Using Hidden Markov Models in Credit Card Transaction Fraud Detection

Tanya Chetcuti
University of London
tanya@chetcuti.org

Dr Alexiei Dingli
Department of Artificial Intelligence
University of Malta
alexiei.dingli@um.edu.mt

Abstract

In this paper we shall propose a credit card transaction fraud detection framework which uses Hidden Markov Models, a well established technology that has not as yet been tested in this area and through which we aim to address the limitations posed by currently used technologies. Hidden Markov Models have for many years been effectively implemented in other similar areas. The flexibility offered by these models together with the similarity in concepts between Automatic Speech Recognition and credit card fraud detection has instigated the idea of testing the usefulness of these models in our area of research.

The study performed in this project investigated the utilisation of Hidden Markov Models by means of proposing a number of different frameworks, which frameworks are supported through the use of clustering and profiling mechanisms. The profiling mechanisms are used in order to build Hidden Markov Models which are more specialised and thus are deployed on training data that is specific to a set of cardholders which have similar spending behaviours. Clustering techniques were used in order to establish the association between different classes of transactions. Two different clustering algorithms were tested in order to determine the most effective one. Also, different Hidden Markov Models were built using different criteria for test data.

The positive results achieved portray the effectiveness of these models in classifying fraudulent and legitimate transactions through a resultant percentage value which indicates the prominence of the transaction being contained in the respective model.

1.0 Introduction

The use of plastic money, particularly credit cards, is becoming increasingly popular. It has in fact become a critical component in the ever increasing world of electronic commerce.

The success of each means of payment is determined by the opportunities it provides as against the risks posed. Credit cards, although they have proved to be successful and are a popular instrument for effecting payments, still pose risks particularly in the area of e-commerce whereby online transactions are effected.

Online transactions are better known as Cardholder Not Present (CNP) transactions in the financial world. The name itself is an indication of the added risk posed by such transactions since no physical verification can be performed. Only electronic details are being transferred from the person effecting the payment to the merchant, which details normally include the card number and expiry date as well as the cardholder's personal details such as the name, surname and address. Trust is an essential element in this regard however, as with all areas that provides leeway for malicious financial gain, there are risks and makes it a target to fraudsters. As can be seen, the details used for effecting online transactions can very easily be stolen by fraudsters. Given that these details do not offer any form of verification they can be easily used in order to effect CNP fraudulent transactions. According to APACS, the UK Payments Association [11], CNP fraudulent transactions are the most common form of fraud amongst the different types which include lost and stolen cards as well as counterfeit cards. In fact, CNP fraud in the UK during 2005 amounted to £183.2 million, a substantial 41.7% of the total plastic card fraud incurred during the same year. A significant improvement was the reduction in total plastic card fraud during 2005, which amounted to £439.4 million, a decrease of 12.9% over the total plastic card fraud incurred during 2004, which figure stood at £504.8 million. However, contrary to the general scenario, CNP fraud has showed a further increase of 21.5% during 2005 as compared to 2004, highlighting the urgency for fraud prevention and detection in this area.

Our aim in this project is to formulate the structure of a credit card fraud detection system which works at transaction authorisation level, i.e. a real-time system which tries to prevent fraudulent transaction from being effected. Amongst the main challenges posed by such a system is the ability to take fast and valid decisions since time and accuracy are critical elements in such an environment.

2.0 Framework Architecture

We shall hereunder present the proposed framework of a real-time credit card fraud detection system whose engine is driven through the use of Hidden Markov Models. This engine however, needs to be supported by a complete framework and we shall thus propose the complete idea whereby we make use of a collection of technologies, merging them together in order to provide a thorough structure.

The diagram below gives a high level idea of how these concepts will be merged during the deployment of the proposed system in three main steps. Step 1 is a representation of the profiling performed on all credit card holders based on their spending behaviour. Step 2 represents the process of grouping the data of each profile on a chosen criteria and performing clustering on each group of transactions. The clusters generated for each group are ultimately used in order to represent the hidden states of each HMM and in connection with the computed vectors and matrices and the symbol set we will compute an HMM for each group of transaction patterns.

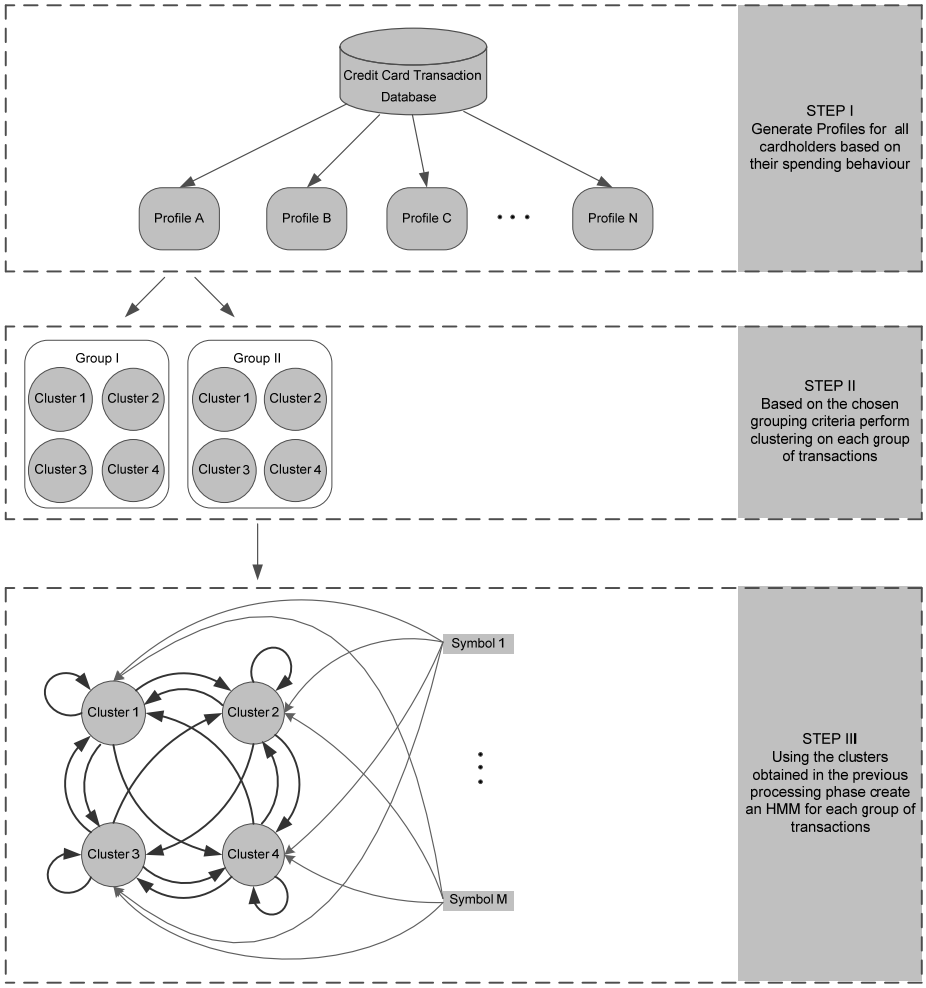

Figure 1: System Methodolgoy
Source: T Chetcuti

Once patterns of past transactions have been managed effectively, the system shall then make use of these same patterns in order to analyse and perform fraud detection on each incoming transaction authorisation. This mechanism will enable us to perform fraud detection at transaction authorisation level. This gives us the capability, in the case that the authorisation

request is suspicious, of performing authorisation scoring and consequently appropriate actions can be taken accordingly. Actions taken might include declining the authorisation request.

Transaction Data

The message format of a credit card authorisation request is based on the international standard ISO 8583 [10]. This standard defines the different attributes that a transaction authorisation may contain. At this stage we shall base our analysis on the following six attributes, which attributes are deemed as effective enough in order to monitor the nature of each transaction:

- *Transaction Type*: identifies the type of transaction being effected, i.e. whether it is a purchase, a cash withdrawal, a refund etc.;
- *Business Class*: identifies the type of business from which the transaction has originated. Codes used by this field are system defined;
- *Merchant Country*: identifies the country whereby the merchant is established;
- *Capture Method*: identifies the means by which the transaction was captured for instance ATM, POS or an electronic-commerce transaction;
- *Transaction Amount*: represents the amount in monetary value for which the transaction was effected;
- *Transaction Currency*: indicates the currency in which the transaction was effected.

Using this choice of fields in order to represent a single transaction, we will then use a sequence of these transactions per cardholder, known as the time window, for the deployment of our models. At this point we have chosen our transaction sequence to be made up of five consecutive transactions.

2.1 Profiling

Profiling is a significantly powerful technology since it has the ability of processing large amounts of data, such as historical transactions of cardholders, and deducing from them a smaller set of significant elements. For instance, if we had to compare each cardholder with the other cardholders it will be very time consuming to match two or more cardholders which have similar spending history. However, with the use of Peer Group Analysis we will be able to group together those cardholders which more or less have the same spending history and therefore we can compare the behaviour of each one to the behaviour of the other cardholders that are grouped under the same profile.

We will define the different profiles and assign the appropriate profile to each cardholder. From this point onwards cardholders' profiles will be updated with every new incoming authorisation request.

3.0 Hidden Markov Models

Hidden Markov Models shall be used in order to extract transaction patterns from past transactions and subsequently compare an incoming authorisation request, and the latest effected transactions of the same cardholder, with the sets of extracted patterns. The comparison will generate a percentage value which represents the probability of the analysed pattern being contained in that particular model.

Based on the profiling mechanism applied previously we propose that independent HMMs are devised for each profile. Such an implementation will increase the effectiveness of HMMs since spending patterns that are popular in one profile may not be popular in another profile and therefore the HMMs built will be specific for each group of cardholders.

3.1 Clustering

In order to manage better transactions that are not identical however similar in nature it is deemed important to make use of an unsupervised clustering technique. This technique will aid in capturing similar transactions into a common cluster. The derived clusters will subsequently be used to represent each of the hidden states of the HMM. In this section we shall give a detailed description of the proposed clustering mechanism.

Hidden States

Due to the fact that clustering engines are already widely deployed, it is deemed appropriate to reuse already developed engines. In fact, the clustering process within our prototype has been based on the Waikato Environment for Knowledge Analysis (WEKA) Package. It was decided that two separate prototypes shall be developed, each using a different machine learning algorithms, i.e. the partitioning algorithms SimpleKMeans and EM (Expectation-Maximization). Ólafsson [14] states that partitioning algorithms are deemed to be appropriate in particular when having large data sets.

The SimpleKMeans requires the user to enter the number of clusters required to be constructed for each data set. The best results are normally achieved through analysing the data and hence, as suggested by Mobasher [13], performing a series of tests based on different number of clusters. With each test performed the SimpleKMeans will generate the mean vectors, also known as the centroids, of each cluster together with the percentage value representing the number of instances assigned with each cluster.

On the other hand, the EM algorithm is capable of deriving the appropriate number of clusters based on the test data utilised, through the use of cross-validation. This might prove to be particularly useful in our case since depending on the training set we are using a different number of clusters might be required. The main idea behind the EM algorithm is to firstly calculate the cluster probabilities in terms of mean and standard deviation with regards to numeric attributes and value counts with regards to nominal attributes. Subsequently the algorithm will calculate the distribution parameters.

An important feature of both clustering algorithms is their ability to handle both numeric as well as nominal attributes, which attributes are assumed to be independent of each other. Secondly these algorithms have the capability of automatically normalising numeric attributes.

The fact that through the use of the SimpleKMeans and the EM algorithms each instance is associated to just one cluster allows us to use the clusters as our models' hidden states and hence compute the necessary probability values in order to have a complete model.

3.2 HMM Implementation

In this section we shall analyse what other components need to be formulated and how these will be integrated with each other, together with the hidden states obtained in the previous section, in order to be able to construct the HMMs.

Observation States

The set of observations states, also known as the symbol set, refers to those states that can be viewed by the naked eye.

For the scope of this project this set is taken to be the set of different transactions present in our test data set. Each distinct transaction, based on our choice of attributes, is considered to be a different element of the symbol set.

Initial Probability Vector

The initial probability vector will include a probability value for each of the clusters present in the model. Each value represents the possibility of having a transaction associated with that cluster at the start of the transaction pattern, i.e. at time t = 1.

These values are derived by means of keeping a count of how many of the first transactions within each pattern are associated with each cluster. Once the counts are in hand each cluster is assigned the initial probability value by means of computing the proportion between the respective count and the total count for all clusters.

State Transition Probability Vector

When processing the clustering logic it is necessary that we keep a track of the associations between each transaction and its respective cluster. This will help us, amongst other things, in computing the state transition probabilities. The latter is derived by means of keeping a count of how many transactions associated to a particular cluster are followed by transactions from each of the other clusters as well as transactions from the same cluster. Once we have these counts, i.e. the count of how many transactions associated to cluster a are followed by transactions associated with clusters a, b, c … n, we may sum up the total counts per cluster and subsequently compute the proportion of each count to the total. This same procedure is to be applied to each cluster. The derived values will constitute the state transition probability vector.

Observation Probability Matrix

Previously we discussed the process of capturing the symbol set for each HMM. It is important that whilst capturing each distinct symbol we also keep track of its popularity within the whole data set. This count is required for computing the observation probability matrix, which matrix signifies the probability value of each symbol entering each hidden state, i.e. cluster.

In addition to the symbol count, this computation requires a track of the association between the symbols and the clusters, which association is already being used for computing the state

transition probability vector. Once we possess both the count for each symbol and the associations between the symbols and the clusters we can compute the probability of each symbol entering each of the clusters. This is computed by summing up the total counts for the symbols associated with a particular cluster and then deriving the proportion of each count with respect to the total counts of the cluster in question. All remaining symbols not associated with this same cluster will obtain an observation probability of 0%.

4.0 Evaluation

The prototypes discussed in section 3 have been deployed and various tests have been performed using each prototype in different data scenarios. The intention of the tests performed was to determine whether the expected behaviour can be actually achieved and also to determine which of the prototypes generates the most efficient results.

Data Pre-processing

All of the chosen attributes accept for transaction amount were of categorical data types, which categories have been defined by the provider of the data. After analysing the categories defined for each attribute it was decided that the choices available for field business class were too specific and could therefore be mapped onto a list of more generic categories.

4.1 Test Cases

The first set of test cases shall contain the following:

- Build an HMM using the SimpleKMeans for fraudulent patterns;
- Build an HMM using the SimpleKMeans for non-fraudulent patterns;
- Build an HMM using the EM for fraudulent patterns;
- Build an HMM using the EM for non-fraudulent patterns.

On the other hand, the second set of test cases shall be further split according to the business classes. In order to determine which business cases are to be considered amongst the whole set of business classes a statistical analysis was performed. Given that the source of our data is not a specialist on fraud detection labelled fraudulent transactions were not very popular and therefore through this analysis we aim at using the largest data sets. It was found that fraudulent transactions effected at merchants having business class Transport (4000) are the most popular, at 17.84%. Next in line were the business classes Financial/Banking (6000) at 10.06% and Department Stores (5300) at 7.01%.

4.2 Test Results

Each test performed has been tested on both the Fraudulent Training Set as well as the Legitimate Training Set, whereby a percentage indicating the possibility of the test pattern being contained in the respective training set is deduced. This result is subsequently compared to the expected categorisation of the test data. Furthermore, each test performed has been performed on two different prototypes, one of which uses the SimpleKMeans clustering algorithm and the other uses the Expectation Maximization clustering algorithm.

Test Case	Clustering Algorithm	Fraud training set		Legitimate training set		Classification	
		No. of Clusters	Result	No. of Clusters	Result	Expected	Actual
1	SKM	12	0.324%	14	0.003%	F	F
	EM	3	0.529%	6	0.001%	F	F
2	SKM	12	4.688%	14	0.006%	F	F
	EM	3	2.768%	6	0.002%	F	F
3	SKM	12	0%	14	0%	F	N
	EM	3	0%	6	0%	F	N
4	SKM	12	0.003%	14	0%	F	F
	EM	3	0%	6	0%	F	N
5	SKM	12	0%	14	0%	F	N
	EM	3	0%	6	0%	F	N
6	SKM	12	5.513%	14	0%	F	F
	EM	3	0.071%	6	0%	F	F
7	SKM	12	0%	14	0%	L	N
	EM	3	0%	6	0%	L	N
8	SKM	12	0%	14	0%	L	N
	EM	3	0%	6	0%	L	N
9	SKM	12	0%	14	0%	L	N
	EM	3	0%	6	0%	L	N
10	SKM	12	0%	14	0.001%	L	L
	EM	3	0%	6	0%	L	N
11	SKM	12	0%	14	0.437%	L	L
	EM	3	0%	6	0.001%	L	L
12	SKM	12	0%	14	0.016%	L	L
	EM	3	0%	6	0.013%	L	L

SKM = SimpleKMeans
EM = Expectation Maximization
F = Fraud, L = Legitimate, N= None

Table 1: Test Results Fraud/Non-Fraud Models
Source: T Chetcuti

Table 1 above depicts the results obtained for the first set of test cases. The most prominent characteristic is that test cases 3 and 5 have not generated any percentage values whatsoever. This is in itself a result. Although we expected

the models to categorize the test cases to Fraud, a result of 0% in each model indicates that the pattern being tested is not prominent in any of the models. In actual fact the pattern under test is contained in the training data of the fraud model however it is not a popular pattern when compared to the other patterns. These results show us an important feature of HMMs, the capability of not only determining the classification of a pattern but better still assign a probability based on the likelihood of that pattern occurring.

An interesting result is that generated by test case 4 on the fraud training set whereby we can observe that the prototype utilising the SimpleKMeans algorithm has managed to detect the pattern under test whereas the prototype utilising the EM algorithm failed to do so. The fact that SimpleKMeans generates higher percentage values can in actual fact be observed in all test cases which generated a result larger than 0% but one. The fact that the model using SimpleKMeans does not always generate a higher percentage value is a positive outcome since this indicates that the SimpleKMeans algorithm is sensitive to some characteristics which we are yet to identify. In fact, upon analysing both the data used as well as the clusters constructed by each clustering algorithm we can conclude that the difference in the results is mainly due to the fact that the SimpleKMeans algorithm, which uses 12 clusters in this test case, is more accurate since it can determine more accurate transition probabilities between the clusters. On the other hand, the EM algorithm makes use of 3 clusters and therefore the transition probabilities between the three clusters are less realistic.

The latter analysis also holds for those cases whereby the SimpleKMeans generated higher percentage values such as test case 1 (for legitimate training data), test case 2 and test case 6 (for fraud training data). In these cases we can conclude that the SimpleKMeans is more specific and thus more realistic in its results.

Analysing the last 6 test cases depicted in table 1, all of which make use of test data that is a legitimate transaction pattern, we can observe similar behaviour to the previous 6 test cases. In this case we can observe that there are 3 test cases, i.e. test cases 7, 8 and 9, which generate a percentage value of 0% in both training data sets. Due to the fact that the set of data for legitimate patterns is much larger in size, since fraudulent transactions constitute a small percentage of total transactions, we only used a small random sample. However, this random sample is representative of the total sample since it can be observed that in both cases the variety of transaction patterns is vast. This variety makes it harder to generate a percentage value larger than 0% since the popularity of a transaction pattern in a vast data set is less significant.

Test case 10, when performed on the legitimate training data set, shows similar behaviour to test case 4 when generated on the fraud training data set. As can be observed in this case the prototype based on the SimpleKMeans algorithm is capable of generating a classification whereas the prototype based on the EM algorithm is not.

Test cases 11 and 12 both managed to generate a correct classification result irrespective of the prototype being used.

Based on the results illustrated in table 1 above, the following charts show an aggregate view of the effectiveness of both algorithms. As can be observed the SimpleKMeans algorithm shows a better success rate over EM.

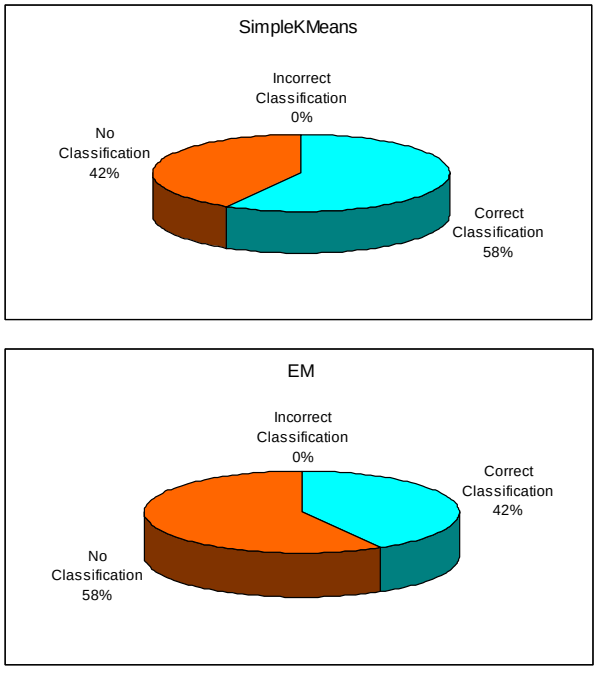

Figure 2: Classification Capability based on first set of test cases
Source: T Chetcuti

Similar to the charts displayed in figure 2, figure 3 overleaf gives a pictorial analysis of the effectiveness of building models based on the second suggested grouping criteria, i.e. based on building an independent model for each group of patterns having the last transaction pertaining to the same merchant business class.

The charts displayed in figure 3 depict a comparison between the effectiveness of the two clustering algorithms in terms of the models deployed for the second grouping suggestion. The comparison gives a clear indication that the SimpleKMeans provides more effective results when compared to the results of the models using the EM algorithm.

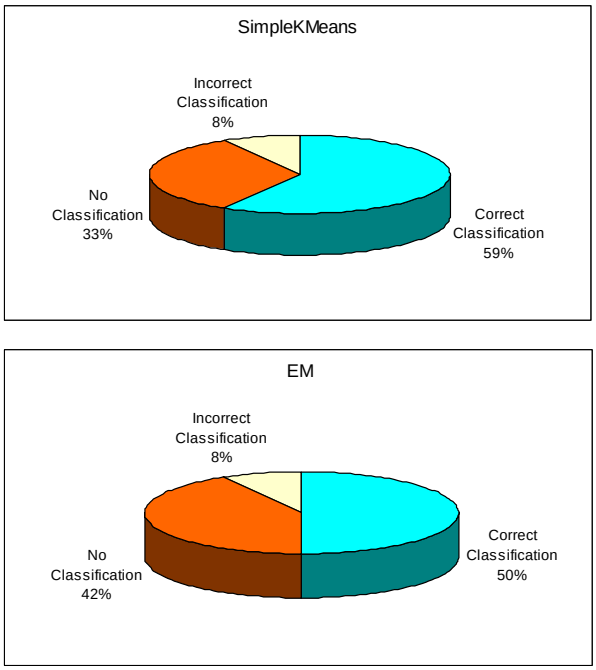

Figure 3: Classification Capability based on second set of test cases
Source: T Chetcuti

5.0 Conclusion and future work

Through the analysis performed on each of the tests described in section 4 it was observed that in general the results obtained were quite optimistic and thus promote the potential use of HMMs in the area of credit card transaction fraud detection. Most importantly this analysis highlighted the capability of HMMs in determining not only the classification of a pattern but better still assign a probability based on the likelihood of that pattern occurring.

With regards to the two grouping criteria suggested for building models one can analyse, through the use of the charts in figures 2 and 3, that although the two modelling concepts had more or less the same classification success rate when comparing the use of the SimpleKMeans algorithm there was a higher success rate for the second grouping criteria when using the EM algorithm. With regards to the incorrect classification performed through the use of the second grouping criteria we shall consider this as acceptable at this stage. Although the expected classification was different than the generated classification, the percentage values obtained indicate that although the pattern is in actual fact present in the fraud data set it is more prominent in the legitimate data set. A higher percentage value in the legitimate training data set signifies a higher risk in declining the transaction authorisation request since there is a bigger probability that the transaction is legitimate rather than fraudulent. Considering all of these issues it is therefore suggested that the second grouping criteria may be more effective, particularly when making use of the profiling mechanism.

From an efficiency perspective we can only indicate that whilst performing each of the test cases documented, models based on the EM clustering algorithm took much longer to build. In worst case scenario the EM algorithm took 10 times as much to complete than models using the same training and test data sets but based on the SimpleKMeans algorithm.

Based on all of the discussed aspects and the evaluation results obtained it is believed that the SimpleKMeans algorithm might potentially be more effective as well as efficient. A correct classification of 59% is a very positive result considering that the data we are using is only a sample of the real data and that the profiling mechanism has not been implemented. It is believed that by implementing the profiling mechanisms, which will help in building models that are more specific and specialised, the classification success rate is increased drastically. Furthermore, we should keep in mind that the 8% of misclassification is also a positive result. Despite not obtaining the expected result the actual result is a true representation of the prominence of the transaction pattern in the chosen samples of data. Such situations indicate that the risk of declining the respective transaction authorisation is higher since there is a high probability that the same transaction is a legitimate transaction. It is also believed that the profiling mechanism will reduce drastically the 33% non-classification results since specific models will help in capturing data better.

Future Work

The ideas proposed in this project can be further enhanced particularly through further in-depth analysis in order to determine the most effective clustering algorithms to be used, apart from the K-Means or the Expectation Maximization algorithms. Furthermore, one can also use the

structure and concepts proposed in this project in order to broaden the scope of the system. For instance, one can analyse how the same ideas can be applied to the merchants as well as to the cardholders. Another alternative would be analysing the application of the same structures with respect to credit card application fraud detection rather than credit card transactions.

Finally we suggest that a full implementation of the proposed framework is performed, which implementation will allow us to test the complete concepts in particular the idea of profiling users and thus having more specific Hidden Markov Models. Further thorough investigation can also be performed on implementing an effective scoring mechanism to the proposed framework. This will be capable of giving meaning to the results achieved through the use of Hidden Markov Models.

References

[1] Balfe, S., Paterson, K. G., Augmenting Internet-based Card Not Present Transactions with Trusted Computing: An Analysis, Royal Holloway, University of London, 24th October 2006, URL:http://www.ma.rhul.ac.uk/techreports/2006/RHUL-MA-2006-9.pdf [cited: December 2006]

[2] Bilmes, J., What HMMs Can Do, UWEE Technical Report, January 2002, URL:https://www.ee.washington.edu/techsite/papers/documents/UWEETR-2002-0003.pdf [cited: December 2006]

[3] Blunsom, P., Hidden Markov Models, The University of Melbourne, Department of Computer Science and Software Engineering, 19th August 2004, URL:http://www.cs.mu.oz.au/460/2004/materials/hmm-tutorial.pdf [cited: February 2007]

[4] Bolton, R. J., Hand, D. J., Unsupervised Profiling Methods for Fraud Detection, Credit Scoring and Credit Control VII, Edinburgh, UK, 5-7 September 2001, URL:http://stats.ma.ic.ac.uk/rjbolton/public_html/Edinburgh.pdf [cited: August 2006]

[5] Boyle, R.D., Hidden Markov Models, University of Leeds, URL:http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev/main.html [cited: January 2007]

[6] _____, Card Fraud Facts and Figures, APACS the UK payments association, URL:http://www.apacs.org.uk/resources_publications/card_fraud_facts_and_figures.html [cited: March 2007]

[7] Dugad, R., Desai, U. B., A Tutorial on Hidden Markov Models, Indian Institute of Technology, Bombay, May 1996, URL:http://vision.ai.uiuc.edu/dugad/ [cited: February 2007]

[8] Gamberger, D., Šmuc, T., Data Mining Tutorial: Clustering Techniques, Zagreb, Croatia: Rudjer Boskovic Institute, Laboratory for Information Systems, 2001, URL:http://dms.irb.hr/tutorial/tut_clustering_short.php [cited: March 2007]

[9] Kou, Y., Lu, C., Sirwongwattana, S., Huang, Y., Survey of Fraud Detection Techniques, Virginia Polytechnic Institute and State University USA, Tatung University Taiwan, 2002, URL:http://www.stttelkom.ac.id/staf/MAB/CS4943/ref/anomaly%20detection%20-%20outlier-fraud/fraud%20detection/00%20Survey%20of%20Fraud%20Detection%20Techniques%20(2002).pdf [cited: April 2007]

[10] Marshall, A., ISO8583 Overview, URL:http://www.amarshall.com/resix/iso8583.html [cited: April 2007]

[11] Matteucci, M., A Tutorial on Clustering Algorithms, Politecnico di Milano, URL:http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/ [cited: April 2007]

[12] McDaniel, T. L., HMMPak v1.2, Center for Cognitive Ubiquitous Computing (CUbiC), Arizona State Univeristy, 2004, URL:http://www.public.asu.edu/~tmcdani/hmm.htm [cited: February 2007]

[13] Mobasher, B., K-Means Clustering in WEKA, School of CTI, DePaul University, URL:http://maya.cs.depaul.edu/~classes/ect584/WEKA/k-means.html [cited: April 2007]

[14] Ólafsson, S., Unsupervised Learning, Department of Industrial and Manufacturing Systems Engineering, Iowa State University, URL:http://www.public.iastate.edu/~olafsson/unsupervised_learning.ppt [cited: April 2007]

[15] Salvador, S., Chan, P., Determining the Number of Clusters/Segments in Hierarchical Clustering/Segmentation Algorithms, Proc. 16th IEEE Intl. Conf. on Tools with AI, pp. 576-584, 2004, URL:http://www.cs.fit.edu/~pkc/papers/ictai04salvador.pdf [cited: September 2006]