Credit Card Fraud Detection Using Hidden Markov Model

Arun DV1, M.V Harshavardhan2, Arjun M[3]

[1,2,3]*Dept. of Computer Science and Engineering, Atria Institute of Technology, Bangalore, Karnataka, India*

Abstract **- Due to rapid advancement of computer science technology and commercial, there is a rapid increase in the use of credit cards. As the credit card is the most popular way of purchasing the goods by offline or online. There is a rise of frauds involving the credit cards. In this paper, we have proposed Hidden Markov Model (HMM) to detect the fraudulent transactions. It is trained with the card holder's spending habits If it rejected by the system, then transaction is considered as fraudulent. At the same time, no genuine transaction should be rejected.**

Index Terms **- Hidden Markov Model, Credit card, fraudulent transaction and genuine transaction, mode of payment, offline and online.**

I.INTRODUCTION

The popularity of online shopping is growing day by day. Due to the increase of popularity, the most popular method of payment is credit card. Retailers such as Walmart, Amazon handle much larger number of credit card transactions including online and regular transactions. The credit card user are increases worldwide, the opportunities of attacks by fraudsters also increase. The total credit card fraud in India itself is reported to be Rs 128cr in 2019 and Rs127cr in 2017 are the estimates of online Fraud.

Credit Card based purchase are categorized into two types: 1) physical card and 2) virtual card. Physical card is provided by the user to the merchant for making payment. To carry out the fraudulent transaction the fraudster should have the details of the card holder by the stealing the card from the card holder. The card holder does not realize the loss of his/her card. It can lead to loss of financial to the credit card company. Second kind, the purchase of goods is done through the internet or over the telephone. The purchase done through the client inputting the card's details such as card number, security code and expiration date. To commit fraud, fraudster needs to know the card's details.

Most time, the genuine cardholder may not realize that someone has stolen or seen his/her credit card information. The only way to detect this kind of fraud is to analyze the spending pattern on every card and figuring out the inconsistency with the usual spending pattern. Fraud detection is based on analyzing of the existing purchase data of the cardholder is the promising way to reduce the fraudulent transactions. Since Humans tend to exhibit the specific behaviorist profiles. The card holders can be represented the set of patterns containing information about the typical purchase category, the amount spent, date and time of the purchase. Deviation from patterns can be identified as the potential threat to the system.

II. CREDIT CARD FRAUD DETECTION

Credit card Fraud Detection has drawn a lot of interest in the field of research and number of techniques with special emphasis on data mining and neural network that have been suggested. The problem of the approaches is that they require label data of both genuine and fraudulent transactions, to train the classifiers. Getting real-world fraud data is one of the biggest problems associated with the credit card fraud detection. These approaches cannot detected new kinds of frauds without any labeled data which is not available. We have proposed the Hidden Markov model (HMM) based credit card fraud detection which does not require fraud signatures and yet is able to detect the fraud by considering the cardholders spending habits. HMM based credit card fraud detection approach will reduce the false positive transactions which are identified as malicious transaction by the fraud detection system (FDS) although the transaction is genuine. Since the number of genuine transaction are more than the transaction which are fraudulent. A Fraud detection system should be designed in such way that the number of transactions which are indentified as the false positive (FP) should be low as possible

A. Fraud Detection System: All the information about the credit card such as card number, name of the cardholder, expiry date and validity year. When the user enters his/her card details such as Personal information identity number (PIN) should be entered when purchasing the items or goods from the merchant either by physical card or online payment. PIN will be checked system with the given account, the fraud check module will be activated. The purchase amount will be checked with the user spending habits. If the transaction is genuine, the transaction will be allowed to continue or else the system will detected the fraudulent transaction and rejected.

The below figure shows th¼ working of hidden Markov module in our proposed system:

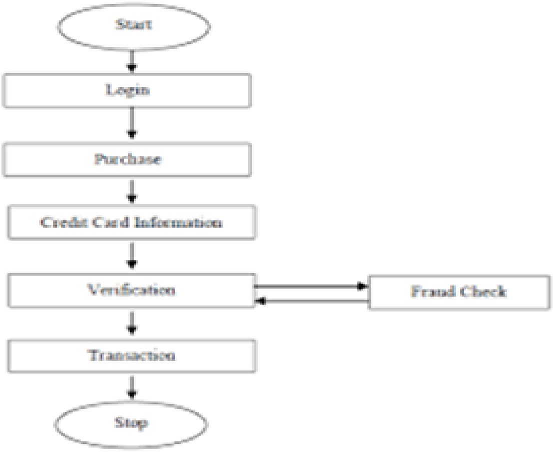


Fig 1: The flow chart of HMM for credit card fraud detection.

III. HMM BACKGROUND

An HMM is a double embedded stochastic process with two hierarchy levels. It can be used to model much more complicated stochastic processes as compared to a traditional Markov model.

A HMM has a finite set of states governed by a set of transition probabilities. In a particular state, an outcome or observation can be generated according to an associated probability distribution. It is only the outcome and not the state that is visible to a next observer.

An HMM can be characterized by the following

1. N is the number of states in the model. We denote the set of states $S = \{S_1,S_2,S_3,..,S_N\}$,where S_i , i = 1,2,...,N is an individual state. The state at time instant t is denoted by q_t.

2.M is the number of distinct observation symbols per state. The observation symbols correspond to the physical output of the system being modeled. We denote the set of symbols, $V = \{V_1,V_2,\ldots,V_M\}$ where V_i, i =1,2..,M is an individual symbol.

3. The state transition probability matrix $A = [\alpha_{ij}]$, where, $\alpha_{ij} = P(q_{t+1} = S_j |q_t = S_i)$, $1 \le i \le N$, $1 \le j \le N$.

For the general case where any state j can be reached from any other state I in a single step, we have a_{ij}>P.

4. The observation symbol probability matrix $B = [b_j(k)]$ where, $bj(k) = 1$, $1 \le j \le N$, $1 \le k \le M$ and

5. The initial probability vector $\pi = [\pi_i]$, where $\pi_i = P(q_1 = S_i)$,

$\pi_i = P(q1 = Si)$, $1 \le i \le N$

6.The observation sequence *O* $O_1; O_2; O_3; \ldots O_R$, where each observation is one of the symbols from *V*, and *R* is the number of observations in the sequence.

IV.USE OF HMM FOR CREDIT CARD FRAUD DETECTION

An FDS runs at a credit card issuing bank. Each incoming transaction is submitted to the FDS for verification. FDS receives the card details and the value of purchase to verify whether the transaction is genuine or not. The types of goods that are bought in that transaction are not known to the FDS. It tries to find any anomaly in the transaction based on the spending profile, of the cardholder, shipping address, and billing address etc. If the FDS confirms the transaction to be malicious, it raises an alarm, and the issuing bank declines the transaction. The concerned cardholder may then be contacted and alerted about the possibility that the card is compromised. In this section, we explain how HMM can be used for credit card fraud detection.

We use V_k, *k* ¼ 1*;* 2*;* ... *M*, to represent both the observation symbol, as well as the corresponding price range. In this work, we consider only three price ranges, namely, low *l* , medium *m* , and high *h* .Our set of observation symbols is ,therefore, *l;m;* making *M*3. For example, let *l* 0*;*$100,*m*$100*;*$500,and *h*$500*;* credit card limit. If a card holder performs a transaction of $190, then the corresponding observation symbol is *m*.

A credit cardholder makes different kinds of purchases of different amounts over a period of time. One possibility is to consider the sequence of transaction amounts and look for deviations in them. However, the sequence of types of purchase is more stable compared to the sequence of transaction amounts. The reason is that, a cardholder makes purchases depending on his need for procuring different types of items over a period of time.

This, in turn generates a sequence of transaction amounts. Each individual transaction amount usually depends on the corresponding type of purchase. Hence, we consider the transition in the type of purchase as state transition in our model. The type of each purchase is linked to the line of business of the corresponding merchant. This information about the merchant's line of business is not known to the issuing bank running the FDS.

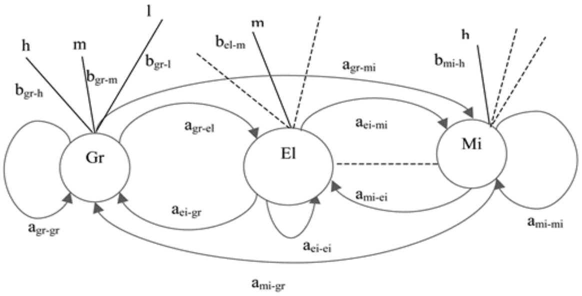


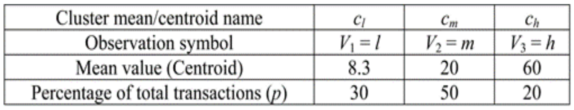
Fig 2: HMM for credit card fraud detection

Consider the sequence of transaction amounts and look for deviations in them. However, the sequence of types of purchase is more stable compared to the sequence of transaction amounts. The reason is that, a cardholder makes purchasing depending on his need for procuring different types of items over a period of time. This, in turn, generates a sequence of transaction amounts. Each individual transaction amount usually depends on the corresponding type of purchase.

4.1. Model Parameter Estimation and Training

We use Baum-Welch algorithm to estimate the HMM parameters for each cardholder. The algorithm starts with an initial estimate of HMM parameters A, B, and r and converges to the nearest local maximum of the likelihood function. Initial state probability distribution is considered to be uniform, that is, if there are N states, then the initial probability of each state is 1=N. Initial guess of transition and observation probability distributions can also be considered to be uniform. However, to make the initial guess of observation symbol probabilities more accurate, spending profile of the cardholder, as determined in Section 4, is taken into account.

We make three sets of initial probability for observation symbol generation for three spending groups—ls, ms, and hs. Based on the cardholder's spending profile, we choose the corresponding set of initial observation probabilities. The initial estimate of symbol generation probabilities using this method leads to accurate learning of the model. Since there is no a priori knowledge about the state transition probabilities, we consider the initial guesses to be uniform. In case of a collaborative work between an acquiring bank.



Cluster mean/centroid name	c_l	c_m	c_h
Observation symbol	$V_1 = l$	$V_2 = m$	$V_3 = h$
Mean value (Centroid)	8.3	20	60
Percentage of total transactions (p)	30	50	20

Fig3: Output of K-Means Clustering Algorithm

We now start training the HMM. The training algorithm has the following steps:

1) Initialization of HMM parameters,

2) Forward procedure, and

3) Backward procedure.

Details of these steps can be founded. For training the HMM, we convert the cardholder's transaction amount into observation symbols and form sequences out of them. At the end of the training phase, we get an HMM corresponding to each cardholder. Since this step is done offline, it does not affect the credit card transaction processing performance, which needs online response.

V. FRAUD DETECTION

After the HMM parameters are learned, we take the symbols from a cardholder's training data and form an initial sequence of symbols. Let $O_1; O_2; ... O_R$ be one such sequence of length *R*. This recorded sequence is formed from the cardholder's transactions up to time *t*. We input this sequence to the HMM and compute the probability of acceptance by the HMM. Let the probability be α_1, which can be written as follows:

$$\alpha_1 = P(O_1,O_2,O_3,..,O_R|ʎ)$$

Let $O_{Rþ1}$ be the symbol generated by a new transaction at time *t* þ 1. To form another sequence of length *R*, we drop O_1 and append $O_{Rþ1}$ in that sequence, generating $O_2; O_3; \ldots O_R; O_{R\text{þ}1}$ as the new sequence. We input this new sequence to the HMM and calculate the probability of acceptance by the HMM. Let the new probability be $\alpha_{2.}$

$$\alpha_2 = P(O_2, O_3, O_4, .., O_{R+1} | ʎ)$$

$$\text{Let } \Delta\alpha = \alpha 1 - \alpha 2$$

If $\Delta 2 > 0$, it means that the new sequence is accepted by the HMM with low probability, and it could be a fraud. The newly added transaction is determined to be fraudulent if the percentage change in the probability is above a threshold, that is,

$$\Delta\alpha/\alpha_1 \geq \text{Threshold}$$

The threshold value can be learned empirically, as will be discussed in Section 5. If $O_{R\text{þ}1}$ is malicious, the issuing bank does not approve the transaction, and the FDS discards the symbol. Otherwise, $O_{R\text{þ}1}$ is added in the sequence permanently, and the new sequence is used as the base sequence for determining the validity of the next transaction. The reason for including new non malicious symbols in the sequence is to capture the changing spending behavior of a cardholder. Fig. 2 shows the complete process flow of the proposed FDS. As shown in the figure, the FDS is divided into two parts—one is the training module, and the other is detection.

Training phase is performed offline, whereas detection is an online process.

5.1. Spending Profile of Cardholders

The spending profile of a cardholder suggests his normal spending behavior. Cardholders can be broadly categorized into three groups based on their spending habits, namely, high-spending (hs) group, medium-spending (ms) group, and low-spending (ls) group. Cardholders who belong to the hs group, normally use their credit cards for buying high- priced items. Similar definition applies to the other two categories also.

Spending profiles of cardholders are determined at the end of the clustering step. Let p_i be the percentage of total number of transactions of the cardholder that belong to cluster with mean c_i. Then, the spending profile (SP) of the cardholder u is determined as follows:

$$SP(u) = \underset{i}{\arg\max}(p_i)$$

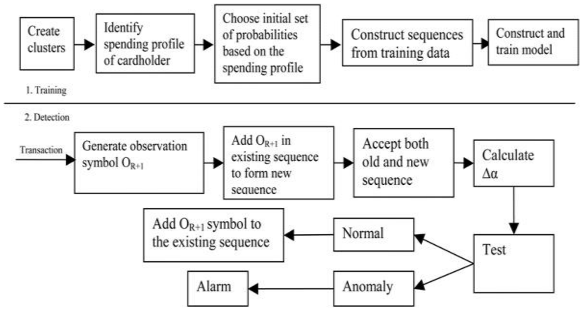

Create clusters
Identify spending profile of cardholder
Choose initial set of probabilities based on the spending profile
Construct sequences from training data
Construct and train model
1. Training
2. Detection
Transaction
Generate observation symbol O_{R+1}
Add O_{R+1} in existing sequence to form new sequence
Accept both old and new sequence
Calculate Δα
Test
Normal
Anomaly
Add O_{R+1} symbol to the existing sequence
Alarm

Fig4: Process flow of the proposed FDS

VI. RESULTS

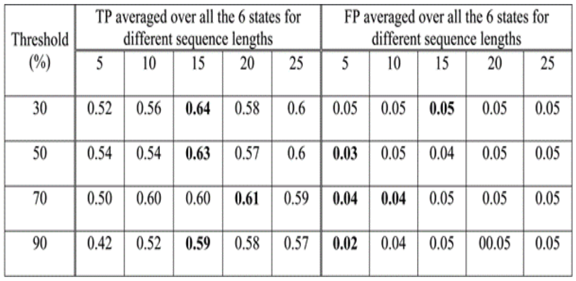
Testing credit card FDSs using real data set is a difficult task. Banks do not, in general, agree to share their data with researchers. There is also no benchmark data set available for experimentation. We have, therefore, performed large-scale simulation studies to test the efficacy of the system. A simulator is used to generate a mix of genuine and fraudulent transactions. The number of fraudulent transactions in a given length of mixed transactions is normally distributed with a user specified μ (mean) and σ (standard deviation), taking cardholder's spending behavior into account. μ specifies the average number of fraudulent transactions in a given transaction mix. In a typical scenario, an issuing bank, and hence, its FDS receives a large number of genuine transactions sparingly intermixed with fraudulent transactions.

The genuine transactions are generated according to the cardholders' profiles. The cardholders are classified into three categories as mentioned before—the low, medium, and hs groups. We have studied the effects of spending group and the percentage of transactions that belong to the low-, medium-, and high-price-range clusters. We use standard metrics—True Positive (TP) and FP, as well as TP-FP spread and Accuracy metrics, as proposed in [7] to measure the effectiveness of the system. TP represents the fraction of fraudulent transactions correctly identified as fraudulent, whereas FP is the fraction of genuine transactions identified as fraudulent. Most of the design choices for a FDS that result in higher values of TP, also cause FP to increase. To meaningfully capture the performance of such a system, the difference between TP and FP, often called the TP-FP spread, is used as a metric. Accuracy represents the fraction of total number of transactions (both genuine and fraudulent) that have been detected correctly. It can be expressed as follows:

We first carried out a set of experiments to determine the correct combination of HMM design parameters, namely, the number of states, the sequence length, and the threshold value. Once these parameters were decided, we performed comparative study with another FDS.

5.1 Choice of Design Parameters

Since there are three parameters in an HMM, we need to vary one at a time keeping the other two fixed, thus generating a large number of possible combinations. For choosing the design parameters, we generate transaction sequences using 95 percent low value, 3 percent medium value, and 2 percent high value transactions. The reason for using this mix is that it represents a profile that strongly resembles a ls customer profile. We also consider the μ and σ values to be 1.0 and 0.5, respectively. This is chosen so that, on the average, there will be 1 fraudulent transaction in any incoming sequence with some scope for variation. After the parameter values are fixed, we will see in Section 5.2, how the system performs as we vary the profile and the mix of fraudulent transactions.



Threshold (%)	TP averaged over all the 6 states for different sequence lengths					FP averaged over all the 6 states for different sequence lengths				
	5	10	15	20	25	5	10	15	20	25
30	0.52	0.56	**0.64**	0.58	0.6	0.05	0.05	**0.05**	0.05	0.05
50	0.54	0.54	**0.63**	0.57	0.6	**0.03**	0.05	0.04	0.05	0.05
70	0.50	0.60	0.60	**0.61**	0.59	**0.04**	**0.04**	0.05	0.05	0.05
90	0.42	0.52	**0.59**	0.58	0.57	**0.02**	0.04	0.05	00.05	0.05

Variation of TP and FP with Different Sequence

We have also analyzed the time taken by the training phase, which is performed offline for each cardholder's HMM. Fig. 4 shows the plot of model learning time against the number of sequences in the training data. As the size of training data increases, learning time increases, especially beyond 100. We therefore, use 100 sequences for training the HMM. Although done offline, the model learning time has a strong impact on the scalability of the system. Since an HMM is trained for each cardholder, it is imperative that the training time is kept as low as possible especially when an issuing bank is meant to handle millions of cardholders with many new cards being issued everyday,

The online processing time of about 200 ms on a 1.8 GHz

Pentium IV machine also shows that the system will be able to handle a large number of concurrent operations and, hence, is scalable.

5.2 Comparative Performance

In this section, we show performance of the proposed system as we vary the number of fraudulent transactions and also the spending profile of the cardholder. Our design parameter setting is as obtained in the previous section. We compare performance of our approach (denoted by OA below) with the credit card fraud detection technique proposed by Stolfo et al.(denoted by ST below). For comparison, we consider the metrics TP and FP, as well as TP-FP and Accuracy.

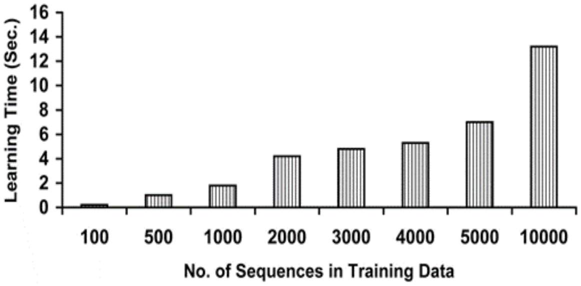


Fig5: Model learning time versus number of sequences in training data

We carried out experiments by varying both the transaction amount mix, as well as the number of fraudulent transactions intermixed with a sequence of genuine transactions. Transaction amount mix is captured by the cardholder's profile. We consider four profiles. One of them is the mixed profile, which means that spending profile is not considered at all by our approach, as explained in Section 5.1. The other profiles considered are (55 35 10),(70 20 10), and (95 3 2). Here, a b c profile represents a is profile cardholder who has been found to carry out a percent of his transactions in the low, b percent in medium, and c percent in the high range. Thus, our attempt is to see how the system performs in the presence of different mixes of transaction amount ranges in the transactions. It may be noted that for cardholders in the other two groups, namely, hs and ms, will show similar performance as only the relative ordering of a, b, and c will change. We also vary the mean value μ of malicious transactions from 0.5 to 4.0 in steps of 0.5. The σ value is kept fixed at 0.5 for all the experiments. Thus, every sequence of transaction that we use for testing is a mixed sequence containing both genuine, as well as malicious, transactions. For each combination of spending profile and malicious transaction distribution, we carried out 100 runs and report the average result. The same set of data was used to determine the performance of both OA and ST.

From the above results, it can be concluded that the proposed system has an overall Accuracy of 80 percent even under large input condition variations, which is much higher than the overall Accuracy of the method proposed by Stolfo et al. Our system can, therefore, correctly detect most of the transactions. However, when there is no profile information at all, the system shows some performance degradation in terms of TP-FP. This observation highlights the importance of profile selection as explained in Section 5. Also, when there is little difference between genuine transactions and malicious transactions, most of the credit card FDSs suffer performance degradation, either due to a fall in the number of TPs or a rise in the number of FPs.

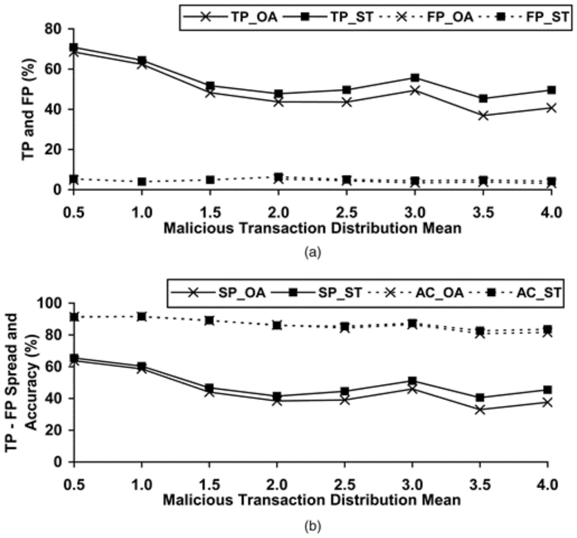


Fig6: Performance variation of two systems (OA and ST) with the mean of malicious transaction distribution for the spending profile (a) TP and FP. (b) TP-FP spread(SP) and Accuracy(AC).

VI. CONCLUSIONS AND DISCUSSIONS

In this paper, we have proposed an application of HMM in credit card fraud detection. The different steps in credit card transaction processing are represented as the under- lying stochastic process of an HMM. We have used the ranges of transaction amount as the observation symbols, whereas the types of item have been considered to be states of the HMM. We have suggested a method for finding the spending profile of cardholders, as well as application of this knowledge in deciding the value of observation symbols and initial estimate of the model parameters. It has also been explained how the HMM can detect whether an incoming transaction is fraudulent or not. Experimental results show the performance and effectiveness of our system and demonstrate the usefulness of learning the spending profile of the cardholders. Comparative studies reveal that the Accuracy of the system is close to 80 percent over a wide variation in the input data. The system is also scalable for handling large volumes of transactions.

REFERENCES

[1] “Global Consumer Attitude Towards On-Line Shopping,” http://www2.acnielsen.com/reports/documents/2005_cc_online shopping.pdf, Mar. 2007.

[2] D.J. Hand, G. Blunt, M.G. Kelly, and N.M. Adams, “Data Mining for Fun and Profit,” Statistical Science, vol. 15, no. 2, pp. 111-131, 2000.

[3] “Statistics for General and On-Line Card Fraud,” http://www. epaynews.com/statistics/fraud.html, Mar. 2007.

[4] S. Ghosh and D.L. Reilly, “Credit Card Fraud Detection with a Neural-Network,” Proc. 27th Hawaii Int’l Conf. System Sciences: Information Systems: Decision Support and Knowledge-Based Systems, vol. 3, pp. 621-630, 1994.

[5] M. Syeda, Y.Q. Zhang, and Y. Pan, “Parallel Granular Networks for Fast Credit Card Fraud Detection,” Proc. IEEE Int’l Conf. Fuzzy Systems, pp. 572-577, 2002.

[6] S.J. Stolfo, D.W. Fan, W. Lee, A.L. Prodromidis, and P.K. Chan, “Credit Card Fraud Detection Using Meta-Learning: Issues and Initial Results,” Proc. AAAI Workshop AI Methods in Fraud and Risk Management, pp. 83-90, 1997.

[7] S.J. Stolfo, D.W. Fan, W. Lee, A. Prodromidis, and P.K. Chan, “Cost-Based Modeling for Fraud and Intrusion Detection: Results from the JAM Project,” Proc. DARPA Information Survivability Conf. and Exposition, vol. 2, pp. 130-144, 2000.

[8] R. Brause, T. Langsdorf, and M. Hepp, "Neural Data Mining for Credit Card Fraud Detection," Proc. IEEE Int'l Conf. Tools with Artificial Intelligence, pp. 103-106, 1999.

[9] C. Phua, D. Alahakoon, and V. Lee, "Minority Report in Fraud Detection: Classification of Skewed Data," ACM SIGKDD Explora- tions Newsletter, vol. 6, no. 1, pp. 50-59, 2004.

[10] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proc. IEEE, vol. 77, no. 2, pp. 257-286, 1989.

[11] T. Lane, "Hidden Markov Models for Human/Computer Interface Modeling," Proc. Int'l Joint Conf. Artificial Intelligence, Work-shop Learning about Users, pp. 35-44, 1999.