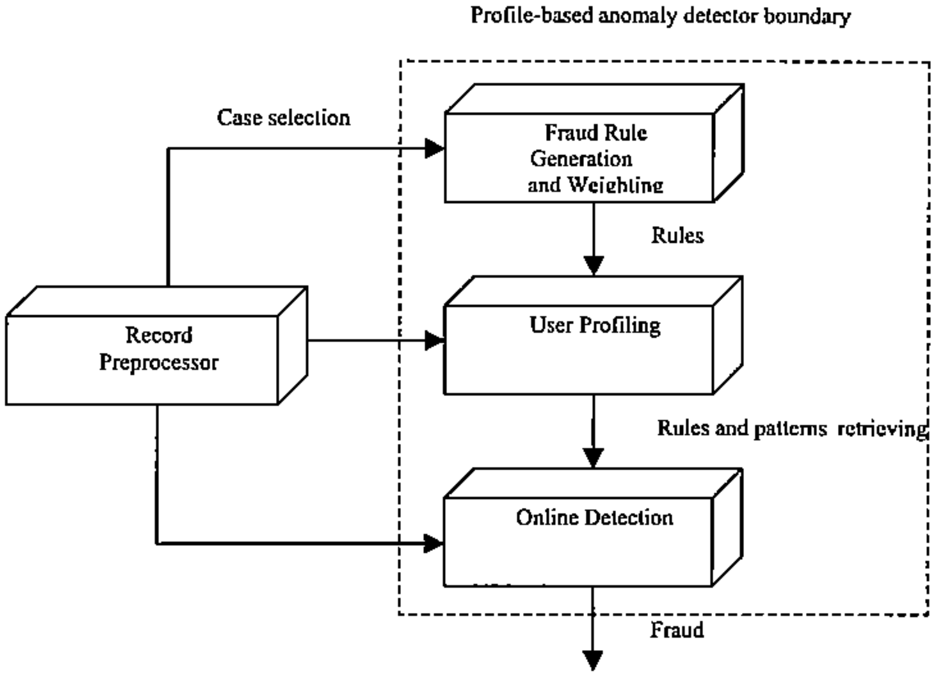
Profile-based anomaly detector boundary
Case selection
Fraud Rule Generation and Weighting
Rules
Record Preprocessor
User Profiling
Rules and patterns retrieving
Online Detection
Fraud

Fig. 5. Profile-based anomaly detector

Fraud rule generation and weighting applies data mining techniques to existing massive amount of activity records. From this information, fraud rules are generated and assigned weights according to their frequency of occurrence. Both entity-level and behaviour-level attributes are used in mining fraud rules and weighting. Normally, a large volume of rules will be generated.

The second component in the profile-based anomaly detector is user profiling. The profiling information characterizes entity-level information, such as age, location and financial status. It also captures an entity's behaviour patterns, such as how often he/she buys or sells, price range and interested products. There are two sets of profiling data, one for current profiles and the other for history profile. In order to reflect an entity's current behaviour patterns, the current profile set will be dynamically updated according to behaviours. As behaviour level data will grow larger, for efficiency reasons, decay is needed to reduce the data volume. This part also involves rule selection for a specific entity, based on profiling results and rules. When combined with profiling information, a set of rules as fraud indicators for monitoring a specific entity is selected. The rule selection also triggers the measurements of normal behaviors with respect to the rules. These statistics are stored in history profiles for later quick online detection reference.

The third component is online detection. When an activity occurs, the detection engine will retrieve the related rules from the profiling component. It may also need to retrieve the entity's current behavior patterns and behavior pattern history. Each rule will be checked and output a weight. For example, there are rules to check how abnormal current behavior is comparing to the behavior pattern history. If the deviation reaches the defined threshold, it will be caught. A weight will be output according to the rules. These results are combined to determine fraud confidence.

4.2 State transition analysis

State transition analysis models fraud scenarios as series of states changing from an initial secure state to a final compromised state. The initial state is the start state prior to actions that lead to a fraud. The final state is the resulting state of completing the fraud. There may be several intermediate states between them. The action, which causes one state to transit to another, is called the signature action. Signature actions are the minimum actions required to lead the state transition nearer the final state. Without such action, this fraud scenario will not be completed.

This model requires collecting fraud scenarios at the beginning and identifies the initial states and the final states. Then the signature actions for that scenario are identified in backward direction. The fraud scenario is represented as a state transition graph by the states and signature actions.

A *danger* factor is associated with each state. It is defined as the distance from the current state to the final state that indicates a fraud. If one state leads to several final states, the minimum distance is used. For each activity, state transition analysis checks the potential next states. If the maximum value of the *danger* factors associated with the potential states exceeds a threshold, a warning is raised and detailed state description is sent to decision-making component.

The state transition analysis model is a high-level abstraction in that it does not directly depend on data records. This analysis also has the advantage of foreseeing the impending states and enabling preventive actions before reaching the compromise state.

4.3 Deceiving intention predictor

The kernel of deceiving intention predictor is the deceiving intention prediction (DIP) algorithm that views the belief of deceiving intention as the complementary of trust belief. The trust belief about an entity is evaluated based on the satisfied sequence $<R_1, R_2, \dots, R_n>$. R_n is the most recent one, which contributes to α portion of the trust belief. The rest $(1 - \alpha)$ portion comes from the previous trust belief that is determined recursively. For each entity, DIP maintains a pair of factors (i.e. *construction factor* W_c and *destruction factor* W_d). If integrating current satisfied rating will increase trust belief, $\alpha = W_c$. Otherwise, $\alpha = W_d$. W_c and W_d are initialized by the user. They satisfy the constraint $W_c < W_d$ so that the property of easy-destruction-hard-construction can be implemented. This property stems from the fact that more efforts are needed to gain the same amount of trust than to loose it[14]. W_c and W_d are modified when a *foul event* occurs. A foul event is triggered when a satisfied rating is lower than a threshold. Upon a foul event, the target entity is put under supervision in the sense that his W_c is decreased and W_d is increased. If the entity does not conduct any foul event during the supervision period, his W_c and W_d are restored to the initial values. Otherwise, his W_c and W_d are further decreased and increased respectively. The supervision period associated with an entity will increase each time when he is put under supervision, so that he will be punished longer next time. In this way, an entity with worse history is treated harsher. The DI-confidence is computed as 1 – current trust belief.

```
DIP algorithm
$P_{(k)}$ is entity k's profile. It has six fields: tValue, $W_c$, $W_d$, period ($\tau'$), rest, DI-confidence. tValue stores
current trust belief. DI-confidence keeps current belief about entity k's deceiving intention
Input:  Foul event threshold $\gamma$. Initial construction factor $W_c$, destruction factor $W_d$ such that $W_c < W_d$.
        Initial supervision period $\tau$. $\rho_1$ is the penalty ratio used to decrease construction factor.
        $\rho_2$ and $\rho_3$ are the penalty ratios used to increase destruction factor and supervision period
        respectively. $\rho_1, \rho_2 \in (0, 1)$ and $\rho_3 > 1$.
        $P_{(k)}$.tValue = 0
        $P_{(k)}.W_d = W_d$
        $P_{(k)}.W_c = W_c$
        $P_{(k)}$.period = $\tau$
        $P_{(k)}$.rest = 0
        while there are new rating R
            if $R \leq \gamma$ then
                $P_{(k)}.W_d = P_{(k)}.W_d + \rho_1 \times (1 - P_{(k)}.W_d)$
```

```
9:             $P_{(k)}.W_c = \rho_2 \times P_{(k)}.W_c$
10:            $P_{(k)}.rest = P_{(k)}.rest + P_{(k)}.period$
11:            $P_{(k)}.period = \rho_3 \times P_{(k)}.period$
12:          end if
13:          if $R \le P_{(k)}.tValue$ then
14:            $W = P_{(k)}.W_d$
15:          Else
16:            $W = P_{(k)}.W_c$
17:          end if
18:          $P_{(k)}.tValue = P_{(k)}.tValue \times (1-W) + R \times W$
19:          if the entity k is under supervision and $R > \gamma$ then
20:            $P_{(k)}.rest = P_{(k)}.rest - 1$
21:            if $P_{(k)}.rest = 0$
22:              $P_{(k)}.W_d = W_d$
23:              $P_{(k)}.W_c = W_c$
24:            end if
25:          end if
26:          $P_{(k)}$.DI-Confidence = $1 - P_{(k)}.tValue$
27:        end while
```

The first to the fifth lines initialize an entity's profile. Line 7 determines if a foul event occurs. If so, the entity is put under supervision. His current construction factor is decreased and destruction factor is increased (lines 8-9). Line 10 computes how long he will stay under supervision. Line 11 increments the supervision period that will be used when he conducts a foul event next time. The thirteenth line updates tValue. If the entity is under supervision and the new interaction is not a foul event, his rest supervision period is reduced by one (lines 19-20). The construction factor and destruction factor are restored when the supervision period ends (lines 21-24). DI-confidence is (1 – tValue).

The increase of destruction factor w_d needs to satisfy the following constraints:

1. $\textit{for any } n, w_n^d \le 1 \textit{ and } w_n^d \le w_{n+1}^d$
2. $\lim_{n \to \infty} w_n^d = 1$

w_n^d is the destruction factor when an entity is put under supervision n times without being released even time. The first constraint ensures that w_n^d is monotonic increasing with n. The upper bound is 1. The second constraint indicates that the destruction factor can be close to 1 to any extent if n is large enough. The function we use is represented as follows.

$$w_{n+1}^d = w_n^d + \rho_1 \times (1 - w_n^d) \qquad (1)$$

By solving equation 1, we get:

$$w_n^d = 1 - (1 - w_0^d)(1 - \rho_1)^n$$

where, w_0^d is the initial destruction factor.

∴Equation 1 satisfies the above constraints.

Symmetrically, the decrease of construction factor w_c must satisfy the following constraints:

1. $\textit{for any } n, 0 \le w_n^c \textit{ and } w_{n+1}^c \le w_n^c$
2. $\lim_{n \to \infty} w_n^c = 0$

w_n^c is defined similarly as w_n^d is. The function we use is as follows.

$$w_{n+1}^c = \rho_2 \times w_n^c \qquad (2)$$

By solving equation 2, we get:

$$w_n^c = \rho_2^n \times w_0^c$$

where, w_0^c is the initial value of construction ratio.

∴ Equation 2 satisfies the above constraints.

4.3.1 Experimental study

The purpose of the experiments is to investigate the effectiveness of DIP to discover the three deceiving intentions defined in section 3.1. The parameters used in DIP are shown in table 1. W_c and W_d represent the initial construction and destruction factors respectively. ρ_1, ρ_2 and ρ_3 are penalty ratios for construction factor, destruction factor and supervision-period. γ is the threshold for a foul event.

W_c	W_d	ρ_1	ρ_2	ρ_3	γ
0.05	0.1	0.9	0.1	2	0.18

Table 1. Parameter values

The results are shown in figure 6 - 8. The x-axis of each figure is the number of ratings. The y-axis represents the evaluated DI-confidence.

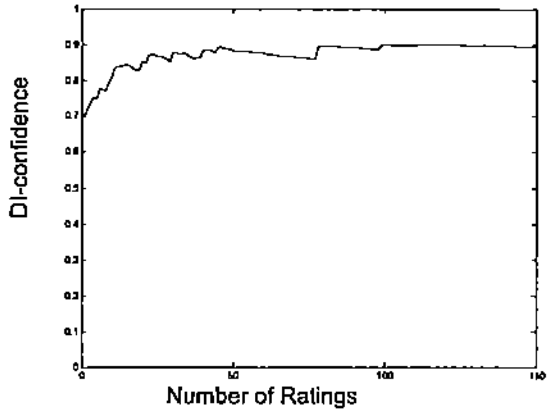

Fig. 6. Experiment on swindler with uncovered deceiving intention

Swindler with uncovered deceiving intention: The result of applying DIP algorithm to a swindler with uncovered deceiving intention is illustrated in figure 6. The mean function f_m (n) are shown in figure 1. Since the possibility for the swindler to conduct foul events is high, he is under supervision at most of the time. The construction and destruction factors become close to 0 and 1 respectively because of the punishment for foul events. The trust values are close to the minimum rating of interactions that is 0.1. The DI-confidence is around 0.9, which matches the experiment result.

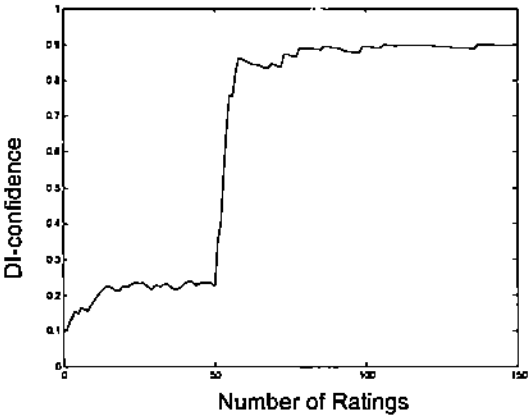

Fig. 7. Experiment on swindler with trapping intention

Swindler with trapping intention: The results of applying DIP algorithm to a swindler with trapping intention are illustrated in figure 7. The mean function f_m (n) are shown in figure 2. DIP algorithm responds to the sharp drop of f_m (n) very quickly. After f_m (n) changes from 0.8 to 0.2 it only takes 6 ratings for DI-confidence increasing from 0.2239 to 0.7592.

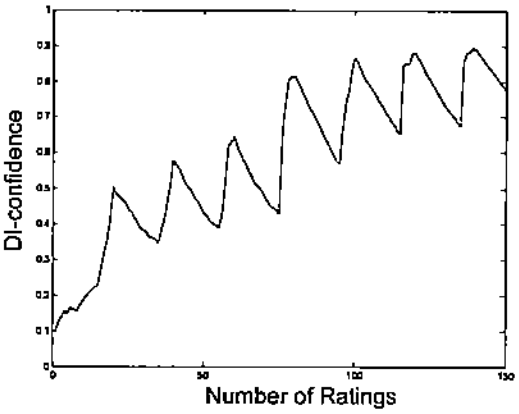

Fig. 8. Experiment on swindler with illusive intention

Swindler with illusive intention: The results of applying DIP algorithm to a swindler with illusive intention are illustrated in figure 8. The mean function f_m (n) are shown in figure 3. When the mean function f_m (n) changes from 0.8 to 0.2, DI-confidence increases. When f_m (n) changes back from 0.2 to 0.8, DI-confidence decreases. DIP algorithm is able to catch this smart swindler in the sense that his DI-confidence eventually increases to about 0.9. Figure 8 shows that the smart swindler's effort to cover a fraud with good behaviours has less and less effect with the number of frauds.

4.4 Decision-making

Decision-making component takes fraud confidence, state description, and DI-confidence as inputs. It passes warnings from state transition analysis to user and display the description of next potential state in a readable format. The expected risk is computed as follows.

f (fraud confidence, DI-confidence, estimated cost) = max (fraud confidence, DI-confidence) × estimated cost

Users can replace this function according to their specific requirements. A fraud alarm will arise when expected risk is greater than fraud-investigating cost.

5. Conclusion

In this paper, we classify fraudsters as impersonators and swindlers and present a mechanism to detect swindlers. The concepts relevant to frauds conducted by swindlers are formally defined. Three deceiving intentions are identified. The architecture for swindler detection consists of four components: profile-based anomaly detector, state transition analysis, deceiving intention predictor, and decision-making component. Profile-based anomaly detector monitors the sharp deviation from usual patterns. State transition analysis constructs state transition graph for known frauds. An alarm is raised when current activity results in a danger state possibly leading to a fraud. Deceiving intention predictor discovers malicious intentions based on target entity's history. DIP algorithm is developed for deceiving intention prediction. Experimental study shows that DIP effectively discovers three defined deceiving intentions.

References

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

[2] P. Burge, J. Shawe-Taylor, Y. Moreau, B. Preneel, C. Stoermann and C. Cooke. Fraud detection and management in mobile telecommunications networks, In Proc. of the European Conference on Security and Detection, 1997.

[3] 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.

[4] T. Fawcett and F. Provost. Adaptive fraud detection. In Data Mining and Knowledge Discovery, pages 291-316. 1997.

[5] D. W. Abbott, I. P. Matkovsky and J. F. Elder. An evaluation of high-end data mining tools for fraud detection. In Proc. of the 1998 IEEE International Conference on Systems, Man, and Cybernetics, 1998.

[6] 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.

[7] 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.

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

[9] 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.

[10] S. Abu-Hakima and M. Toloo. A multi-agent systems approach for fraud detection in personal communication systems. In Proc. of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97), 1997.

[11] 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.

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

[13] Ravi Sandhu. Lattice-Based Access Control Models. In IEEE Computer, Vol. 26, No. 11, pages 9-19, 1999.

[14] Y. Zhong, Y. Lu, B. Bhargava. Dynamic Trust Production Based on Interaction Sequence. Technique Report. The department of computer sciences, Purdue university. CSD-TR 03-006. 2003.