ISSN:0975-9646

Hidden Markov Model for Credit Card Fraud Detection

Ankit Vartak[#1], Chinmay D Patil[*2],Chinmay K Patil[#3]

#Vidyavardhini's College of Engineering & Technology,
Mumbai,Maharashtra,India
**Viva Institute of Technology,*
Mumbai,Maharashtra,India

Abstract-Due to a rapid advancement in the electronic commerce technology, the use of credit cards has dramatically increased. As credit card becomes the most popular mode of payment for both online as well as regular purchase, cases of fraud associated with it are also rising. In this paper, we model the sequence of operations in credit card transaction processing using a Hidden Markov Model (HMM) and show how it can be used for the detection of frauds. An HMM is initially trained with the normal behavior of a cardholder. If an incoming credit card transaction is not accepted by the trained HMM with sufficiently high probability, it is considered to be fraudulent. At the same time, we try to ensure that genuine transactions are not rejected. We present detailed experimental results to show the effectiveness of our approach and compare it with other techniques available in the literature.

1. INTRODUCTION

In day to day life credit cards are used for purchasing goods and services by the help of virtual card for online transaction or physical card for offline transaction. In physical transaction, Credit cards will insert into payment machine at merchant shop to purchase goods. Tracing fraudulent transactions in this mode may not be possible because the attacker already steal the credit card. The credit card company may go in financial loss if loss of credit card is not realized by credit card holder. In online payment mode, attackers need only little information for doing fraudulent transaction. To commit fraud in these types of purchases, a fraudster simply needs to know the card details (secure code, card number, expiration date etc.). Most of the time, the genuine cardholder is not aware that someone else has seen or stolen his card information. In this purchase method, mainly transactions will be done through Internet or telephone. Small transactions are generally undergo less verification, and are less likely to be checked by either the card issuer or the merchant. Card issuers must take more precaution against fraud detection and financial losses. Credit card fraud cases are increasing every year. In 2008, number of fraudulent through credit card had increased by 30 percent because of various ambiguities in issuing and managing credit cards. Credit card fraudulent is approximately 1.2% of the total transaction amount.

Hidden Markov Model will be helpful to find out the fraudulent transaction by using spending profiles of user. It works on the user spending profiles which can be divided into major three types such as -

1) Lower profile
2) Middle profile
3) Higher profile

The only way to detect this kind of fraud is to analyze the spending patterns on every card and to figure out any inconsistency with respect to the "usual" spending patterns. Fraud detection based on the analysis of existing purchase data of cardholder is a promising way to reduce the rate of successful credit card frauds. Since humans tend to exhibit specific behaviorist profiles, every cardholder can be represented by a set of patterns containing information about the typical purchase category, the time since the last purchase, the amount of money spent, etc. Deviation from such patterns is a potential threat to the system.

For every credit card, the spending profile is different, so it can figure out an inconsistency of user profile and try to find fraudulent transaction. It keeps record of spending profile of the card holder by both way, either offline or online. Thus analysis of purchased commodities of cardholder will be a useful tool in fraud detection system and it is assuring way to check fraudulent transaction, although fraud detection system does not keep records of number of purchased goods and categories. The set of information contains spending profile of card holder, money spent in every transaction, the last purchase time, category of purchase etc. The potential threat for fraud detection will be a deviation from set of patterns. Several techniques for the detection of credit card fraud have been proposed in the last few years.

2. HMM (HIDDEN MARKOV MODEL) BACKGROUND:

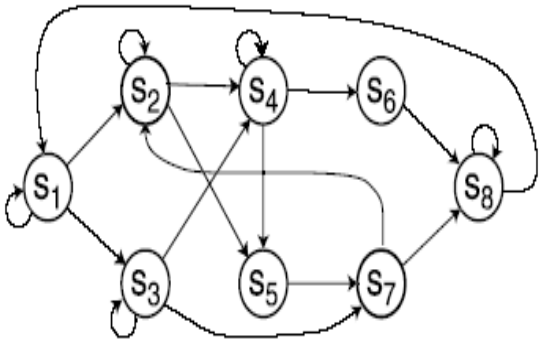
An HMM is a double embedded stochastic process with two hierarchy levels which can be based to model much more complicated stochastic processes as compared to a traditional Markov model. An HMM has a finite set of states governed by a set of transition probabilities. In a given state, an outcome can be generated according to an associated probability distribution only the outcome that is visible to an external observer and not the state.

HMM can be characterized by the following:

1. N is the number of states in the model.

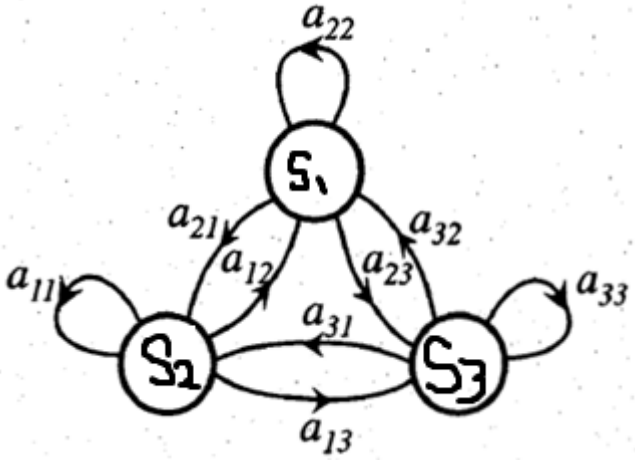
The set of states is denoted as S={S1 ; S2 ; . . SN} N is an individual state.

The state at time instant t is denoted by qt.



In the above diagram we have a HMM with 8 stated state S1 to state S8.

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= {V1 ; V2 ; . . . VM}
   Eg:- in the above diagram we can say that the observation symbol of the state S1 is V1 state S2 is V2 and so on till state S8 is V8.
3. The state transition probability A= $[a_{ij}]$, where
   $a_{ij}= P(q_{t+1} =S_j|q_t=S_i),\ 1<=i<=N,\ 1<=j<=N;\ t=1,2,\ldots.$ (1)
   For the general case where any state j can be reached from any other state i in a single step, we have $a_{ij} >0$ for all i, j. Also, $\sum^{N}_{i=1} a_{ij}=1,\ 1<=i<=N.$



The above diagram shows a hmm with 3states and aij as the state transition probabilities.

4. The observation symbol probability matrix B= $[b_j(k)]$, where
   $b_j(k)= P(V_k|S_j),\ 1<=j<=N,\ 1<=k<=M$ and $\sum^{M}_{k=1} b_j(k)= 1,\ 1<=j<=N.$ (2)
5. The initial state probability vector $\pi = [\pi_i]$, where
   $\Pi_i =P(q_1=S_i),\ 1<=i<=N,$ such that $\sum^{N}_{i=1} \Pi_i =1$ (3)
6. The observation sequence $O = O_1, O_2, O_3,...$ OR, where each observation O_t is one of the symbols from V, and R is the number of observations in the sequence.

Notation	Meaning
M	Number of observation symbols
N	Number of hidden states
V_k k=1..M	Observation symbols
l, m, h	Price ranges – low, medium and high
$a_{x\text{-}y}$	Probability of transition from the Hidden Markov Model state x to state y
K	Number of clusters
c_i	Centroid of cluster i
R	Sequence length
α	Probability of acceptance of a sequence by Hidden Markov Model
Acronym	**Expanded Form**
FDS	Fraud Detection System
HMM	Hidden Markov Model
SP	Spending Profile
hs, ms, ls	High spending, Medium spending, Low spending group
TP, FP	True Positive, False Positive

It is evident that a complete specification of an HMM requires the estimation of two model parameters, N and M, and three probability distributions A, B, and π. We use the notation $\lambda = (A, B, \pi)$ to indicate the complete set of parameters of the model, where A, B implicitly include N and M.

An observation sequence O, as mentioned above, can be generated by many possible state sequences. Consider one such particular sequence

$Q=q_1,q_2,\ldots\ldots,q_R.$ (4)

where q_1 is the initial state.

The probability that O is generated from this state sequence is given by

$P(O|Q,\lambda) = \pi^{R}_{t=1} P(O_t|q_t, \lambda),$ (5)

where statistical independence of observations is assumed.

Equation (5) can be expanded as

$P(O|Q,\lambda)= b_{q1}(O_1).b_{q2}(O_2)\ldots.b_{qR}(O_R).$ (6)

The probability of the state sequence Q is given as

$P(Q|\lambda)=\pi_{q1,}a_{q1q2,}a_{q2q3}\ldots\ldots.a_{qR\text{-}1,qR}$ (7)

Thus, the probability of generation of the observation sequence O by the HMM specified by λ can be written as follows:

$P(O|\lambda) = \sum_{all\ Q} P(O|Q,\lambda)\ P(Q|\lambda).$

Deriving the value of $P(O|\lambda)$ using the direct definition of (8) is computationally intensive. Hence, a procedure named as Forward-Backward procedure is used to compute $P(O|\lambda)$.

3. USE OF HMM FOR CREDIT CARD FRAUD DETECTION:

A FDS (Fraud Detection System) runs at a credit card issuing bank. It is sent the card details and the value of purchase to verify whether the transaction is genuine or not and tries to find any anomaly in the transaction. This calculation is based on spending profile of the cardholder, shipping address, and billing address, etc. If found to be fraudulent, it raises an alarm.

3.1 HMM MODEL FOR CREDIT CARD TRANSACTION PROCESSING:

To map the credit card transaction processing operation in terms of an HMM, we start by first deciding the observation symbols in our model. We quantize the purchase values x into

M price ranges V1;V2;... VM, forming the observation symbols at the issuing bank. ie map the amount into observation sybols based on clustering.

In our work, we consider only three price ranges, namely, low (l), medium (m), and high (h). Our set of observation symbols is, therefore, V ={l,m,h} making M =3.For example, let l ={0, 100}, m ={100, 500} and h ={500, limit of card}.If a cardholder performs a transaction of 190, then the corresponding observation symbol is m.

The set of all possible types of purchase amount forms the set of hidden states of the HMM. The actual items purchased in the transaction are not determined. After deciding the state and symbol representations, the next step is to determine the probability matrices A, B, and π so that representation of the HMM is complete. These three model parameters are determined in a training phase using the Baum-Welch algorithm. We consider the special case of fully connected HMM in which every state of the model can be reached in a single step from every other state also known as ergodic model of HMM.

3.2 DYNAMIC GENERATION OF OBSERVATION SYMBOLS

For each cardholder, we train and maintain an HMM. To find the observation symbols corresponding to individual cardholder's transactions dynamically, we run a clustering algorithm on his past transactions. Normally, the transactions that are stored in the issuing bank's database contain many attributes. For our work, we consider only the amount that the cardholder spent in his transactions. Although various clustering techniques could be used, we use K-means clustering algorithm [24] to determine the clusters. K-means is an unsupervised learning algorithm for grouping a given set of data based on the similarity in their attribute (often called feature) values. Each group formed in the process is called a cluster. The number of clusters K is fixed a priori. The grouping is performed by minimizing the sum of squares of distances between each data point and the centroid of the cluster to which it belongs.

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 pi be the percentage of total number of transactions of the cardholder that belong to cluster with mean ci. Then, the spending profile (SP) of the cardholder u is determined as follows:

$$SP(u) = \arg\max(p_i).$$

In our work, K is the same as the number of observation symbols M. Let $c_1, c_2, \ldots c_M$ be the centroids of the generated clusters. These centroids or mean values are used t o decide the observation symbols when a new transaction comes in. Let x be the amount spent by the cardholder u in transaction T. FDS generates the observation symbol for x (denoted by Ox) as follows:

$$O_x = V_{\arg\min|x-ci|}.$$

As mentioned before, the number of symbols is 3 in our system. Considering M =3, if we execute K-means algorithm on the example transactions in Table 2, we get the clusters, as shown in Table 3, with c_l, c_m, and c_h as the respective centroids. It may be noted that the dollar amounts 5, 10, and 10 have been clustered together as c_l resulting in a centroid of 8.3. The percentage (p) of total number of transactions in this cluster is thus 30 percent. Similarly dollar amounts 15, 15, 20, 25, and 25 have been grouped in the cluster cm with centroid 20, whereas amounts 40 and 80 have been grouped together in cluster c_h. c_m and c_h, thus, contain 50 percent and 20 percent of the total number of transactions. When the FDS receives a transaction T for this cardholder, it measures the distance of the purchase amount x with respect to the means c_l, cm, and c_h to decide (using (9)) the cluster to which T belongs and, hence, the corresponding observation symbol. As an example, if x =$10, then in Table 3 using (9), the observation symbol is V_1 =l.

TABLE 2
Example Transactions with the Dollar Amount Spent in Each Transaction

Transaction no.	1	2	3	4	5	6	7	8	9	10
Dollar Amount	40	25	15	5	10	25	15	20	10	80

TABLE 3
Output of K-Means Clustering Algorithm

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

3.3 SPENDING PROFILE OF CARDHOLDERS

Thus, spending profile denotes the cluster number to which most of the transactions of the cardholder belong.

A credit cardholder makes different kinds of purchases of different amounts over a period of time. The sequence of types of purchase is more stable compared to the sequence of transaction amounts.. Dynamic Generation of Observation Symbols: We train and maintain an HMM for each cardholder the amount that the cardholder spent in his transactions are determined from the bank database and K-means clustering algorithm to determine the clusters. The grouping is performed by minimizing the sum of squares of distances between each data point and the centroid of the cluster to which it belongs.

3.4 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 π 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 2.4.3, 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 and an issuing bank, we can have better initial guess about state transition probabilities as well.

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.

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.

3.5 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 O1 ;O2 ;... OR 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:

$\alpha_1 = P(O_1, O_2, O_3, \ldots\ldots O_R | \lambda)$

Let O_{R+1} be the symbol generated by a new transaction at time t +1. To form another sequence of length R, we drop O_1 an d append O_{R+1} in that sequence, generating O_2 ,O_3,... 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 α_2

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

Let $\Delta\alpha = \alpha_1 - \alpha_2$ If $\Delta\alpha > 0$ 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 $\Delta\alpha / \alpha_1 \geq$ Threshold.

The threshold value can be learned empirically, as will be discussed in Section 5. If O_{R+1} is malicious, the issuing bank does not approve the transaction, and the FDS discards the symbol. Otherwise, O_{R+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.

FDS is divided into two parts—one is the training module, and the other is detection.

4 K-MEANS ALGORITHM

K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids shoud be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as bary centers of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more. Finally, this algorithm aims at minimizing an *objective function*, in this case a squared error function.

The objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

,

where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the *n* data points from their respective cluster centres.

The algorithm is composed of the following steps:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

Although it can be proved that the procedure will always terminate, the k-means algorithm does not necessarily find the most optimal configuration, corresponding to the global

objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centres. The k-means algorithm can be run multiple times to reduce this effect.

K-means is a simple algorithm that has been adapted to many problem domains. As we are going to see, it is a good candidate for extension to work with fuzzy feature vectors.

An example Suppose that we have n sample feature vectors $x_1, x_2, ..., x_n$ all from the same class, and we know that they fall into k compact clusters, $k < n$. Let m_i be the mean of the vectors in cluster i. If the clusters are well separated, we can use a minimum-distance classifier to separate them. That is, we can say that x is in cluster i if $|| x - m_i ||$ is the minimum of all the k distances. This suggests the following procedure for finding the k means:

- Make initial guesses for the means $m_1, m_2, ..., m_k$
- Until there are no changes in any mean
  - Use the estimated means to classify the samples into clusters
  - For i from 1 to k
    - Replace m_i with the mean of all of the samples for cluster i
  - end_for
- end_until

5. BAUM WELCH ALGORITHM

In general, if we have labeled data (that is, we know the state sequence), we can obtain the parameters using Maximum Likelihood Estimation. The Baum-Welch algorithm is used to estimate the model parameters when the state path is unknown. Given sequences $O^1, O^2, . . .$, we wish to determine $\lambda = \{a_{ij} , e_i(\cdot), \pi_i\}$. We generally want to choose parameters that will maximize the likelihood of our data.

However, finding a global maximum is intractable. We would have to enumerate over all parameter sets, λ_k , and then calculate

$$\text{Score}(\lambda_k) = \sum_d P(O^d| \lambda_k) = \sum_d \sum_Q P(O^d| \lambda_k, Q)$$

for each λ_k . Instead, people settle for heuristics which are guaranteed to find at least a local

maximum. Since these are heuristics, evaluation is usually done empirically by withholding some of the training data for testing.

For a given sequence, O^d, probability of transiting from state i to j at time t is

$P(q^d_t = I, q^d_{t+1} = j|O^d, \lambda) = P(q^d_t = i, q^d_{t+1} = j, O^d)/ P(O^d) = \alpha_t(i)\, \alpha_{ij} e_j (O^d_{t+1})\, \beta_{t+1}(j) / P(O^d)$

The term $\alpha_t(i)$ is the probability that the model has emitted symbols $O_1^d . . . O_t^d$ and is in state Siat time t. This probability can be obtained using the Forward algorithm. Similarly, the Backward algorithm yields $\beta_{t+1}(j)$, the probability of emitting the rest of the sequence if we are in state j at time t+1. The remaining two terms, a_{ij} and $e_j(O^d_{t+1})$give the probability of making the transition from i to j and emitting the t + 1st character.

From this we can estimate

$$A_{ij} = \sum_d 1/ P(O^d) \sum_t \alpha(t, i)\, a_{ij}\, e_i(O^d_{t+1})\, \beta(t+1, i)$$

The probability of O_d can be estimated using current parameter values using the Forward algorithm.

Similarly,

$$E_i(\sigma) = \sum_d 1/ P(O^d) \sum_{\{t|odt=\sigma\}} \alpha(t,i)\, \beta(t,i).$$

From A_{ij} and $E_i(\sigma)$ we re-estimate the parameters.

Stated formally:

Algorithm: Baum Welch

Input:

A set of observed sequences, $O^1, O^2, . .$

Initialization:

Select arbitrary model parameters, $\lambda' = a_{ij} , e_i()$.

score $= \sum_d P(O^d| \lambda')$

Repeat

{

$\lambda = \lambda'$, $S = S'$

For each sequence, O^d,

{

/* Calculate ‘‘probable paths’’ $Q_d = q^d_1, q^d_2, ...$ */

Calculate $\alpha(t, i)$ for O^d using the Forward algorithm.

Calculate $\beta(t, i)$ for O^d using the Backward algorithm.

Calculate the contribution of O^d to A using (1).

Calculate the contribution of O^d to E using (2).

}

$a_{ij} = A_{ij} / \sum_l A_{il}$

$e_i(\partial) = E_i(\partial) / \sum_T E_i(\tau)$

score $= \sum_d P(O^d| a_{ij} , e_i())$.

}

Until (the change in score is less than some predefined threshold.)

The estimation of “probable paths” $Q_d = q^d_1, q^d_2 ...$ in the inner loop is done efficiently using dynamic programming.

6. CONCLUSION

In this paper, we have proposed an application of HMM in credit card fraud detection. The various 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] “Credit Card Fraud Detection Using Hidden Markov Model,” VOL. 5, NO. 1, JANUAR Y-MARCH 2008 by Abhinav Srivastava, Amlan Kundu, Shamik Sural, Senior Member , IEEE , and Arun K. Majumdar, Senior Member , IEEE.

[2] 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.

[3] 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.
[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. Prodromidis, and P.K. Chan, "Cost-Based Modelingfor fraud and Intrusion Detection: Results from the JAM Project," Proc. DARPA Information Survivability Conf.n and Exposition, vol. 2, pp. 130-144, 2000.
[7] E. Aleskerov, B. Freisleben, and B. Rao, "CARDWATCH: A Neural Network Based Database Mining System for Credit Card Fraud Detection," Proc. IEEE/IAFE: Computational Intelligence for Financial Eng., pp. 220-226, 1997.
[8] M.J. Kim and T.S. Kim, "A Neural Classifier with Fraud Density Map for Effective Credit Card Fraud Detection," Proc. Int'l Conf. Intelligent Data Eng. and Automated Learning, pp. 378-383, 2002.
[9] W. Fan, A.L. Prodromidis, and S.J. Stolfo, "Distributed Data Mining in Credit Card Fraud Detection," IEEE Intelligent Systems, vol. 14, no. 6, pp. 67-74, 1999.
[10] WWW.GOOGLE.COM
[11] WWW.WIKIPEDIA.ORG