Unsupervised Profiling for Identifying Superimposed Fraud

Uzi Murad and Gadi Pinkas

Tel Aviv University, Ramat-Aviv 69978, Israel
Amdocs (Israel) Ltd., 8 Hapnina St., Ra'anana 43000, Israel
{uzimu, gadip}@amdocs.com

Abstract. Many fraud analysis applications try to detect "probably fraudulent" usage patterns, and to discover these patterns in historical data. This paper builds on a different detection concept; there are no fixed "probably fraudulent" patterns, but any significant *deviation* from the normal behavior indicates a potential fraud. In order to detect such deviations, a comprehensive representation of "customer behavior" must be used. This paper presents such representation, and discusses issues derived from it: a distance function and a clustering algorithm for probability distributions.

1 Introduction

The telecommunications industry regularly suffers major losses due to fraud. The various types of fraud may be classified into two categories:
Subscription fraud. Fraudsters obtain an account without intention to pay the bill. In such cases, abnormal usage occurs throughout the active period of the account.
Superimposed fraud [3]. Fraudsters „take over" a legitimate account. In such cases, the abnormal usage is „superimposed" upon the normal usage of the legitimate customers. Examples of such cases include cellular cloning and calling card theft.

Data mining can be used to learn what situations constitute fraud. However, the nature of the problem makes standard pattern recognition algorithms impractical. Following are the unique characteristics of the superimposed fraud detection problem.

Context. Customers differ in calling behaviors (usage patterns and volume). A usage pattern may be normal for one customer and abnormal for another. For example, calls from New York are suspicious if the customer lives and works in Boston, but perfectly normal for New York residents. Sometimes a single customer demonstrates different types of behavior on different occasions. For example, business customers make many calls on business days, but no calls on weekends. Finally, normal behavior may change over time. It is impossible to define global fraud criteria that would be valid for all customers, all the time.

Changing fraud patterns. Following the progress of technology, fraudsters adopt new fraud techniques, which may result in new usage patterns. A set of previously observed fraud-related patterns might not suffice to detect new instances of fraud.

J.M. Zytkow and J. Rauch (Eds.): PKDD'99, LNAI 1704, pp. 251-261, 1999.

Following this discussion, a fraud detection system should be (1) sensitive to customer-unique behavior, (2) adaptable to changes in customer behavior, (3) sensitive to rare yet normal customer behavior and (4) adaptable to new fraud patterns.

2 Related Work

Several techniques concerning telecommunication and credit card fraud detection are described in literature. The majority of techniques monitor customer behavior regarding specific usage patterns. [10] describes a rule based system, which accumulates number or duration of calls that match specific patterns (e.g., international calls) in one day and calculates the average and standard deviation of the daily values. Then it compares new values against a user-defined threshold in terms of standard deviations from the average. [9] describes a neural network based system, which uses similar parameters, but learns from known cases what situations (combinations of current value, average value and standard deviation) are fraudulent. In this approach, the threshold is determined from known cases, rather than being user-defined.

[3] uses supervised learning to discover from historical data what usage patterns are „probably fraudulent", but still deals with specific patterns. In general, any supervised learning algorithm has to receive a significant number of cases of a pattern in order to learn it. It might be difficult to obtain enough known cases of each pattern. In addition, new patterns of fraud must be discovered by other means (such as customer complaints), before being presented to the learning algorithm. A long time may pass from the first occurrence of the pattern until it is used for detection. Finally, accurately classified data may be hard to get.

A different approach, the one this paper follows, is to create a general model of behavior, without using predefined patterns. [7] deals with credit card fraud, and profiles customers by their typical transactions. A new transaction is examined against the profile, to see whether similar transactions have appeared in the past. There is no element of usage volume, or frequency of transactions. [1] represents short-term behavior by the probability distribution of calls made by a customer in one day. This „Current User Profile" (CUP) is examined against the „User Profile History" (UPH), which is the average of the CUPs generated by that user. Call volume is not taken into consideration, and, since the UPH is an average, it does not reflect rare yet normal patterns.

3 Three Level Profiling

The technique for fraud detection suggested in this paper does not search for specific fraud patterns; instead, it points out any significant deviation from the customer's normal behavior as a potential fraud. In order to detect such deviations, a comprehensive representation of behavior, which can capture a variety of behavior patterns, must be used, and the definition of „deviation" should reflect the actual degree of dissimilarity between „behavior instances". To meet these goals, three profile levels are used:

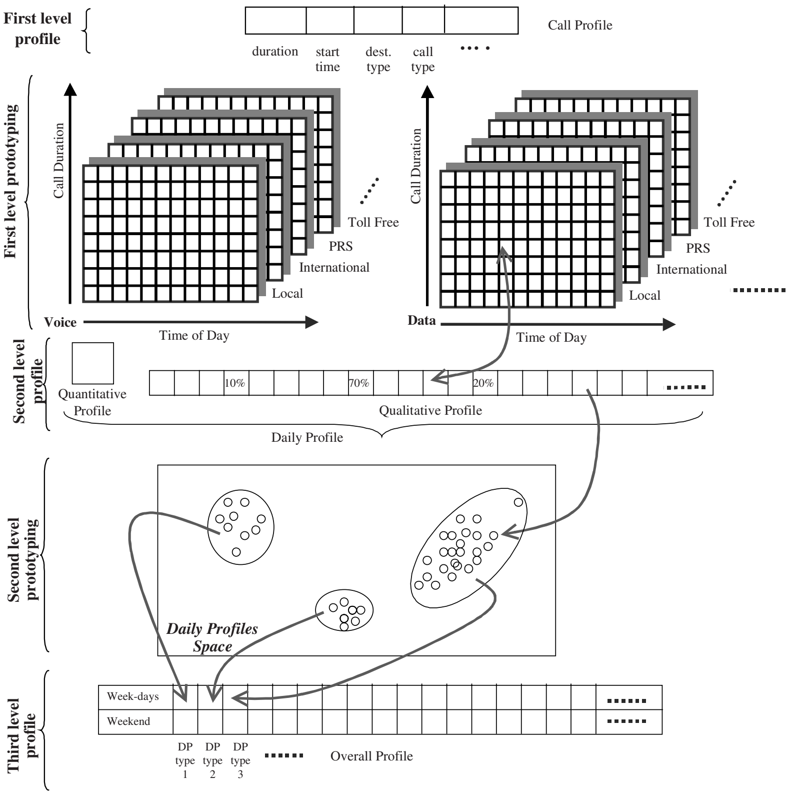


Fig. 1. Three Level Profiling

Call Profile – represents a single call and includes all fields of the Call Detail Record (CDR) that are relevant to behavior.

Daily profile – represents the short-term behavior of a customer. The daily profile consists of two parts: The qualitative profile describes the kind of calls (when, from where, to where) the customer made during the day, and the quantitative profile describes the usage volume. Each attribute in the call profile is seen as a random variable, and the qualitative profile is the empirical multi-dimensional probability distribution of calls on a given day. Since the space of call profiles is huge, it is partitioned into a finite number of subspaces, each of which is represented by a „prototypical call“. The qualitative profile is a vector, which contains an entry for each „call prototype“ with the percentage of calls (on a given day) of that prototype.

Overall Profile – represents the long-term „normal“ behavior of a customer, and reflects the daily behaviors normally exhibited by the customer. Since the space of Daily Profiles is infinite, a clustering algorithm is applied to the daily profiles, which extracts „prototypical days“. The overall profile is a vector with entries for all „daily prototypes“, containing information about the „normal“ usage level for that customer in days of that type. Since customers may behave differently on different „types of days“, a separate vector is kept for each type. In the current framework there are two types: weekdays and weekends. Fig. 1 illustrates the concept of Three Level Profiling.

4 The Call Profile

The call profile $c=(c_1,c_2,\ldots,c_d)$ includes all those features of the Call Detail Record (CDR) that are relevant to behavior. For the data set used for the first implementation (wireline data), the following attributes were used: call start time, call duration, destination type (local, international, premium rate service or toll-free) and call type (voice or data). Other attributes should be considered for different data sets. In a cellular context, for example, the originating location would probably be included.

Each attribute c_i corresponds to a random variable X_i, which gets values from domain D_i. The domain of call profiles $D = D_1$ x D_2 x …x D_d is huge; call duration may be any number of seconds, and start time may be any time (in seconds) in the day. The huge domain of CDR profiles should be represented by a relatively small number of call prototypes. For our purpose, a good set of prototypes will be such that any new call has a prototype similar enough to it, and different prototypes are dissimilar enough. We extract call prototypes as follows. For continuous or ordinal attributes (call duration and start time) the domain D_i of attribute i is split into n_i ranges, resulting in a discrete domain $D_i^*=\{ x_1^i, x_2^i, \ldots, x_{n_i}^i \}$. For discrete non-ordinal attributes $D_i^* = D_i$.

In our framework, we partitioned the start time into 12 two-hour time windows. Call duration was partitioned to 12 five-minute ranges, which cover calls up to an hour long. There is, of course, inaccuracy in calls longer than one hour. However, since 99.9% of the calls are shorter than one hour, and long duration calls can be monitored easily by other means, this compromise is reasonable. The resulting discrete domain contains $l = n_1 \cdot n_2 \cdot \ldots \cdot n_d$ values.

5 The Daily Profile and the Curse of Dimensionality

The daily profile *DP* consists of a quantitative profile *DP.n* (the number of calls made in that day) and a qualitative profiles $DP.q = (q_1, q_2, \ldots, q_l)$ where $0 \le q_i \le 1$, $\Sigma q_i = 1$ and l is the number of call prototypes. Each entry q_i corresponds to a call prototype, and contains the percentage of calls of that type in the given day.

The huge dimension of the vector ($12 \cdot 12 \cdot 4 \cdot 2 = 1{,}152$ in our case) seems to be problematic in terms of both storage space and computation time. However, only few prototypes actually appear in a single daily profile. The average number of non-zero call prototypes in a daily profile (calculated using over 400,000 Daily Profiles) is 2.8, and 99% of the daily profiles have 13 or less non-zero call prototypes. If time and space complexity depend on the number of non-zero prototypes only, then the dimensionality problem ceases to exist. To achieve this, daily profiles are saved in variable-length arrays. Computations with daily profiles are discussed next.

6 Distances Between Qualitative Profiles

A distance function between daily profiles is required for clustering daily profiles and for detection of deviations. It is extremely important that the distance function reflect the actual level of similarity between two daily profiles.

We examined several known distance functions. The Euclidean distance is not adequate for the problem. Consider, for example, the three following call prototypes:
Call prototype 1: 5-minute local voice call at 2:00PM
Call prototype 2: 10-minute local voice call at 2:00PM
Call prototype 3: 60-minute international data call at 2:00AM
and three daily profiles, each of which contains only calls of prototype 1, 2 and 3, respectively (i.e., each daily profile has value 1 in one entry and 0 in all others). While it is obvious that the first daily profile is similar to the second, and absolutely different from the third, the Euclidean distances are identical and equal to $\sqrt{2}$, which is the maximal distance. The reason for this distortion is that Euclidean distance does not take into account the similarity between attributes. Each attribute is treated as totally different in meaning from all other attributes. In our case, however, the attributes represent points in the *d*-dimensional space of call profiles and, as such, some attributes are „closer" (in meaning) to each other than others. The Helinger distance (suggested by [1]) and the Mahalonobis distance [5] suffer from the same distortion.

Since the qualitative profiles represent multidimensional probability distributions, distance functions between probability distributions should be considered. Distance functions discussed in [6] (for example, Patrick-Fisher, Matusita and Divergence) share the same problem.

An obvious solution is to compare the probability for *neighborhoods* of values. However, the performance depends on the neighborhood size. Small neighborhoods will not be sensitive enough in cases where the values are distant, and large neighborhoods will loose information. Moreover, large neighborhoods may require many more prototypes to be handled than the non-zero ones.

Cumulative Distribution Based Distance. We propose the *CD-distance*, which is based on cumulative distribution. The cumulative distribution enables to capture the „closeness" of values, in addition to their probabilities. The distance function sums the squared differences of the *cumulative* distribution functions (instead of the *density* functions).

Formally, in the one-dimensional case we define the distance as follows. Let $f_1(x)$ and $f_2(x)$ be two continuous probability distributions of a random variable X. The distance between $f_1(x)$ and $f_2(x)$, denoted by $d(f_1, f_2)$ is

$$d(f_1, f_2) = \sqrt{\frac{1}{x_{\max} - x_{\min}} \int_{x_{\min}}^{x_{\max}} \left(F_1(x) - F_2(x)\right)^2 dx} \quad . \tag{1}$$

where $F(x)$ is the cumulative distribution ($F(x) = P(X \le x)$), and $x_{\max}$ and $x_{\min}$ are the maximum and minimum values, respectively, of the random variable X. We define $x_{\max}$

and $x_{\min}$ such that the probability of values outside ($x_{\min}$, $x_{\max}$) is redundant. $x_{\max}$ and $x_{\min}$ do not depend on the two density functions being compared, but solely on the random variable X. In order to obtain distance values on a normalized scale, the sum is divided by ($x_{\max}$ - $x_{\min}$), so we get $0 \le d(f_1, f_2) \le 1$.

In the discrete, ordinal case, the domain of the random variable X is an ascending list of values ($\mathcal{D} = \{x_1, x_2, \ldots, x_n\}$, $x_i > x_j \Leftrightarrow i > j$), and the distance between two probability distributions is

$$d(f_1, f_2) = \sqrt{\frac{1}{x_n - x_1} \sum_{j=1}^{n-1} \left(F_1(x_j) - F_2(x_j)\right)^2 \delta_j} \qquad (2)$$

where $\delta_j = x_{j+1} - x_j$.

In this case, the cumulative distribution is a step-function, and (2) is a straightforward simplification of (1) for the subset of discrete probability distributions. x_1 and x_n, the smallest and largest discrete values, respectively, serve as $x_{\min}$ and $x_{\max}$. δ_j are the differences between consecutive discrete values. Note that δ_j are not necessarily equal, therefore, to improve computation efficiency, we can consider only the non-zero entries (entries k such that $P_i(X = x_k) \neq 0$, i=1,2).

Binary random variables are treated as discrete ordinal random variables, with the domain {0,1}. In the discrete, non-ordinal case (e.g., call type), the domain consists of non-numeric values. We assume that the similarity between each pair of values is equal. In this case, we replace the call profile attribute i with n_i binary attributes, where n_i is the cardinality of domain D_i.

The generalized distance function in the multidimensional case is:

$$d(f_1, f_2) = \sqrt{\sum_{i=1}^{d} w_i \left(\frac{1}{x_{n_i}^i - x_1^i} \sum_{j=1}^{n_i-1} \left(F_1(x_j^i) - F_2(x_j^i)\right)^2 \delta_j^i \right)} \qquad (3)$$

where x_j^i is the jth prototype of call profile attribute i, $F(x_j^i) = P(X^i \le x_j^i)$, and w_{i} is a weight for call profile attribute i. $\Sigma w_i = 1$.

It can be shown [11] that the *cd-distance* function maintains, for any probability distribution functions f_1, f_2 and f_3:

(1) $0 \le d(f_1, f_2) \le 1$

(2) $d(f_1, f_1) = 0$

(3) $d(f_1, f_2) = d(f_2, f_1)$ (symmetry)

(4) $d(f_1, f_2) + d(f_2, f_3) \ge d(f_1, f_3)$ (triangle inequality)

7 Extracting Daily Prototypes

The space of daily qualitative profiles is infinite, and we need to represent it by K „prototypical“ qualitative profiles, which represent prototypical behaviors. This discretization is important not only for performance, but also to facilitate the investigation of alerts by the human analyst.

We use a clustering algorithm to extract such prototypes. Since the distance function is not Euclidean, a proximity-matrix-based algorithm seems to be essential. However, such algorithms are restricted to small sample sets, due to their space and time complexity. In our case, however, extracting a sufficiently small sample may result in losing unique prototypes.

The partitional algorithm *K-means* is based on Euclidean distance and does not require a proximity matrix. The algorithm seeks to minimize the sum of squared distances between samples and their associated cluster:

$$\sum_{i=1}^{N} d^2(p_i, c_{p_i}) \; . \qquad \textbf{(4)}$$

Recalculating the new cluster center as the Euclidean centroid of samples assigned to it locally minimizes this criterion. It can be shown [11] that this method of recalculating cluster centers also minimizes the criterion function with the *CD-distance*, therefore the K-means algorithm can be used, with the *CD-distance* replacing the Euclidean distance. An „adaptive“ version K-means [8] is used in order to determine dynamically the number of clusters.

8 The Overall Profile

The overall profile *OP* is an array containing an entry for each daily prototype. Each entry OP_i contains the number of days of prototype *i* observed for the account ($OP_i.n$), the sum (OP_i $.sn$) and sum of squares ($OP_i.ssn$) of number of calls in these days. These components are later used to calculate the average M_i and standard deviation σ_i of the number of calls per day of a prototype, as described in [10]. In order to adapt to changes in customer behavior, the components $OP_i.x$ with $x = n, sn, ssn$ may be updated using a decay function, like the one used in [10]. In our current prototype, two such arrays are used; one for business days and another for weekends.

9 Deviation Detection

Matching a daily profile to the overall profile includes qualitative and quantitative checks. The qualitative profile matches the overall profile if it is closer than threshold $T_{\text{qualitative}}$ to the nearest non-zero daily prototype of that customer. The quantitative profile matches the overall profile if $(DP.n - M_i)/\sigma_i \leq T_{\text{quantitative}}$, where *i* is the prototype closest to *DP*) and $T_{\text{quantitative}}$ is a threshold in terms of standard deviations.

In order to reduce false alerts on daily profiles in which the deviating activity is low, we wish to ignore such profiles, which are of no interest to investigate. We assign a value to each daily profile based on call durations and destinations, which represents the „interestingness level“ of that day (not necessarily the cost of calls). Daily profiles with a value smaller than T_{value} are not checked, but still update the overall profiles. Finally, quantitative deviations on days with less than T_{ncalls} calls do not issue alerts.

A detection process constantly reads CDRs and updates the daily profiles. Once a day it performs the deviation detection and updates the overall profiles.

10 Evaluation

To test the technique, we used the data set of wireline CDRs. The data set covered three months' usage of about 7,000 accounts. We used the first two months' calls only to learn overall profiles. The third month's data was checked for behavioral changes and used to update the profiles. We considered only customers with more than 20 active days in the first two months and without fraudulent usage during this period. A total of 6,334 accounts maintained these conditions, out of which 82 accounts (1.3%) included superimposed fraudulent usage in the third month. We ran the system with 240 combinations of the four thresholds. Alerts on one of the first two fraudulent days are considered as „hits".

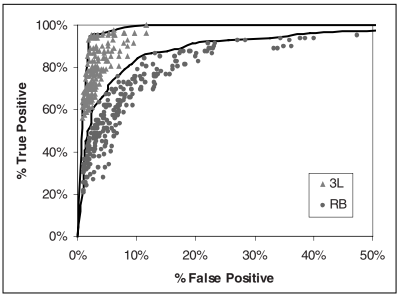
For comparison we took the widely used rule-based method (for example, [10]). This method is also based on unsupervised learning, and attempts to detect significant changes (increase) in usage. This method accumulates usage (number or duration of calls of certain types) and calculates the average and standard deviation of the daily values of each accumulator. The average and standard deviation are updated with each newly introduced accumulator value. An alert is issued whenever the value of a certain accumulator exceeds a threshold T_{stdevs}, defined in terms of a standard deviation from the average. We used accumulators of voice, data, international, PRS, toll-free and nightly calls. We accumulated both the number and duration of calls of each type. Here also, only the first two months' calls were used for learning. In order to prevent alerts on days with low activity, we also used T_{value} (where the daily value was calculated as in our method), T_{ncalls} and T_{duration} thresholds. We ran the algorithm with 240 different combinations of values for these four thresholds.

Evaluation Technique. For our purpose, we cannot compare classifiers using accuracy alone, since the class distribution is not constant and the costs of false negative errors and false positive errors are not equal. In addition, the class distribution is very skewed [4]; in our case, a „do nothing" strategy will give 98.7% accuracy.

In both methods, the thresholds control the number of alerts. The lower the thresholds, the more true positives and more false positives are produced. Therefore, to evaluate the system's performance, we use two measures:

- *True positive (hit) rate*. the ratio of detected fraud cases of all fraud cases
- *False positive (false alarm) rate*. the ratio of nofraud cases classified as fraud of all nofraud cases.

Then we compare the hit rates achieved by each method, given a fixed false alarm rate.



Allowed False Positive Rate	Possible True Positive Rate	
	3L	RB
1%	65.85%	25.61%
2%	93.90%	50.00%
3%	95.12%	59.76%
4%	95.12%	63.41%
5%	96.34%	68.29%
10%	97.56%	81.71%
15%	100.00%	86.59%

Fig. 2. Performance comparison on real data. Hit rate vs. false alarm rate of all 240 runs of each method, with the non-decreasing hulls (graph), and a summary of best results (table).

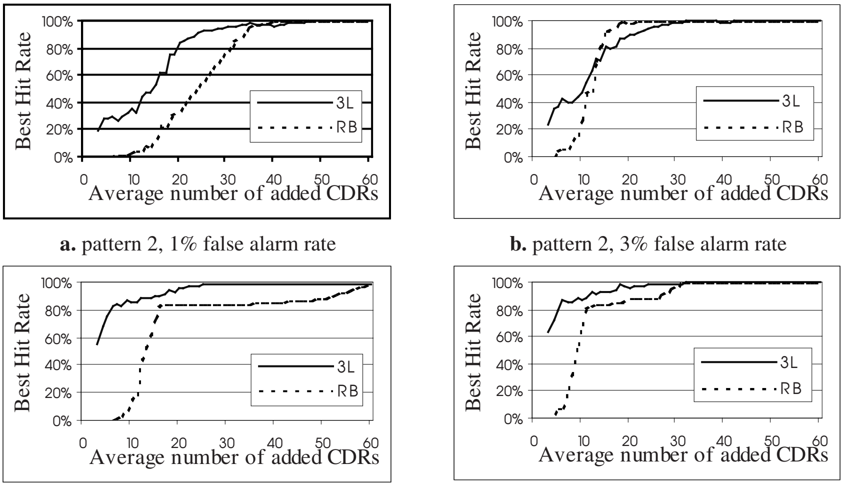
The graph in Fig. 2 depicts the results of the 240 runs of each algorithm, with the corresponding non-decreasing hulls. The non-decreasing hulls reflect the best results of each method. The best results are also summarized in the table in Fig. 2. It can be seen that the 3L method outperforms the naive RB approach, as even the worst results of 3L dominate the best results of RB. 3L reaches high detection rate given low false alarm rates, where RB performs poorly.

Semi-Synthetic Data. In order to evaluate the 3L method on a larger set of fraud cases, and to analyze its sensitivity to various fraud patterns and to different volumes of fraud, we use semi-synthetic data. For this purpose, we added a synthetically generated fraudulent usage to the original usage of 10% (625) of the fraud-free accounts. To each of the selected accounts we added „fraudulent“ usage covering three consecutive days and matching one of six fraud patterns. The patterns differ in quantity of calls and in structure and are distributed equally over the 625 cases.

Table 1 shows the total hit rate, as well as the rates of the detected cases of 4 selected patterns, given various false alarm rates. The hit rates correspond to the configuration which generated the best *total* hit rate (rather than best hit rate for each pattern separately), therefore the hit rates of the patterns are not always increasing.

Table 1. Performance comparison on semi- synthetic data. The average daily number of added cdrs is shown in parentheses in the titles, following by the pattern's description

Allowed	**Possible True Positive Rate**									
False	Pattern 1 (7)		Pattern 2 (100)		Pattern 3 (4)		Pattern 4 (25)		Total	
Positive	Long PRS		Call reselling		Intl. In evenings		No pattern			
Rate	3L	RB	3L	RB	3L	RB	3L	RB	3L	RB
1%	90.7%	-	100.0%	-	68.7%	-	70.5%	-	78.6%	
1%	91.6%	23.4%	100.0%	100.0%	71.7%	1.0%	72.4%	56.2%	81.9%	50.1%
2%	91.6%	100.0%	100.0%	100.0%	71.7%	1.0%	81.0%	99.0%	85.1%	77.6%
3%	91.6%	100.0%	100.0%	100.0%	72.7%	3.0%	89.5%	99.0%	88.0%	82.2%
4%	91.6%	100.0%	100.0%	100.0%	72.7%	3.0%	89.5%	100.0%	88.0%	83.5%
5%	91.6%	100.0%	100.0%	100.0%	73.7%	25.3%	90.5%	99.0%	90.9%	86.4%
10%	92.5%	100.0%	100.0%	100.0%	73.7%	41.4%	95.2%	100.0%	93.4%	89.8%



a. pattern 2, 1% false alarm rate

b. pattern 2, 3% false alarm rate

c. pattern 3, 1% false alarm rate

d. pattern 3, 3% false alarm rate

Fig. 3. Sensitivity to pattern size. The points on each line represent the highest hit rate achieved under the given false alarm rate

On the „intensive“ patterns (especially pattern 2) both systems perform well, with the rule-based method performing slightly better. On the other hand, the 3L method is more sensitive to „less obvious“ patterns (such as pattern 4). The 3L method performs better under a lower false positive rate, but gradually loses its superiority when given higher false alarm rates.

We have also examined the sensitivity of the technique to the „size“ of fraud (i.e., the number of fraudulent calls). To do this, we used fraud patterns 2 and 3. Pattern 2 („call reselling“) is „general“: all types of voice calls, throughout the day, with various durations. Pattern 3 is „specific“: international data calls in the evenings. For each pattern, we added fraudulent calls to the selected accounts, increasing gradually the number of added calls per day from 3 to 60. For each number of added calls, we ran both methods, each with 240 combinations of thresholds. We then checked how the hit rate increases with the number of added calls, given a fixed rate of false alarm rate. Fig. 3 shows the results. On the general pattern (Fig. 3(a) and 3(b)), the 3L method is more sensitive to small cases of fraud, and gradually looses its superiority when the number of added calls increases. The difference is more significant under lower false alarm rates. On the specific pattern (Fig. 3(c) and 3(d)) 3L reaches relatively high hit rate for small numbers of added calls, significantly better than that of RB.

11 Conclusion and Future Work

Three Level Profiling provides a comprehensive representation of customer behavior. Using this profiling method, rather than profiles based on predefined usage patterns, the system can cope with the dynamic nature of telecommunication fraud.

Initial experiments show a superiority of this method over the rule-based method. In the future, we intend to apply the profiling technique to identify changes in calling behavior for other purposes, such as identifying marketing opportunities and offering new incentives or price plans to customers following a behavioral change.

Acknowledgements

This paper is based on an M.Sc. thesis written by Uzi Murad under the supervision of Professor Victor Brailovsky at Tel Aviv University. We thank Victor Brailovsky for insightful discussions and Saharon Rosset for helpful comments and suggestions.

References

1. Burge, P., Shawe-Taylor, J.: Detecting Cellular Fraud Using Adaptive Prototypes. In: Proceedings of AAAI-97 Workshop on AI Approaches to Fraud Detection and Risk Management. Providence RI (1997) 9-13
2. Duda, R. O., Hart, P.E.: Pattern Classification and Scene Analysis. John Wiley & Sons, Inc., New York NY (1973)
3. Fawcett, T., Provost, F.: Adaptive Fraud Detection. In: Fayyad, U., Mannila, H., Piatetsky-Shapiro, G. (Eds.): Data Mining and Knowledge Discovery, vol 1. Kulwer Academic Publishers, Boston CA (1997) 291-316
4. Fawcett, T., Provost, F.: Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions. In: Agrawal, R., Stolorz, P., Piatetsky-Shapiro, G. (Eds.): Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, AAAI Press, Menlo Park CA (1997) 43-48
5. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs NJ (1988)
6. Kittler, J.: Feature Selection and Extraction. In: Young, T.Y., Fu, K. (Eds.): Handbook of Pattern Recognition and Image Processing. Academic Press Inc., Orlando FA (1986) 59-83
7. Kokkinaki, A.I.: On Atypical Database Transactions: Identification of Probable Fraud using Machine Learning for User Profiling. In: Proceedings of IEEE Knowledge and Data Engineering Exchange Workshop (1997) 107-113
8. McQueen, J.B.: Some Methods for Classification and Analysis of Multivariante Observations. In: Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability (1967). 281-297
9. Moreau, Y., Vandewalle, J: Detection of Mobile Phone Fraud using Supervised Neural Networks: A First Prototype. Available via ftp://ftp.esat.kuleuven.ac.be/pub/SISTA/moreau/reports/icann97_TR97-44.ps (1997)
10. Moreau, Y., Preneel, B., Burge, P., Shawe-Taylor, J., Stoermann C., Cook, C.: Novel Techniques for Fraud Detection in Mobile Telecommunication Networks. In: ACTS Mobile Summit, Grenada Spain (1997)
11. Murad, U.: Three Level Profiling for Telecommunication Fraud Detection. M.Sc. thesis, Tel Aviv University, Israel (1999)