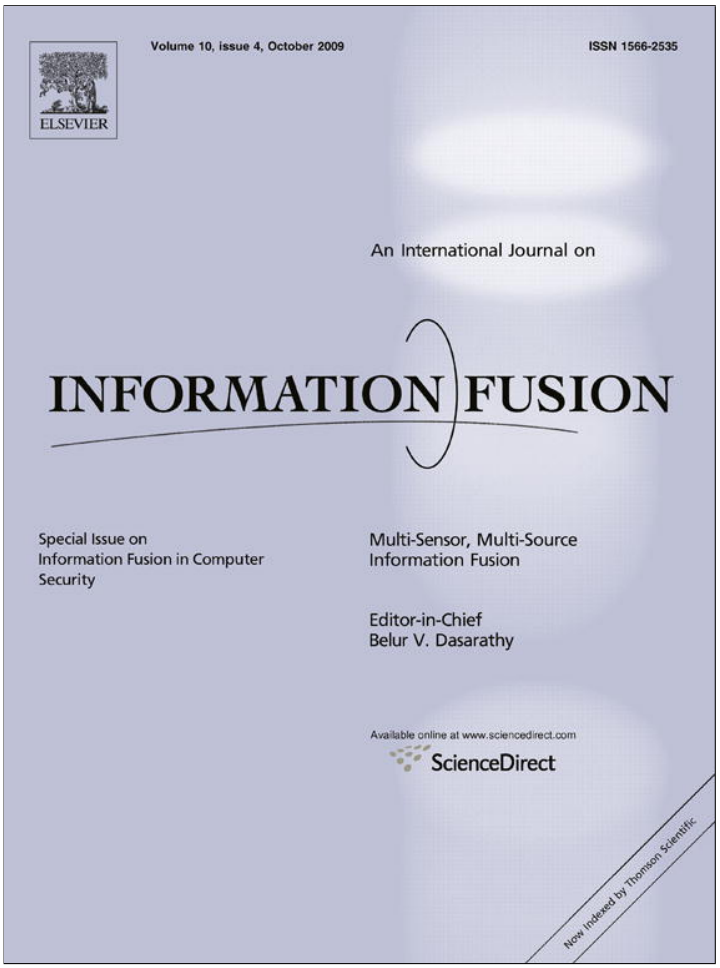

Information Fusion 10 (2009) 354–363

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Credit card fraud detection: A fusion approach using Dempster–Shafer theory and Bayesian learning

Suvasini Panigrahi [a], Amlan Kundu [a], Shamik Sural [a,*], A.K. Majumdar [b]

[a] School of Information Technology, Indian Institute of Technology, Kharagpur, India
[b] Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, India

ARTICLE INFO

Article history:
Received 30 November 2006
Received in revised form 19 July 2007
Accepted 14 April 2008
Available online 4 February 2009

Keywords:
FDS
Credit card
Dempster–Shafer theory
Bayesian learning
Suspicion score

ABSTRACT

We propose a novel approach for credit card fraud detection, which combines evidences from current as well as past behavior. The fraud detection system (FDS) consists of four components, namely, rule-based filter, Dempster–Shafer adder, transaction history database and Bayesian learner. In the rule-based component, we determine the suspicion level of each incoming transaction based on the extent of its deviation from good pattern. Dempster–Shafer's theory is used to combine multiple such evidences and an initial belief is computed. The transaction is classified as normal, abnormal or suspicious depending on this initial belief. Once a transaction is found to be suspicious, belief is further strengthened or weakened according to its similarity with fraudulent or genuine transaction history using Bayesian learning. Extensive simulation with stochastic models shows that fusion of different evidences has a very high positive impact on the performance of a credit card fraud detection system as compared to other methods.

1. Introduction

In today's electronic society, e-commerce has become an essential sales channel for global business. Due to rapid advancement of e-commerce, use of credit cards for purchases has dramatically increased. Unfortunately, fraudulent use of credit cards has also become an attractive source of revenue for criminals. Occurrence of credit card fraud is increasing dramatically due to the exposure of security weaknesses in traditional credit card processing systems resulting in loss of billions of dollars every year. Fraudsters now use sophisticated techniques to perpetrate credit card fraud. The fraudulent activities worldwide present unique challenges to banks and other financial institutions who issue credit cards. In case of bank cards (Visa and MasterCard) a study done by American Bankers Association in 1996 reveals that the estimated gross fraud loss was $790 million in 1995 [1]. The majority of the loss due to credit card fraud is suffered by the USA alone. This is not surprising since 71% of all credit cards are issued in the USA only. In 2005, the total fraud loss in the USA was reported to be $2.7 billion and it has gone up to $3.2 billion in 2007 [2]. Another survey of over 160 companies revealed that online fraud (committed over the Web or phone shopping) is 12 times higher than offline fraud (committed by using a stolen physical card) [3].

To address this problem, financial institutions employ various fraud prevention tools like real-time credit card authorization, address verification systems (AVS), card verification codes, rule-based detection, etc. But fraudsters are adaptive, and given time, they devise several ways to circumvent such protection mechanisms. Despite the best efforts of the financial institutions, law enforcement agencies and the government, credit card fraud continues to rise. In addition to significant financial losses, the main concern of the law enforcement agencies is that this money is also used to support other criminal activities worldwide. Thus, once fraud prevention measures have failed, there is a need for effective technologies to detect fraud in order to maintain the viability of the payment system. Fraudsters constitute a very inventive and fast moving fraternity. As preventive technology changes, so does the technology of criminals and the way they go about with their fraudulent activities.

The possibility of enhancing existing operations by introducing an effective FDS constitutes the objective of our work.

2. Related work

The approaches used in detecting credit card fraud mainly include neural network, data mining, meta-learning, game theory and support vector machine.

Artificial neural networks (ANN) have been considered for credit card fraud detection by Ghosh and Reilly [4], Aleskerov et al. [5] and Dorronsoro et al. [6]. Ghosh and Reilly [4] carried out a feasibility study for Mellon Bank to determine the effectiveness

* Corresponding author. Tel.: +91 3222 282330; fax: +91 3222 282206.
E-mail addresses: Suvasini.Panigrahi@sit.iitkgp.ernet.in (S. Panigrahi), kunduamlan@sit.iitkgp.ernet.in (A. Kundu), shamik@sit.iitkgp.ernet.in (S. Sural), akmj@cse.iitkgp.ernet.in (A.K. Majumdar).

1566-2535/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.inffus.2008.04.001

of neural network for credit card fraud detection. The authors concluded that it was possible to achieve a reduction of 20–40% in the total fraud losses. Aleskerov et al. [5] present CARDWATCH, a neural network based data mining system for credit card fraud detection. The system trains a neural network with the past data of a particular customer, which can then be used to analyze the current spending behavior of that customer and detect anomalies. They use three transaction features to represent a customer's spending pattern – category of purchase, transaction amount and time since last purchase of the same category. The system was tested with synthetically generated data. Dorronsoro et al. [6] describe the domain of fraud detection as having two particular characteristics – a very limited time span for decisions and a large number of credit card operations to be processed. They have used Fisher's discriminant analysis to separate the fraudulent operations from the normal ones.

More recently, Syeda et al. [7] have suggested the use of parallel granular neural networks for speeding up the data mining and knowledge discovery process. Maes et al. [8] have outlined an automated credit card fraud detection system by ANN as well as Bayesian belief networks (BBN). They show that BBN gives better results related to fraud detection and the training period is faster whereas the actual detection process is substantially faster with ANN. The neural network based methods are, in general, fast but not so accurate. Re-training the neural networks is also a major bottleneck since the training time is quite high.

Chen et al. [9] propose a novel method in which an online questionnaire is used to collect questionnaire-responded transaction (QRT) data of users. A support vector machine (SVM) is trained with this data and the QRT models are used to predict new transactions. Chen et al. [10] have recently presented a personalized approach for credit card fraud detection that employs both SVM and ANN. It tries to prevent fraud for users even without any transaction data. However, these systems are not fully automated and depend on the user's expertise level.

Some researchers have applied data mining for credit card fraud detection. Chan et al. [11] divide a large set of transactions into smaller subsets and then apply distributed data mining for building models of user behavior. The resultant base models are then combined to generate a meta-classifier for improving detection accuracy. Brause et al. [12] have explored the possibility of combining advanced data mining techniques and neural networks to obtain high fraud coverage along with a low false alarm rate. Use of data mining is also elaborated in the work by Chiu and Tsai [13]. They consider web services for data exchange among banks. A fraud pattern mining (FPM) algorithm has been developed for mining fraud association rules which give information regarding the features that exist in fraud transactions. Banks enhance their original fraud detection systems by using the new fraud patterns to prevent attacks. While data mining techniques are relatively accurate, they are inherently slow.

Meta-learning is a general strategy that provides a means for combining and integrating a number of separately learned classifiers or models. A meta-learning system allows financial institutions to share their models of fraudulent transactions by exchanging classifier agents. Stolfo et al. [14] suggest a meta-learning technique to learn patterns of fraudulent credit card transactions. They apply four base classifiers, namely, ID3, CART, Bayes and RIPPER and use the class-combiner strategy [15] to select the best classifier for meta-learning. It has been shown that meta-learning with Bayes gives good accuracy. Prodromidis and Stolfo [16] describe an artificial intelligence based approach that combines inductive learning algorithms and meta-learning methods to build accurate classification models for electronic fraud detection. The field of game theory has also been explored for credit card fraud detection. Liu and Li [17] suggest a game-theoretic approach for prediction of attacks on IDS protected systems and a specific prediction model for credit card fraud. Vatsa et al. [18] have modeled the interaction between an attacker and an FDS as a repeated game between two players, each trying to maximize its payoff. Such game-theoretic models make a number of assumptions, like availability of strategies, actions and payoffs to both the players, which are not often valid in practice. For example, it is quite unusual for a bank to advertise its strategies for fraud detection.

Some survey papers have been published which categorize, compare and summarize articles in the area of fraud detection. Phua et al. [19] did an extensive survey of data mining based FDSs and presented a comprehensive report. Kou et al. [20] have reviewed the various fraud detection techniques including credit card fraud, telecommunication fraud as well as computer intrusion detection. Bolton and Hand [21] describe the tools available for statistical fraud detection and areas in which fraud detection technologies are most commonly used.

Majority of the FDSs as described above show a lot of variation in their accuracy. The main challenge identified by most of them is that the bulk of the transactions flagged as fraudulent by the FDSs are in fact genuine. A substantial amount of time and money is spent by bankers in investigating a large number of legitimate cases. It also causes customer inconvenience and potential dissatisfaction. In credit card application, since occurrence of fraud is sparse, it involves detecting a relatively rare event from a very large collection of routine transactions. Axelsson [22] has pointed out that due to the base-rate fallacy problem, the factor limiting the performance of an intrusion detection system is not the ability to identify intrusive behavior correctly but its ability to minimize false alarms. While failure to detect a fraud causes direct loss to the company, follow up actions needed to pursue false alarms also tend to be costly. Any design choice that attempts to improve the rate of correct detection of fraud, usually causes a rise in the false alarms as well. One of the motivations of our current research is to address this challenge.

It is well known that every cardholder has a certain shopping behavior, which establishes an activity profile for him. Almost all the existing fraud detection techniques try to capture these behavioral patterns as rules and check for any violation in subsequent transactions. However, these rules are largely static in nature. As a result, they become ineffective when the cardholder develops new patterns of behavior that are not yet known to the FDS. The goal of a reliable detection system is to learn the behavior of users dynamically so as to minimize its own loss. Thus, systems that cannot evolve or "learn", may soon become outdated resulting in large number of false alarms. A fraudster can also attempt new types of attacks which should still get detected by the FDS. For example, a fraudster may aim at deriving maximum benefit either by making a few high value purchases or a large number of low value purchases in order to evade detection. Thus, there is a need for developing fraud detection systems which can integrate multiple evidences including patterns of genuine cardholders as well as that of fraudsters.

We propose a credit card fraud detection system that combines different types of evidences using Dempster–Shafer theory. The purpose of aggregation is to meaningfully summarize and simplify bulk data which might be coming from a single source or multiple sources. Familiar examples of aggregation techniques include arithmetic, geometric and harmonic averages, maximum and minimum functions, etc. [23]. Cremer et al. [24] have shown that sensor fusion improves detection rate and reduces false alarms over single sensor solutions. They use different sensor data fusion techniques, namely, Dempster–Shafer, Bayes and fuzzy logic for the detection of anti-personnel land mines. By testing on synthetic data set, they have shown that Dempster–Shafer and Bayes approach outperform the fuzzy technique. Comparing the

receiver operator characteristics (ROC) curves (detection rate plotted against false alarm rate) for Dempster–Shafer and Bayes, they found that Dempster–Shafer has a slight advantage over Bayes. Besides combining evidences, we also incorporate learning in our system through application of prior knowledge and observed data on suspicious cards. Bayesian learning is used, which is a probabilistic approach to inferencing and provides a framework for building intelligent learning systems. It gives a formal and consistent way of reasoning in presence of uncertainty and optimal decisions can be made on the quantities of interest. Moreover, Bayesian methods match human intuition very closely and provide a promising model for neurological processes. The mathematical foundation of Bayesian reasoning is quite mature and widely used in many areas of science and engineering. To the best of our knowledge, this is the first ever attempt to develop a credit card fraud detection system using information fusion and Bayesian learning.

The rest of the paper is organized as follows. We present the components of our credit card fraud detection system in Section 3 along with a description of the methodology and the details of implementation. In Section 4, we discuss the results obtained from simulation studies. Finally, we conclude in Section 5 of the paper.

3. Proposed fraud detection system

The proposed FDS may be abstractly represented as a 6-tuple $\langle \text{System}, C, P, \psi, \theta_{LT}, \theta_{UT} \rangle$, where:

1. *System* refers to the target system that is being attacked.
2. $C = \{C_1, C_2, \ldots, C_n\}$ is the set of credit cards on which fraud detection is performed.
3. $P = \{P(C_1), P(C_2), \ldots, P(C_n)\}$ is the set of profiles, where each $P(C_k)$ corresponds to the profile of the owner of the card C_k. The profile of a cardholder is a set of patterns containing information like card number, transaction amount and time since last purchase.
4. $\psi(T_{j,\rho}^{C_k})$ is the suspicion score of the jth transaction $T_{j,\rho}^{C_k}$ on card C_k and ρ is the time gap from the previous transaction on the same card.
5. θ_{LT} is the lower threshold, where $0 \leqslant \theta_{LT} \leqslant 1$.
6. θ_{UT} is the upper threshold, where $0 \leqslant \theta_{UT} \leqslant 1$ and $\theta_{LT} \leqslant \theta_{UT}$.

3.1. FDS components

In the proposed FDS, a number of rules are used to analyze the deviation of each incoming transaction from the normal profile of the cardholder by assigning initial beliefs to it. The initial belief values are combined to obtain an overall belief by applying Dempster–Shafer theory. The overall belief is further strengthened or weakened according to its similarity with fraudulent or genuine transaction history using Bayesian learning. In order to meet this functionality, the proposed FDS is designed with the following four major components:

(1) Rule-based filter.
(2) Dempster–Shafer adder.
(3) Transaction history database.
(4) Bayesian learner.

3.1.1. Rule-based filter (RBF)

The RBF consists of generic as well as customer-specific rules which classify an incoming transaction as fraudulent with a certain probability. It measures the extent to which the transaction's behavior deviates from the normal profile of the cardholder. This layer can have rules like average daily/monthly spending of a customer, shipping address being different from billing address, etc. We briefly discuss two of the rules here.

3.1.1.1. Address mismatch. The most basic check performed by various credit card companies is "billing address and shipping address mismatch". Orders could be shipped to an address different from the billing address. This check does not help us in declaring a transaction as fraudulent with complete certainty since a genuine cardholder could gift some items to his friend. However, a transaction that clears this check can be classified as genuine with very high probability (except for the cases where the fraudster's aim is only to harass the cardholder). The transactions that violate this check are labeled as suspect.

3.1.1.2. Outlier detection. A customer usually carries out similar types of transactions in terms of amount, which can be visualized as part of a cluster. Since a fraudster is likely to deviate from the customer's profile, his transactions can be detected as exceptions to the cluster – a process known as outlier detection. It has important applications in the field of fraud detection and has been used for quite some time to detect anomalous behavior. Hodge and Austin [25] have done an extensive survey of outlier detection techniques and given a comparative review of the different methodologies.

DBSCAN (density based spatial clustering of applications with noise) is a density based clustering algorithm [26] which can be used to filter out outliers and discover clusters of arbitrary shapes. Formally, let $C' = \{c_1, \ldots, c_n\}$ denote the clusters in a database D for a particular card C_k and $A = \{a_1, a_2, \ldots, a_n\}$ be the set of attributes used to generate these clusters. For any credit card transaction, the possible attributes are transaction amount, billing address, shipping address and inter-transaction time gap. A transaction $T_{j,\rho}^{C_k}$ is detected as an outlier if it does not belong to any cluster in the set C'. Such an observation gives evidence that the transaction could be fraudulent. We measure the extent of deviation of an incoming transaction by its degree of outlierness. If the average distance of the amount p of an outlier transaction $T_{j,\rho}^{C_k}$ from the set of existing clusters in C' is v_{avg}, then its *degree of outlierness* $d_{outlier}$ is given by:

$$d_{outlier} = \begin{cases} 1 - \frac{\varepsilon}{v_{avg}} & \text{if } |N_\varepsilon(p)| < MinPts \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where

MinPts: Minimum number of points required in the ε-neighborhood of each point to form a cluster.
ε: Maximum radius of the neighborhood $N_\varepsilon(p) = \{q \in D | \text{dist}(p,q) \leqslant \varepsilon\}$.

The key idea of the DBSCAN algorithm is that for each point p in a cluster c_i, there are at least a minimum number of points (*MinPts*) in the ε-neighborhood of that point p denoted as $N_\varepsilon(p)$ i.e., the density in the ε-neighborhood has to exceed some threshold. The values of the parameters ε and *MinPts* are determined using an effective heuristic given by Ester et al. [26]. The larger the ε-neighborhood, the less is the number of clusters formed. In the limit, there will be only one large cluster. Also, the higher the value of *MinPts*, less is the number of clusters formed. If it is set too high, there will be no cluster since the *MinPts* condition is not satisfied. However, if both the parameters are small, there can be a lot of clusters. If *MinPts* is set to 1, then each point in the database is treated as a separate cluster and even noise gets identified as a separate cluster. The clusters can be formed by using different attributes, although in the current work, we use 'transaction amount' as the attribute for generating outliers. Here, transaction

amount refers to the total value of a transaction. The algorithm can be extended to include other attributes as well.

Usually, an FDS is subjected to a large number of transactions for authorization, a high percentage of them being genuine. The RBF is essential since it separates out most of the easily recognizable genuine transactions from the rest.

For convenience, we refer to the "Address Mismatch" rule as R_1 and the "Outlier Detection" rule as R_2.

3.1.2. Dempster–Shafer adder (DSA)

The role of the DSA is to combine evidences from the rules R_1 and R_2 and compute an overall belief value for each transaction. It may be noted that some attempts have been made to apply Dempster–Shafer theory (DST) to computer security. Wang et al. [27] present a distributed intrusion detection system, which uses DST to combine evidences from distributed sensors. They show that multi-sensor data fusion scheme gives better performance than a single sensor. Chen and Venkataramanan [28] have applied Dempster–Shafer approach to distributed intrusion detection in ad hoc networks. Data from multiple nodes are combined to estimate the likelihood of intrusion. A good application of DST is covered in the work of Yi et al. [29]. They have introduced a novel way of using the conflict value in DST for a given sensor model and experimentally shown considerable improvement in performance.

For the credit card fraud detection problem, DST is more relevant as compared to other fusion methods since it introduces a third alternative: "unknown", along with the measure of confidence in each of the alternatives. It provides a rule for computing the confidence measures of three states of knowledge: *fraud*, *fraud* and *suspicious* (unknown) based on data from new as well as old evidence. Furthermore, in DST, evidence can be associated with multiple possible events unlike traditional probability theory where evidence is associated with only one event. As a result, evidence can be more meaningful at a higher level of abstraction. Hence, we use Dempster–Shafer theory for combining evidences for this problem. However, one of the shortcomings of DST is that, for evidences with a high degree of conflict, the modeling may not be accurate. An extension of DST has recently been suggested to overcome this problem [30].

DST is a mathematical theory of evidence based on belief functions and plausible reasoning. It assumes a Universe of Discourse U, also called the Frame of discernment, which is a set of mutually exclusive and exhaustive possibilities [31]. For every incoming transaction $T_{j,\rho}^{C_k}$, the rules R_1 and R_2 contribute their independent observations about the behavior of the transaction. The part of DST that is of direct relevance is Dempster's rule for combination [32]. It gives us a numerical procedure for fusing together observations from the rule-based filter to compute an overall belief for a transaction. Two basic probability assignments $m_1(h)$ and $m_2(h)$ are combined into a third basic probability assignment $m(h)$ as follows:

$$m(h) = m_1(h) \oplus m_2(h) = \frac{\sum_{x \cap y = h} m_1(x) * m_2(y)}{1 - \sum_{x \cap y = \phi} m_1(x) * m_2(y)} \tag{2}$$

For the credit card fraud detection problem, the frame of discernment U consists of two possible values for any suspected transaction $T_{j,\rho}^{C_k}$ which is given as $U = \{fraud, \neg fraud\}$. For this U, the power set has three possible elements: hypothesis $h = \{fraud\}$ implying that $T_{j,\rho}^{C_k}$ is fraudulent, hypothesis $\bar{h} = \{\neg fraud\}$ that it is not, and universe hypothesis U that $T_{j,\rho}^{C_k}$ is suspicious. The basic probability assignments (BPAs) for the two rules R_1 and R_2 can now be given as follows:

- BPA for R_1: We assume that if address mismatch occurs then there is a high probability that it is a fraud transaction and low probability that it is a genuine transaction. We consider the following basic probability assignments:

$$\begin{aligned} m_1(h) &= 0.6 \\ m_1(\bar{h}) &= 0 \\ m_1(U) &= 0.4 \end{aligned} \tag{3}$$

  It may be noted that, if $m_1(h)$ is set too high, the probability that a transaction is detected as fraudulent will go up. As a result, although the detection rate improves, it also raises the number of false alarms. Similarly, if $m_1(U)$ is set high, the number of suspicious transactions goes up, which increases the number of misses (fraudulent transactions are allowed to go through). The values 0.6 and 0.4 have been chosen to maintain a balance between these two requirements.

- BPA for R_2: For a transaction detected as an outlier, we make the following basic probability assignments using the *degree of outlierness* given in Eq. (1) of Section 3.1.1:

$$\begin{aligned} m_2(h) &= 1 - \frac{\varepsilon}{v_{avg}} \\ m_2(\bar{h}) &= 0 \\ m_2(U) &= 1 - \left(1 - \frac{\varepsilon}{v_{avg}}\right) \end{aligned} \tag{4}$$

  Here the zero in the basic probability assignment of $\bar{h}$ does not imply impossibility. It means that neither of the rules R_1 and R_2 give any support to the belief that transaction $T_{j,\rho}^{C_k}$ is genuine. Following Eq. (2), the combined belief of R_1 and R_2 in h is derived as:

$$P(h) = m_1(h) \oplus m_2(h) \tag{5}$$

  Based on this belief value $P(h)$, the transaction on a particular card can be initially classified as normal, abnormal or suspicious. Since $P(h)$ and $P(\bar{h})$ add to unity, $P(\bar{h}) = 1 - P(h)$.

3.1.3. Transaction history database (THD)

THD is the transaction repository component of the proposed FDS. History records of both fraudulent and genuine transactions are used to construct models which allow us to extract characteristic information of the two classes from available data. For accomplishing this, we have built a good transactions history (GTH) for individual customers from their past behavior and a generic fraud transactions history (FTH) from different types of past fraud data. We represent each history transaction by a set of attributes containing information like card number, transaction amount and time since last purchase. While observing the current spending behavior on a credit card by rule R_2, we also accumulate and analyze past spending behavior in terms of the frequency of transactions on that card. The transaction amount information in the THD is required for detecting outliers. The expected behavior of a fraudster is to maximize his benefit from a stolen card [33]. This can be achieved by carrying out high value transactions frequently. However, to avoid detection, the fraudsters can make either high value purchases at longer time gaps or small value purchases at shorter time gaps. Contrary to such usual behavior, a fraudster may also carry out low value purchases at longer time gaps. This would be difficult for the FDS to detect if it resembles the genuine cardholder's profile. However, in such cases, the total loss incurred by the credit card company will also be quite low.

To capture the frequency of card use, we consider the time gap between successive transactions on the same card. The transaction gap is divided into four mutually exclusive and exhaustive events – D_1, D_2, D_3 and D_4. Occurrence of each event depends on the time since last purchase (transaction gap) on any particular card. The event D_1 is defined as the occurrence of a transaction on the same card C_k within 8 h of the last transaction which can be represented as:

$$D_1 = True | \{\exists T_{j,\rho}^{C_k} \wedge (0 < \rho \leqslant 8)\} \tag{6}$$

Similarly, the events D_2, D_3 and D_4 can be expressed as:

$$D_2 = True|\{\exists T^{Ck}_{j,\rho} \wedge (8 < \rho \leqslant 16)\} \tag{7}$$

$$D_3 = True|\{\exists T^{C_k}_{j,\rho} \wedge (16 < \rho \leqslant 24)\} \tag{8}$$

$$D_4 = True|\{\exists T^{C_k}_{j,\rho} \wedge (\rho > 24)\} \tag{9}$$

The event D is the union of all the four events D_1, D_2, D_3 and D_4 such that:

$$P(D) = \sum_{i=1}^{4} P(D_i) = 1 \tag{10}$$

It may be noted that, we chose the above definitions of D_i's to handle frequent as well as infrequent transactions during experimentation. Other values could be similarly defined. Card-specific definitions of D_i's also can be derived by clustering transaction gaps for each cardholder.

We next compute $P(D_i|h)$ and $P(D_i|\bar{h})$ from the FTH and the GTH, respectively. $P(D_i|h)$ measures the probability of occurrence of D_i given that a transaction is originating from a fraudster and $P(D_i|\bar{h})$ measures the probability of occurrence of D_i given that it is genuine. The likelihood functions $P(D_i|h)$ and $P(D_i|\bar{h})$ are given by the following equations:

$$P(D_i|h) = \frac{\#(\text{Occurrences of } D_i \text{ in FTH})}{\#(\text{Transactions in FTH})} \tag{11}$$

$$P(D_i|\bar{h}) = \frac{\#(\text{Occurrences of } D_i \text{ on } C_k \text{ in GTH})}{\#(\text{Transactions on } C_k \text{ in GTH})} \tag{12}$$

We have created two look-up tables FFT (fraud frequency table) and GFT (good frequency table) to maintain the values of $P(D_i|h)$ and $P(D_i|\bar{h})$.

Using Eqs. (11) and (12), $P(D_i)$ can be computed as follows:

$$P(D_i) = P(D_i|h) * P(h) + P(D_i|\bar{h}) * P(\bar{h}) \tag{13}$$

The initial belief $P(h)$ of Eq. (5) can be updated by using Bayes rule after getting the new information D_i from the THD. We update the THD frequently in order to retain the accuracy of the FDS, thus reducing the number of false alarms. THD update is an offline procedure.

3.1.4. Bayesian learner (BL)

Bayesian learning is a tool to measure evidences supporting alternative hypotheses and arrive at optimal decisions. The general idea of belief revision is that, whenever new information becomes available, it may require updating of prior beliefs. Bayes rule gives the mathematical formula for belief revision, which can be expressed as follows:

$$P(h|D_i) = \frac{P(D_i|h) * P(h)}{P(D_i)} \tag{14}$$

By substituting Eq. (13) in Eq. (14) we get:

$$P(h|D_i) = \frac{P(D_i|h) * P(h)}{P(D_i|h) * P(h) + P(D_i|\bar{h}) * P(\bar{h})} \tag{15}$$

We use Bayesian learning to update the suspicion score (Ψ) of a transaction in the light of the new evidence D_i from the THD. Ψ is the probability that the current transaction is fraudulent. The goal of Bayesian learning is to find the most probable hypothesis h_{map} given the training data. This is known as the maximum a posteriori hypothesis (MAP Hypothesis) which can be written as:

$$h_{map} = \max_{h \in H} P(h|D_i) \tag{16}$$

Thus, for each hypothesis h in the hypothesis space H, we calculate the posterior probability $P(h|D_i)$ and $P(\bar{h}|D_i)$ by using Bayes rule and then output the hypothesis with the highest posterior probability as h_{map}. The credit card fraud detection problem has the following two hypotheses: h:*fraud* and $\bar{h}$: ¬*fraud*. By substituting the values obtained from Eqs. (5), (11) and (12) in Eq. (15), the posterior probability for hypothesis h:*fraud* is given as:

$$P(fraud|D_i) = \frac{P(D_i|fraud) * P(fraud)}{P(D_i|fraud) * P(fraud) + P(D_i|\neg fraud) * P(\neg fraud)} \tag{17}$$

Similarly, the posterior probability for hypothesis $\bar{h}$: ¬*fraud* is given as:

$$P(\neg fraud|D_i) = \frac{P(D_i|\neg fraud) * P(\neg fraud)}{P(D_i|\neg fraud) * P(\neg fraud) + P(D_i|fraud) * P(fraud)} \tag{18}$$

Depending on which of the two posterior values is greater, future actions are decided by the FDS.

3.2. Methodology

The working principle of the proposed FDS is presented in Algorithm 1. It takes the transaction parameters – card number (C_k), transaction amount (T_{amount}), billing address (B_{addr}), shipping address (S_{addr}) and transaction gap (ρ) as well as the design parameters – ε, *MinPts*, θ_{LT} and θ_{UT} as input. θ_{LT} and θ_{UT} can be chosen by observing the performance of the FDS over a large number of transactions. In Section 4.2.1, we will show the impact of these settings on the system accuracy.

An incoming transaction is first handled by the rule-based component of the system. Basic probability values BPA_R_1 and BPA_R_2 from the RBF are combined using the DSA to get the initial belief $P(h)$ for the transaction. If $P(h) < \theta_{\text{LT}}$, the transaction is considered to be genuine and is approved. On the other hand, if $P(h) > \theta_{\text{UT}}$ then the transaction is declared to be fraudulent and manual confirmation is made with the cardholder. In case $\theta_{\text{LT}} \leqslant P(h) \leqslant \theta_{\text{UT}}$, the transaction is allowed but the card C_k is labeled as suspicious. If this is the first suspicious transaction on this card, then the card number is inserted into a *suspect* table. The FDS then waits until the next transaction occurs on the same card number.

When the next transaction occurs on the same card C_k, it is also passed to the FDS. The rule-based component of the FDS again assigns a belief to the transaction. In case the transaction is found to be suspicious, it is inserted in the *suspect* table. Since each transaction is time stamped, from the time gap ρ between the current and the last transaction, the FDS determines which event E has occurred out of the four D_i's and retrieves the corresponding $P(E|h)$ and $P(E|\bar{h})$ values from the tables FFT and GFT, respectively. The posterior beliefs $P(h|E)$ and $P(\bar{h}|E)$ are next computed using Eqs. (17) and (18) and MAP hypothesis is applied.

$P(h|E)$ and $P(\bar{h}|E)$are the updated beliefs about the last transaction on card C_k based on the evidence from THD and previous round suspicion score Ψ (last round). Since for the second suspicious transaction on a card, there is no Ψ (last round), the $P(h)$ value of the first round is itself taken as Ψ (last round) and posterior beliefs are computed based on this value. If $P(h|E) \geqslant P(\bar{h}|E)$, then the FDS applies the D–S rule of combination to get the suspicion score Ψ (current round) by combining $P(h|E)$ and current round $P(h)$. The current round Ψ value is inserted into the *suspect* table at the end of each round unless the suspicious score fall below θ_{LT}. The flow of events in the FDS has been depicted in the block diagram of Fig. 1.

Algorithm 1

Input: C_k, T_{amount}, B_{addr}, S_{addr}, ε, *MinPts*, θ_{LT}, θ_{UT}, ρ
$P_{R1}[\,] = BPA\_R_1(B_{addr}, S_{addr})$ // Using Eq. (3)
$P_{R2}[\,] = BPA\_R_2(T_{amount}, \varepsilon, MinPts)$ // Using Eq. (4)
$P_h = ds\_add(P_{R1}[\,], P_{R2}[\,])$ // Using Eq. (5) // $P_{R1}[\,]$ and $P_{R2}[\,]$ are arrays of size 3
if ($P_h < \theta_{LT}$) **then**
  Output("Genuine") // The transaction is approved
else if ($P_h > \theta_{UT}$) **then**
  Output("Fraudulent") // Check with customer
  if (transaction verified to be fraudulent) **then**
    Block_Card(C_k)
  end if
else
  if (exists(C_k) = = false) **then** // Returns false if C_k is not in *suspect* table
    $\Psi = P_h$
    Insert_Suspect_Table(C_k, Ψ) // Enter C_k in *suspect* table
    Wait for the next transaction on the card C_k
else
  $E = find\_event(\rho)$ // Using Eqs. (6)–(9)
  $E_h = compute\_event\_prob_h(E)$ // Using Eq. (11)
  $E_{\bar{h}} = compute\_event\_prob_{\bar{h}}(E)$ // Using Eq. (12)
  $Posterior_h = compute\_posterior\_belief_h(\psi, 1-\psi, E_h, E_{\bar{h}})$ // Using Eq. (17)
  $Posterior_{\bar{h}} = compute\_posterior\_belief_{\bar{h}}(\psi, 1-\psi, E_h, E_{\bar{h}})$ // Using Eq. (18)
  if ($Posterior_h > Posterior_{\bar{h}}$) **then** // Using Eq. (16)
    $A_1[0] = P_h$, $A_1[1] = 0$, $A_2[2] = 1 - P_h$ // $A_1[\,]$ and $A_2[\,]$ are arrays of size 3
    $A_2[0] = Posterior_h$, $A_2[1] = 0$, $A_2[2] = 1 - Posterior_h$
    $\Psi = ds\_add(A_1[\,], A_2[\,])$ // Using Eq. (5)
    Insert_Suspect_Table(C_k, Ψ)
    if ($\Psi < \theta_{LT}$) **then**
      Output("Genuine")
      Delete_Suspect_Table(C_k) // Remove C_k from *suspect* table
    else if ($\Psi > \theta_{UT}$) **then**
      Output("Fraudulent") // Check with customer
      if (transaction verified to be fraudulent) **then**
        Block_Card(C_k)
    end if
    else
      Wait for the next transaction on the card C_k
      end if
    end if
  end if
end if

Whenever a transaction is found to be anomalous and the fraudulent behavior is conformed from the cardholder, the corresponding card number and associated transactions are moved from the GTH to the FTH in order to maintain the consistency of the THD and to build the FTH.

3.3. Sample run

In Table 1, we show sample results over two rounds using the proposed methodology. Let us consider: $\theta_{LT} = 0.3$ and $\theta_{UT} = 0.7$.

Suppose initial belief $P(h) = 0.55$ which is obtained by combining the evidences from rules R_1 and R_2. Since $\theta_{LT} \leqslant 0.55 \leqslant \theta_{UT}$, the transaction is labeled as suspicious. We assume that it is the first suspicious transaction on this card and hence, the card number is entered into the *suspect* table. Suppose the next transaction occurs on the same card number within 8–16 h. It is passed to the RBF and let us assume that we get $P(h) = 0.62$. The transaction is again found to be suspicious. From the transaction gap, we determine that the event D_2 has occurred and retrieve the corresponding $P(D_2|h)$ and $P(D_2|\bar{h})$ values from FFT and GFT, respectively. Let $P(D_2|h) = 0.245$ and $P(D_2|\bar{h}) = 0.289$. By applying Eqs. (17) and (18), we get $P(h|D_2) = 0.51$ and $P(\bar{h}|D_2) = 0.49$. Since $P(h|D_2) \geqslant P(\bar{h}|D_2)$, we compute suspicion score of the current round by D–S combination (Eq. (5)) of current round $P(h) = 0.62$ and the posterior belief $P(h|D_2) = 0.51$. We get $\Psi = 0.81$ which is greater than θ_{UT}. Hence the transaction is declared to be fraudulent and manual confirmation is made with the cardholder. The interesting observation is that although the card is not found to be strictly fraudulent in the two transactions individually, due to Bayesian learning, it is detected as fraudulent by belief update.

It may be noted that suspicion score may sometimes go up for a genuine cardholder. This would represent a situation in which he carries out a number of unusual transactions. Similarly, suspicion score may also come down for a fraudster, which represents a scenario in which the fraudster's behavior matches exactly with the actual cardholder. However, we will show in Section 4 that the system is robust enough to handle deviations from expected patterns to a large extent.

3.4. Implementation environment

The implementation of our FDS has been done in MS-SQL Server 2000. The database consists of a number of tables, the important ones being the *customer* table, *transaction* table, *master* table, *suspect* table, *fraud frequency* table, *good frequency* table and the *caught* table. The *customer* table is used to store the credit card details of genuine cardholders. Transactions are submitted to the *transaction* table and passed to the RBF. The initial belief of each transaction is computed by D–S combination of BPAs of R_1 and R_2. The transactions along with their initial beliefs are logged in the *master* table by a trigger associated with the *transaction* table. Transactions suspected to be fraudulent are logged in the *suspect* table. The belief is updated with each new transaction on a particular card number and whenever $\Psi > \theta_{UT}$, the transactions are logged in the *caught* table.

Stored procedures and triggers were written to facilitate the functioning of this setup. These were used to check the deviation of each transaction from the customer's normal profile, D–S combination of BPAs of rules R_1 and R_2, belief updating, logging transactions into different tables upon insert, implement the THD and the flow of events as given in Algorithm 1.

4. Simulation and results

We demonstrate the effectiveness and usefulness of our FDS by testing it with large scale data. Due to unavailability of real life credit card data or benchmark data set for testing, we developed a simulator to generate synthetic transactions that represent the behavior of genuine cardholders as well as that of fraudsters.

It may be noted that Aleskerov et al. [5] tested the performance of their CARDWATCH system on sets of synthetic data based on Gaussian distribution only. Chan et al. [11] have used skewed distribution to generate a training set of labeled transactions. They have done experiments to determine the most effective training distribution. Li and Zhang [34] have modeled a customer's payment by a Poisson process which can only capture the time gap between two transactions. It is seen that, none of them combine appropriate distributions for generating both the transaction amount and the time gap.

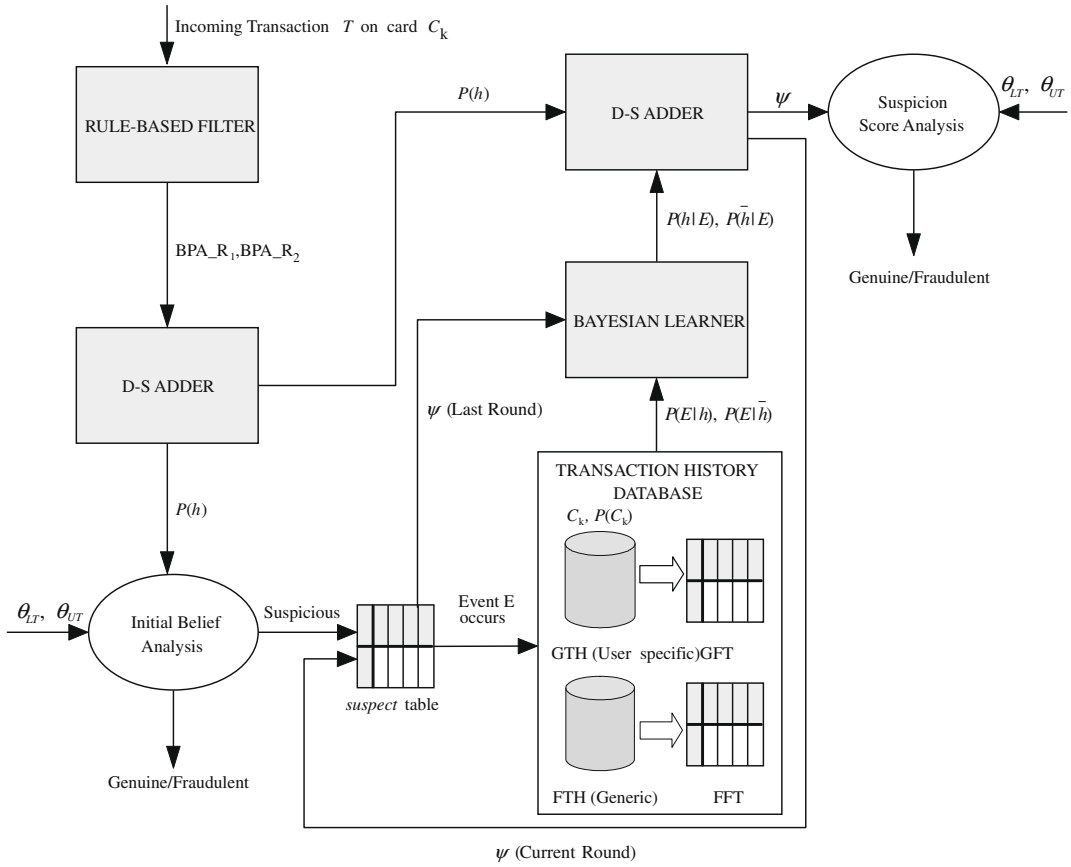

Fig. 1. Block diagram of the proposed fraud detection system.

Table 1
Sample result of Algorithm 1 over various rounds.

Round	$P(h)$	$P(h\|D_i)$	ψ
1	0.55	–	0.55
2	0.62	0.51	0.81

In contrast, our simulator has been designed to handle various real life scenarios that would be normally experienced in a credit card transaction processing application. Firstly, any actual credit card database contains fraudulent transactions interspersed with genuine transactions. Secondly, the genuine transactions are mostly similar for a given customer profile. Thirdly, the genuine transactions and fraudulent transactions are independent events and they have separate arrival rates. We capture these practical situations using a Markov modulated poisson process (MMPP) and two Gaussian distribution functions.

In this section, we first discuss the components of our transaction simulator. Then, we describe the choice of parameters of the proposed FDS in Section 4.2.1 and finally study the performance of the FDS in Section 4.2.2.

4.1. Description of the Simulator

Our simulator has the following three components as shown in Fig. 2:

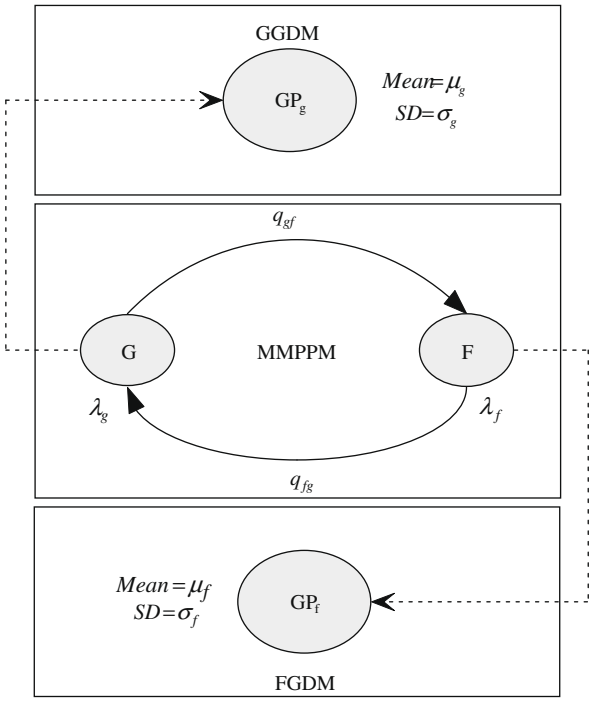

Fig. 2. Transaction simulator.

- Markov modulated poisson process module (MMPPM): It is a Poisson arrival process that has its parameter λ controlled by an underlying Markov process. The proposed MMPPM has two states: good state G and fraud state F with arrival rates λ_g and λ_f, respectively. Transition from G to F takes place with probability q_{gf} and from F to G with probability q_{fg}.

- Genuine Gaussian distribution module (GGDM): We use Gaussian distribution to generate transaction amounts for genuine customers since it is the most commonly observed probability distribution in many natural processes. The simulator can handle different customer profiles by varying mean (μ) and standard deviation (σ). We represent the GGDM by a Gaussian process GP_g having mean μ_g and standard deviation σ_g in Fig. 2.
- Fraud Gaussian distribution module (FGDM): This component is used to generate synthetic transaction amounts for fraudsters and is similar to GGDM. The FGDM is represented by a Gaussian process GP_f with mean μ_f and standard deviation σ_f. The simulator can also handle different categories of fraudsters by varying μ_f and σ_f during generation of fraudulent transactions.

Both the history tables GTH and FTH are initially populated with a large number of transactions. Suppose the number of occurrences of event D_i for a particular card C_k in the GTH is N_i^k, where $i \in \{1,2,3,4\}$ and the total number of transactions in the GTH on that particular card is N. $\frac{N_i^k}{N}$ gives an estimate of the probability $P(D_i|\bar{h})$ for C_k. Higher the number of transactions in the GTH, better is the estimate of this probability. $P(D_i|h)$ is also similarly estimated from the FTH. Variation of the performance of the system with database size is studied in Section 4.2.2.

4.2. Discussion of results

We use standard metrics to study the performance of the system under different test cases. True positives (TP) are the fraudulent transactions caught by the system and false positives (FP) are the genuine transactions labeled as fraudulent (also called false alarms). We first perform a set of experiments to determine a good combination of the design parameters, namely, lower threshold and upper threshold.

4.2.1. Choice of design parameters

From the discussions in Section 3, it is obvious that the effectiveness of the proposed system depends on θ_{LT} and θ_{UT}. If θ_{UT} is set too high, then most of the frauds will go undetected whereas if θ_{UT} is set too low then there will be a large number of false alarms which will lead to serious denial-of-service. Similarly, high value of θ_{LT} will let most of the frauds go through and low value of θ_{LT} will lead to unnecessary investigation of a large number of genuine transactions. Hence, selection of θ_{LT} and θ_{UT} has an associated tradeoff. We, therefore, carried out experiments to determine a good choice of the θ_{LT}, θ_{UT} combination.

Table 2 shows the simulation parameter settings under which the system performance was tested. These combinations capture different groups of fraudsters and genuine users based on the amount and frequency of transactions. For the first few combinations, genuine and fraudulent profiles are quite different. However, the two profiles become similar towards the last few rows of the table.

Table 2
Simulation parameter settings.

Simulator settings	Parameter values					
	λ_g	λ_f	q_{gf}	q_{fg}	μ_g	μ_f
SS1	2	8	0.8	0.2	10	50
SS2	4	8	0.8	0.5	20	50
SS3	2	6	0.5	0.2	10	30
SS4	6	8	0.8	0.8	30	50
SS5	4	6	0.5	0.5	20	30
SS6	2	4	0.2	0.8	10	20
SS7	6	4	0.2	0.8	30	20
SS8	6	6	0.5	0.8	30	30
SS9	4	4	0.2	0.5	20	20

In real life situations, it is usually seen that fraudsters often try to derive maximum benefit from a card by either making high value purchases with longer time gaps or small value purchases at smaller time gaps to avoid detection. The adversaries with this type of spending pattern can be categorized as risk-averse. On the other hand, those in the risk-loving category perform very frequent high value purchases. Similarly, there could also be various behavioral profiles of cardholders. We have considered most of the possible variations that may occur in a real life credit card transaction processing application by setting various arrival rates as well as mean value of transaction amounts as shown in Table 2. In our implementation, we express transaction amount in terms of percentage of credit limit $CL(C_k)$ of a card C_k and the arrival rate in terms of the number of transactions every 72 h. This ensures that all the four events of Eqs. (6)–(9) occur during simulation.

In Table 3, we show the variation of TP/FP for different values of θ_{LT} and θ_{UT}. The values shown in this table represent average of the results obtained for the nine simulator settings of Table 2. Furthermore, for every simulator setting (SSi), we computed the average over 50 independent runs of the simulator consisting of 100 transactions each. Sizes of GTH and FTH were 1000 and 400, respectively.

From Table 3, it is seen that as θ_{LT} increases, TP decreases reaching 78% for $\theta_{LT} = 0.35$. The same trend is true for θ_{UT} also. TP falls to 77% for $\theta_{UT} = 0.85$. FPs also show a similar trend. However, with $\theta_{LT} = 0.3$ and $\theta_{UT} = 0.7$, the difference between TP and FP is the highest. We make this as our choice since it gives a balance between the number of true positives and false positives. Thus, our design parameter settings is $\theta_{LT} = 0.3$ and $\theta_{UT} = 0.7$, which is kept fixed for the rest of the experiments. If a credit card issuing bank wants to be conservative, it can select a θ_{LT}, θ_{UT} combination for which TP is higher. On the other hand, if the target is to reduce denial-of-service situations, θ_{LT}, θ_{UT} combination that gives low FP should be chosen. The effectiveness of the FDS is also dependent on the two parameters ε and *MinPts* of Eq. (1). As discussed in Section 3.1.1, following the heuristic given by Ester et al. [26], we set the parameter $\varepsilon = 2\%$ of credit limit and *MinPts* = 9.

4.2.2. Comparative performance

Once the design parameters have been set, we next show how the system performs as the input is changed. For comparison, we choose the credit card fraud detection system CARDWATCH [5]. Since this system uses features similar to ours, it is the one closest to the proposed approach among all the methods discussed in Section 2. We also study the improvement achieved by using Bayesian learning over and above Dempster's rule of combination.

In Fig. 3, we show variation of mean TP and FP with each SSi of Table 2 for all the three systems as mentioned above. CW denotes CARDWATCH, PA denotes the proposed approach and DS denotes use of DST only. It is seen from Fig. 3 that use of DST itself improves the performance over CARDWATCH by about 15–20% points in TP. Bayesian learning further improves the TP. CARDWATCH on the other hand, has lower FP than only DST based approach. Use of Bayesian learning, however, brings down the FP to values close to 5%.

Table 3
Variation of mean TP/mean FP (%) with θ_{LT} and θ_{UT}.

θ_{UT}	θ_{LT}			
	0.20	0.25	0.30	0.35
0.70	83/8	81/7	81/4	78/4
0.75	82.5/7	80/6.5	79/3.5	76/3
0.80	80/6.5	78/6	78/3	75/2.5
0.85	77/5	75/4	73/3	71/2

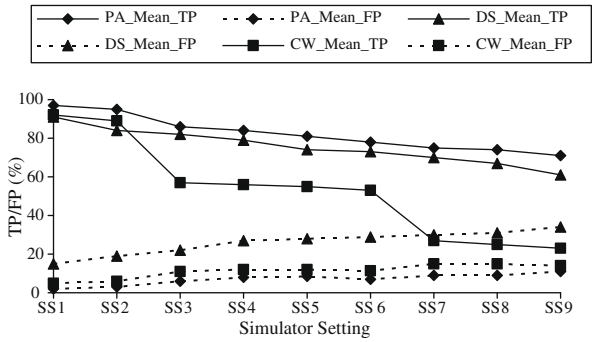

Fig. 3. Variation of mean TP and FP of the proposed approach, D–S part only and CARDWATCH.

Thus, the improvement achieved in the proposed FDS after using Bayesian learning is a substantial reduction in false alarms without compromising the detection rate. It is further observed in the figure that for the simulator setting in which μ_f is high compared to μ_g, TP is high. The reason is that, for these settings the arrival rates of good and bad transactions are quite different. Also, a large percentage of bad transactions in a given transaction mix leads to an overall improvement in TP. However, with each successive SSi, as the genuine behavior becomes similar to fraudster's behavior, the effectiveness reduces (TP decreases and FP increases) for all the three fraud detection systems. Still, the proposed approach shows graceful degradation in accuracy.

We have also studied the performance of the proposed FDS by varying the size of GTH as well as FTH as shown in Fig. 4a and b, respectively. For this set of results, data distribution parameter values are as follows: $\lambda_g = 4$, $\lambda_f = 8$, $q_{gf} = 0.8$, $q_{fg} = 0.5$, $\mu_g = 20$ and $\mu_f = 50$.

It is seen from Fig. 4a that, with increase in the size of the GTH database, the percentage of FP decreases. The reason is that, as the size of this database increases over a period of time, the FDS is able to capture the behavior of a user more consistently. An increase in the size of GTH, however, does not affect the fraud detection rate and hence, the percentage of TP almost remains constant. On the contrary, with an increase in the size of FTH, the FDS is able to capture different possible types of fraudulent patterns, thus increasing the percentage of TP as shown in Fig. 4b. The size of FTH does not, however, affect the genuine transactions due to which the percentage of FP almost remains unchanged. Similar trends were observed for other data distribution parameter settings also.

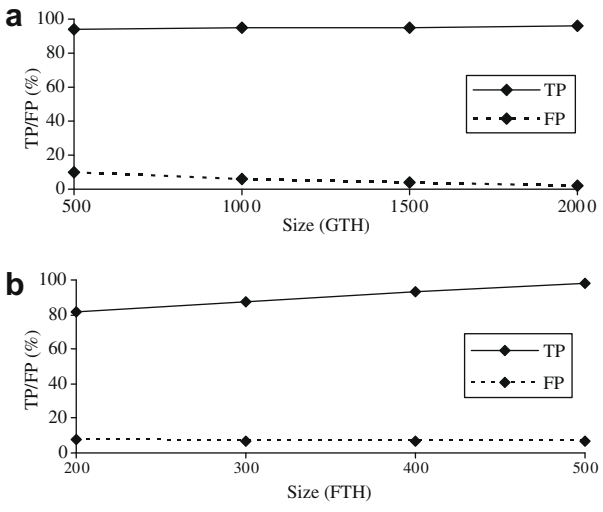

Fig. 4. Variation of TP/FP with (a) size of GTH; (b) size of FTH.

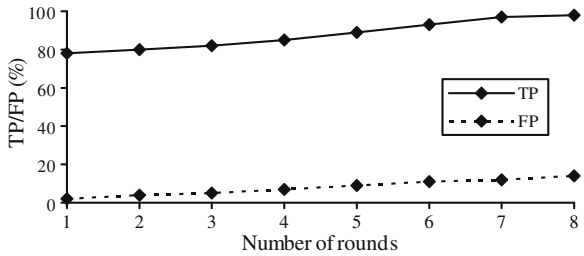

Fig. 5. Variation of TP/FP over successive rounds.

We next study the performance of the proposed FDS over various rounds as shown in Fig. 5. The data distribution parameters are the same as in Fig. 4a and b.

The first round commences with the first suspect transaction on a particular card number. The FDS is able to update the belief values over successive rounds and the process continues as long as the suspicion score is within the two threshold limits. It is seen that with each successive round the percentage of cumulative TP as well as cumulative FP increases. In the example given in Section 3.3 on sample run, the FDS tracks the transactions on the suspicious card and the belief is updated at each round. Finally, the card was caught at the end of the second round.

In Figs. 4 and 5, we only show the results of our proposed approach as these are not relevant for the other two cases.

5. Conclusions

Though most of the fraud detection systems show good results in detecting fraudulent transactions, they also lead to the generation of too many false alarms. This assumes significance especially in the domain of credit card fraud detection where a credit card company needs to minimize its losses but, at the same time, does not wish the cardholder to feel restricted too often. We have proposed a novel credit card fraud detection system based on the integration of three approaches, namely, rule-based filtering, Dempster–Shafer theory and Bayesian learning. Dempster's rule is applied to combine multiple evidences from the rule-based component for computation of initial belief about each incoming transaction. The suspicion score is updated by means of Bayesian learning using history database of both genuine cardholder as well as fraudster. It should be noted that we do not consider any specific fraud model to generate FTH. Instead, FTH is built from history data about past fraudulent behaviors detected by any credit card company. Any other validated fraud model may also be suitably chosen. Moreover, the FDS architecture has been kept flexible so that new rules using any other effective technique can also be included at a later stage to further augment the rule-based component. In addition, Bayesian learning takes place so that the FDS dynamically adapts to the changing behavior of genuine customers as well as fraudsters over time.

We have used stochastic models to generate synthetic transactions for analyzing the performance of the system. The simulation yielded up to 98% TP and less than 10% FP. Comparative studies show significant improvement in accuracy. While combining rules using Dempster–Shafer theory gives good performance, especially in terms of true positives, Bayesian learning helps to further improve the system accuracy. Based on the results, we conclude that fusion of multiple evidences and learning are the appropriate approaches for addressing this type of real world problems where the patterns of behavior are complex and there may be little or no knowledge about the semantics of the application domain.

The system can be further improved by using an extension of DST as proposed in [30] which is more suited for combining conflicting evidences. Possibilities of using other methods for combining evidences like Bayesian combination network may be explored. We could also cluster the transaction gaps to determine separate D_i's for each cardholder. Such card-specific definitions of D_i's can potentially make it more effective. Since we have given all the details of our approach in Section 3 and that of the simulator in Section 4, the experiments can be repeated by interested readers, thus reproducing the results. Though we have tackled a specific application, we feel that with minor application-specific modifications, the present approach can be effectively used to counter intrusion in other database applications as well.

Acknowledgments

We are thankful to the anonymous reviewers for their constructive and useful comments. This work is partially supported by a research grant from the Department of Information Technology, Ministry of Communication and Information Technology, Government of India, under Grant No. 12(34)/04-IRSD dated 07/12/2004.

References

[1] W. Roberds, The impact of fraud on new methods of retail payment, Federal Reserve Bank of Atlanta Economic Review, First Quarter (1998) 42–52.
[2] Statistics for General and Online Card Fraud, 20 June, 2007. <http://epaynews.com/statistics/fraud.html>.
[3] Online fraud is 12 times higher than offline fraud, 20 June, 2007. <http://sellitontheweb.com/ezine/news0434.shtml>.
[4] S. Ghosh, D.L. Reilly, Credit card fraud detection with a neural-network, in: Proceedings of the Annual International Conference on System Science, 1994, pp. 621–630.
[5] E. Aleskerov, B. Freisleben, B. Rao, CARDWATCH: a neural network based database mining system for credit card fraud detection, in: Proceedings of the Computational Intelligence for Financial Engineering, 1997, pp. 220–226.
[6] J.R. Dorronsoro, F. Ginel, C. Sanchez, C.S. Cruz, Neural fraud detection in credit card operations, IEEE Transactions on Neural Networks 8 (July) (1997) 827–834.
[7] M. Syeda, Y.Q. Zhang, Y. Pan, Parallel granular neural networks for fast credit card fraud detection, in: Proceedings of the IEEE International Conference on Fuzzy Systems, 2002, pp. 572–577.
[8] S. Maes, K. Tuyls, B. Vanschoenwinkel, B. Manderick, Credit card fraud detection using Bayesian and neural networks, in: Proceedings of the First International NAISO Congress on Neuro Fuzzy Technologies, 2002.
[9] R.C. Chen, M.L. Chiu, Y.L. Huang, L.T. Chen, Detecting credit card fraud by using questionnaire-responded transaction model based on support vector machines, in: Proceedings of the Fifth International Conference on Intelligent Data Engineering and Automated Learning, vol. 3177, October 2004, pp. 800–806.
[10] R.C. Chen, S.T. Luo, X. Liang, V.C.S. Lee, Personalized approach based on SVM and ANN for detecting credit card fraud, in: Proceedings of the IEEE International Conference on Neural Networks and Brain, October 2005, pp. 810–815.
[11] P.K. Chan, W. Fan, A.L. Prodromidis, S.J. Stolfo, Distributed data mining in credit card fraud detection, in: Proceedings of the IEEE Intelligent Systems, 1999, pp. 67–74.
[12] R. Brause, T. Langsdorf, M. Hepp, Neural data mining for credit card fraud detection, in: Proceedings of the International Conference on Tools with Artificial Intelligence, 1999, pp. 103–106.
[13] C. Chiu, C. Tsai, A web services-based collaborative scheme for credit card fraud detection, in: Proceedings of the IEEE International Conference on e-Technology, e-Commerce and e-Service, 2004, pp. 177–181.
[14] S.J. Stolfo, D.W. Fan, W. Lee, A.L. Prodromidis, Credit card fraud detection using meta-learning: issues and initial results, in: Proceedings of the Workshop on AI Methods in Fraud and Risk Management, 1997, 83–90.
[15] P.K. Chan, S.J. Stolfo, Toward parallel and distributed learning by meta-learning, in: Proceedings of the Workshop on AAAI Workshop in Knowledge Discovery in Databases, 1993, pp. 227–240.
[16] A.L. Prodromidis, S.J. Stolfo, Agent-based Distributed Learning Applied to Fraud Detection, Technical Report CUCS-014-99, 1999.
[17] P. Liu, L. Li, A Game-Theoretic Approach for Attack Prediction, Technical Report, PSU-S2-2002-01, Penn State University, 2002.
[18] V. Vatsa, S. Sural, A.K. Majumdar, A game-theoretic approach to credit card fraud detection, in: Proceedings of the International Conference on Information Systems Security, Lecture Notes in Computer Science, vol. 3803, 2005, pp. 263–276.
[19] C. Phua, V. Lee, K. Smith, R. Gayler, A comprehensive survey of data mining-based fraud detection research, 22 March, 2007. <http://www.bsys.monash.edu.au/people/cphua/>.
[20] Y. Kou, C.T. Lu, S. Sirwonqattana, Y.P. Huanq, Survey of fraud detection techniques, in: Proceedings of the IEEE International Conference on Networking, Sensing and Control, vol. 1, 2004, pp. 749–754.
[21] R.J. Bolton, D.J. Hand, Statistical fraud detection: a review, Journal of Statistical Science (2002) 235–255.
[22] S. Axelsson, The base-rate fallacy and the difficulty of intrusion detection, ACM Transactions on Information and System Security (TISSEC) 3 (2000) 186–205.
[23] B.M. Ayyub, Elicitation of Expert Opinions for Uncertainty and Risks, CRC Press, Boca Raton, 2001.
[24] F. Cremer, E.D. Breejen, K. Schutte, Sensor data fusion for anti-personnel land-mine detection, in: Proceedings of the International Conference on Data Fusion (EuroFusion98), 1998, pp. 55–60.
[25] V. Hodge, J. Austin, A survey of outlier detection methodologies, Journal of Artificial Intelligence Review 22 (2) (2004) 85–126.
[26] M. Ester, H.P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD), 1996, pp. 226–231.
[27] Y. Wang, H. Yang, X. Wang, R. Zhang, Distributed intrusion detection system based on data fusion method, in: Proceedings of the Fifth World Congress on Intelligent Control and Automation, June 2004, pp. 4331–4334.
[28] T.M. Chen, V. Venkataramanan, Dempster–Shafer theory for intrusion detection in ad hoc networks, in: Proceedings of the IEEE Internet Computing, November–December 2005, pp. 35–41.
[29] Z. Yi, H.Y. Khing, C.C. Seng, Z.X. Wei, Multi-ultrasonic sensor fusion for mobile robots, in: Proceedings of the IEEE Intelligent Vehicles Symposium, 2000, pp. 387–391.
[30] F. Campos, S. Cavalcante, An extended approach for Dempster–Shafer theory, in: Proceedings of the IEEE International Conference on Information Reuse and Integration, 2003, pp. 338–344.
[31] G. Shafer, A Mathematical Theory of Evidence, Princeton University Press, Princeton, 1976.
[32] K. Sentz, Combination of Evidence in Dempster–Shafer Theory, Sandia National Laboratories, US Department of Energy, 20 June, 2007. <http://www.sandia.gov/epistemic/Reports/SAND2002-0835.pdf>.
[33] R. Knight, Fraudsters favour brandy and one-way tickets, Financial Times, UK, 20 June, 2007. <http://www.ft.com/cms/s/728ff80c-1698-11da-8081-00000e2511c8.html>.
[34] Y. Li, X. Zhang, Securing credit card transactions with one-time payment scheme, Journal of Electronic Commerce Research and Applications (2005) 413–426.