FRAUD DETECTION IN MOBILE COMMUNICATIONS NETWORKS USING USER PROFILING AND CLASSIFICATION TECHNIQUES

F. N. Ogwueleka
Department of Mathematics,
Statistics and Computer Science, University of Abuja

ABSTRACT

Fraud detection is an important application, since network operators lose a relevant portion of their revenue to fraud. The intentions of mobile phone users cannot be well observed except through the call data. The call data is used in describing behavioural patterns of users. Neural networks and probabilistic models are employed in learning these usage patterns from call data by detecting changes in established usage patterns or to recognize typical usage patterns of fraud. The methods are shown to be effective in detecting fraudulent behaviour by empirically testing the methods with data from real mobile communications networks.

Keywords: *Call data, fraud detection, neural networks, probabilistic models, user profiling*

INTRODUCTION

Fraud detection methods are continuously being developed to checkmate criminals who also adopt new strategies regularly. The development of new fraud detection methods is made more difficult due to the severe limitation imposed by restricted information flow about the outcome of fraud detection efforts. Fraud detection has been implemented by a number of methods such as data mining, statistics, and artificial intelligence. Fraud is discovered from anomalies in data and patterns (Ogwueleka and Inyiama, 2009).

Mobile communication fraud can be defined as any transmission of voice or data across a telecommunications network where the intent of the sender is to avoid or reduce legitimate call charges (Johnson, 1996). With the aid of the fraud detection models, fraudulent activity in a mobile communications network may be revealed. This is beneficial to the network operator, who may lose several percent of revenue to fraud, since the service charges from the fraudulent activity remain uncollected.

In the area of mobile communication fraud, some researches done include the report on the use of a knowledge-based approach to analyze call records delivered from cellular switches in real time by Davis and Goyal (1993). Barson et al. (1996) reported their first experiments detecting fraud in a database of simulated calls. Burge and Shawe-Taylor (1997) focus on unsupervised learning techniques in computing user profiles over sequences of call records. Fawcett and Provost (1997) presented rule-based methods for fraud detection by using adaptive rule sets to uncover indicators of fraudulent behaviour from a database of cellular calls.

Intrusion detection approach can be divided into two classes of anomaly and misuse detection. Anomaly detection is the same as differential analysis and it approaches the problem by attempting to find deviations from the established patterns of usage. Misuse detection is similar to absolute analysis and compares the usage patterns to known techniques of compromising computer security (Kumar 1995).

User profiling is the process of modeling characteristic aspects of user behaviour. In user classification, users are assigned to distinctive groups. Probabilistic networks permits an efficient description of multivariate probability densities and allow quantifying uncertainty in the conclusions made about the problem, which makes the framework of probabilistic networks appealing for real-world problems. Neural networks (NN) are analytic techniques modeled after the hypothesized processes of learning in the cognitive system and the neurological functions of the brain, capable of predicting new observations on specific variables from other observations on the same or other variables after executing a process of so-called learning from existing data. Self-organizing map is the type of NN used in this study.

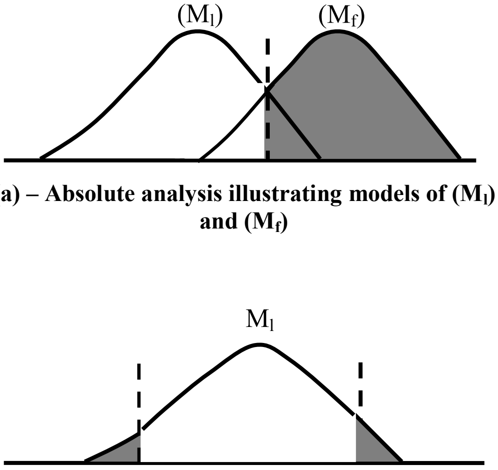
The intentions of mobile communication subscribers are reflected in the observed call data, which is used in describing the behavioural patterns of users. The goal is to use the call data to learn models of calling behaviour so that these models make inferences about users' intentions. Neural networks and probabilistic models are employed in this study for learning these usage patterns from call data by either detecting abrupt changes in established usage patterns or by recognizing typical usage patterns of fraud. Learning these usage patterns means adaptation of the parameterized models so that the inherent problem structure will be coded in the model. Neural networks and probabilistic methods prove to be effective in detecting fraudulent behaviour by testing the methods with data from real mobile communications networks. There is no specific sequence of calls that would be fraudulent with absolute certainty as same series of calls could be fraudulent or legitimate. This means that uncertainty in modeling the problem is needed and this was embodied in the probabilistic model framework.

Detection methodologies designed for one specific scenario are likely to miss plenty of the others. For example, velocity trap and overlapping calls detection methodologies are solely aimed at detecting cloned instances of mobile phones and do not catch any of the subscription fraud cases. The nature of fraud can change from cloning fraud to subscription fraud, which makes specialized detection methodologies inadequate, so the focus of detection methodology in this paper was based on the calling activity of a stream of transactions, which was divided into two categories of analysis – absolute and differential. These two analysis methods proved to be the best when implemented using probabilistic models and neural networks.

METHODOLOGY

Fraud in telecommunications networks can be characterized by fraud conditions, which essentially describe how the fraudster gained the illegitimate access to the network. This study concentrated on the detection methodologies based on the calling activity. This is divided into two categories of absolute and differential analysis as shown in Figure 1 using the probabilistic aspect. In absolute analysis, detection was based on the calling activity models of fraudulent and legitimate behaviour. Differential analysis approached the problem of fraud detection by detecting sudden changes in behaviour. When current user behaviour differs from the established model of user behaviour, alarm is raised. Alarm deviations from the established patterns of usage are observed while using differential analysis.

From Figure 1a, it can be observed that in absolute analysis, models of both legitimate behaviour (M_l) and fraudulent behaviour (M_f) were developed. In differential analysis (Fig.1b), one model was built assuming legitimate behaviour



(a) – Absolute analysis illustrating models of (M_l) and (M_f)

(b) – Differential analysis showing models of (M_l)

Fig. 1(a and b): Absolute and differential analysis

(M_l) so differences from the proven behaviour are classified as fraudulent. The dashed lines indicate some biased decision margins and the shaded area represents the classified fraudulent regions.

In this study, these analysis methods are used extensively in modeling of call behaviour. It also dealt on the dynamical modeling of behavioural patterns for fraud detection. The models are implemented using neural networks and probabilistic models. The methods used solved the learning problem with a mixture setting of data enabling one to have access to data from legitimate call accounts and accounts that contain fraudulent data. Fraud detection in this paper was based on the calling activity of mobile phone subscribers. The calling activity is recorded for the purpose of billing in call records, which store attributes of calls, like the identity of the subscriber, time of the call, and duration of the call.

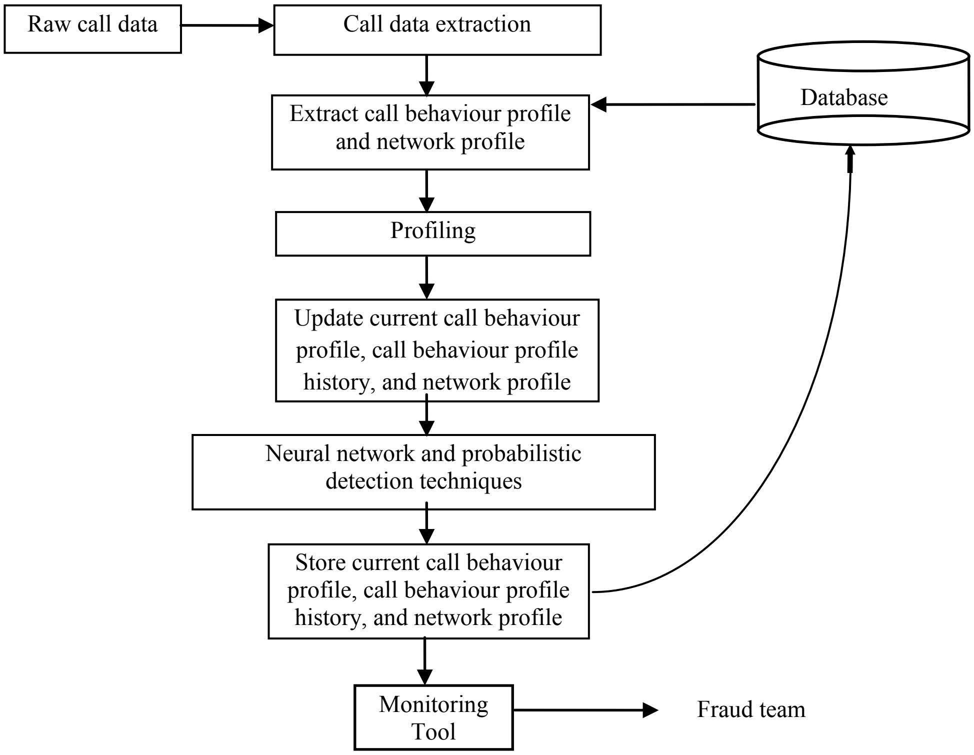

Raw call data
Call data extraction
Extract call behaviour profile and network profile
Database
Profiling
Update current call behaviour profile, call behaviour profile history, and network profile
Neural network and probabilistic detection techniques
Store current call behaviour profile, call behaviour profile history, and network profile
Monitoring Tool
Fraud team

Fig. 2: System architecture

The systems architecture is shown in the Figure 2. The features of a call behaviour profile depend on the nature of the application. The technique was to maintain histories of call usage information, relating to the entity, over differing time periods.

The short term past call behaviour is referred to as the Current Behaviour Profile (CBP) and the long term past call behaviour as the Behaviour Profile History (BPH). The detection engine task was to determine if a significant change in call behaviour has occurred. This is known as performing a differential analysis. When the CBP exceeds predetermined thresholds for acceptable network usage over the lifetime of the CBP, alarms can also be raised. This is known as performing an absolute analysis.

In order to work towards the goal of generalized detection tool for mobile communication fraud detection, the intermediate phase was investigated as shown in Figure 3. In order to develop models of legitimate and fraudulent behaviour and to be able to assess the diagnostic accuracy of the models, call data exhibiting both kinds of behaviour was required. Collecting legitimate call data is relatively easy as this mode dominates the population, but collecting fraudulent call data is more difficult and rare. The procedures in data collecting differ both in the way they are conducted and in the way the data is grouped in the legitimate and fraudulent modes.

The two ways of collecting fraud data for designing and implementation of mobile communication fraud detection system used in this study are block crediting and velocity trap. In block crediting, after each billing period, telephone bills are calculated from the subscriber specific call data using appropriate tariffs for each service. A bill is sent to the customer, who either approves or disapproves the billed amount. If a fraudster has exploited an account during the billing period, the customer is likely to disapprove the high cost of calling. As this process was likely to contain errors, each call was classified as legitimate or fraudulent class, which can be considered a relatively accurate labeling of data. Under velocity trap, it was considered that it would be beneficial if a fraud detection system could be designed using data from legitimate and fraudulent accounts without generally involving human labour, so an approach was used. This approach was to filter fraudulent call data from a large database by creating a simple basic fraud model and testing whether call data is fraudulent or legitimate. This works under the assumption of cloning fraud and using a velocity trap as the basic fraud model. Velocity trap raises alarms if calls are made from locations far apart in terms of closeness and this sets a limit on the velocity a mobile phone subscriber may travel.

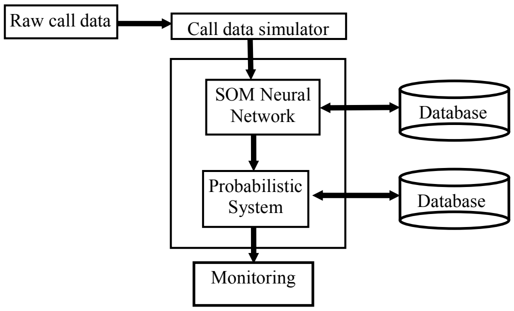
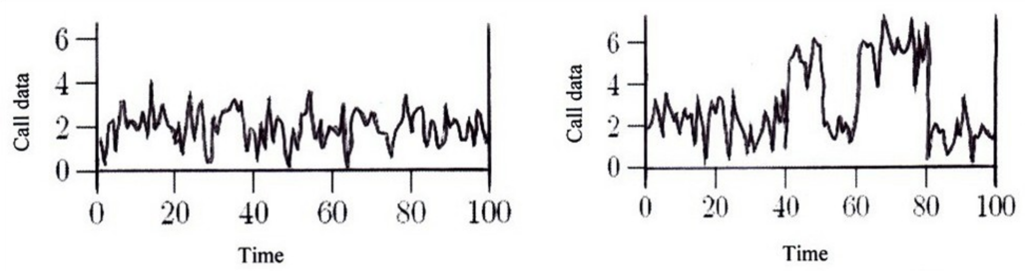


Fig. 3: Mobile communication detection tools

Fraud data used in this study was filtered from a database of call data using a velocity trap detection mechanism to enable the data not to contain information on which calls were fraudulent or which periods contained fraudulent activity. Data labeled as fraudulent is a sample from a mixture of legitimate and fraudulent data, where the mixing coefficient was unknown and changes in time. The position of data is shown in Figure 4. Call data was labeled to classes of fraud and legal on a subscriber basis. No geographical information about the calls was available in neither the call data nor when the velocity trap gives an alarm. The database of fraudulent behaviour contained call data of 180 subscribers during a period of 75 days. The legal call data duration used was 38 days and it was assumed to contain no fraudulent activity.



(a) - Data belongs to class M_l

(b) - Data belongs to the class M_f

Fig. 4 (a and b): Call data class label position

In representation of the call data, call records constituting the call data are transactions ordered in time. Each of the call records has a set of call attributes. These attributes need to be converted to a form that is compatible with the model used. This conversion can take many forms. The two different data representations used in this paper are features through aggregation in time and dynamic description of call data. Input data for the model was created through feature extraction. In feature extraction, descriptors of the region are calculated from unprocessed data, which involves aggregation. As in Taniguchi et al (1998), the detection was based on feature attributes resulting from call data. The unit of aggregation in time is a day and the feature mapping transforms the transaction data ordered in time to static variables inherent in feature space. The features used signify the usage of an account each day, the number of calls made, and the total length of calls.

In the aspect of dynamic description of call data, there was a connection between the length of the aggregation period used in feature extraction and the richness of description. This representation described the immediate behaviour of mobile phone subscribers by moving the length of the aggregation period to the minimum at the price of a representational richness as used by Hollm´en and Tresp (1999). Here, the call data was sampled for one minute interval, and the data indicated whether a mobile phone was used during a particular time and this describes the exact calling behaviour of mobile phone subscribers, permitting the dynamic modeling of the calling behaviour expressed with transitions from one time step to another.

RESULTS AND DISCUSSION

User profiling is the process of modeling characteristic aspects of user behaviour. In user classification, users are assigned into distinctive groups. There are several models that can be utilized such as hidden markov model (HMM) and hierarchical regime-switching model (HRSM), and finite mixture model (FMM). FMM was used in this study.

Probabilistic networks: Probabilistic networks allow an efficient description of multivariate probability densities (Cowell et al, 1999). Probabilistic formulations allow quantifying uncertainty in the conclusions made about the problem, which makes the framework of probabilistic networks appealing for real-world problems. Conditional independence was used in defining qualitative relationship between the variables, while the distributional assumptions defined the quantitative aspect of the probabilistic networks.

Learning by EM algorithm: Learning is the process of estimating the parameters of a model from the available set of data. The maximum likelihood estimate for the parameters maximizes the probability of the data for a given model. The EM algorithm is an iterative algorithm for estimating maximum likelihood parameters in incomplete data problems (McLahlan 1996). Incomplete data means that there is a many-to-one mapping between the hidden state space and the observed measurements. Since it is impossible to recover the hidden variable, EM algorithm works with its expectation by making use of the measurements and the implied form of the mapping in the model. The EM algorithm was guaranteed to converge monotonically to a local maximum of the likelihood function as in Xu and Jordan (1996).

The expected log likelihood of the complete data is introduced as

$$Q(\O|\O^{(old)}) = E(logP(Y,S|\O)|Y, \O^{(old)}) \qquad (1)$$

$$=\int_s logP(Y,S|\O)|P(S|Y, \O^{(old)})$$

where the log-likelihood of the complete data is parameterized by the free parameter value $\O$ and the expectation is taken with respect to the second distribution parameterized by the current parameters $\O(old)$ (Dempster et al, 1977). In the E-step, the Q function in the equation is computed. In the M-step, the parameter values are updated to be

$$\O^{(new)}=arg \max_{\O} Q(\O)| \O^{(old)}) \qquad (2)$$

The solution used for the maximization problem was by setting the derivatives of the maximized function to zero and solving for $\O$.

Self-Organizing Map (SOM): The Self-Organizing Map (SOM) is a neural network model for the analysis and visualization of high -dimensional data. It was invented by Teuvo Kohonen (1995) and is the most popular network model based on unsupervised, competitive learning. SOM has been used in a wide range of applications (Kaski et al. 1998). It has also been applied for the analysis of industrial processes (Alhoniemi et al, 1999). The architecture of the SOM consists of a collection of neurons located at nodes and connections among the nodes. Each node has an associated set of input weights. The SOM provides a topology-preserving mapping from high dimensional space to map units. The map unit is the output layer, a two-dimensional array of nodes that is fully connected to the input layer. The property of topology preservation refers to maintaining the relative distance between the points. The points that are closer to each other in input space are mapped close together on the map unit (output) in the SOM. The distance between the map units can also be defined according to their topological relationship. SOM can thus be used as a cluster-analyzing tool of high dimensional data. An important property associated with SOM is its ability to generalize. This means that the network is able to recognize inputs that it has never encountered before. After training SOM, a vector is presented to the input layer, and the node whose weight vector is most similar to this input vector will be activated. The diagram of a typical SOM neural network is shown in Figure 5.

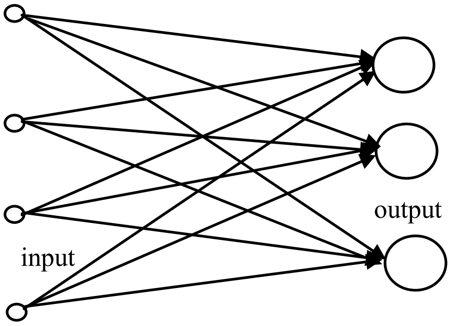


Fig. 5: A simple self-organizing map neural network

The SOM is a collection of prototype vectors, between which a neighborhood relation is defined. This neighborhood relation defines a structured lattice, usually a two-dimensional, rectangular or hexagonal lattice of map units. After initializing the prototype vectors with random values, training takes place. Training the SOM from data was divided into two steps, which are applied alternately. The best matching unit (BMU) or a winner unit m^c was first searched for to minimizes the Euclidean distance between a data sample x and the map units m^k

$$c = arg\ \underset{k}{min}\ //x - m^k// \qquad (3)$$

Then, the map units are updated in the topological neighborhood of the winner unit. The topological neighborhood is defined in terms of the lattice structure, not according to the distances between data samples and map units. The update step can be performed by applying $m^k(t + 1) := m^k(t) + \alpha(t)h^c(t, k)[x(t) - m^k(t)]$ where the last term in the square brackets is proportional to the gradient of the squared Euclidean distance $d(x,m^k) = ||x - m^k||^2$. The learning rate $\alpha(t) \in [0, 1]$ must be a decreasing function of time and the neighborhood function $h^c(t, k)$ is non-increasing function around the winner unit defined in the topological lattice of map units (Hollm´en, 2000). During learning, the learning rate and the width of the neighborhood function are decreased in a linear fashion. The map then tends to converge to a stationary distribution, which reflects the properties of the probability density of data.

SOM for clustering probabilistic models: A SOM algorithm, which enables using probabilistic models as the cluster models was used. In this approach, the map unit indexed by k stores the empirically estimated parameter vector θ^k with an associated probabilistic model $q(x; \theta^k)$. In implementing a SOM algorithm, the distance between the map units (i.e. the θ^k) and data was defined. The distance between θ and a data point itself cannot be defined in Euclidean space since they may have different dimensionality but the distance measured was between probability distributions using the Kullback-Leibler distance, which relates two probability distributions. Minimizing the Kullback-Leibler distance between the unknown true distribution that generated the data point at hand and the empirical model leads to minimizing the negative logarithm of the probability of the data with the empirical model. This justified the use of the probability measure as a distance measure between models and call data.

The described fraud detection task can be considered as pattern recognition or classification problem. The set X_n of all call activity is divided into two disjoint subsets: legal call activity$X^1{}_n \subseteq X_n$ and fraudulent call activity $X^1{}_n \subseteq X_n$, $X^1{}_n \cap X^f{}_n = Ø$. If the points in some multidimensional space of fraudulent and legitimate call data belong to different areas in this space, then it is possible to make a decision about the image of a new transaction x^{n+1}.

The following two hypotheses are considered as a basis for such classification.

1. *Hypothesis H_l* : Transaction $x^{n+1} = (x^{n+1}{}_1, \ldots, x^{n+1}{}_m)$ on call data activity c_d is similar to all previous transactions from the set X_{cd} , which were carried out earlier by the subscriber. If hypothesis H_l is confirmed for transaction x^{n+1} , then the transaction x^{n+1} is classified as legal and included into the set $X^f{}_n$.

2. *Hypothesis H_f* : Transaction $x^{n+1} = (x^{n+1}{}_1, \ldots, x^{n+1}{}_m)$ is similar to earlier executed fraudulent transactions $X^f{}_n = \{x^i$ – *considered fraudulent* $| x^i \in X_n\}$. If hypothesis H_f is confirmed for transaction x^{n+1}, then transaction $x^{n+1}$is classified as fraudulent and included into the set $X^f{}_n$.

Neural network techniques was was for clustering and classification in order to check the proposed hypotheses H_l and H_f . The main idea is to create and later recognize pattern of "legal subscribers" and pattern of "fraudster" on the basis of neural network "learning" from the transactions X_n executed earlier and to develop "rules" of subscriber's behaviour and fraudster's behaviour. Learning algorithms allow the system to follow the subscriber's behaviour and adapt to changes in it. If a call transaction does not correspond to the pattern of "legal subscriber/call data" or is similar to the "fraudulent" pattern it is classified as suspicious for fraud.

The SOM was used for testing the hypotheses H_l and H_f. Testing the hypothesis H_l for call transaction x^{n+1} on call data c_d includes the following steps:

1. Create a typical subscriber's call behaviour pattern model $\underline{W_{cd}}$ on the basis of past call activities $X_{cd} \in X^l{}_n$ executed earlier with the call data c_d. This model W_{cd} is represented as a SOM, which is the subscriber's call profile.
2. Determine the similarity rate $\delta(x^{n+1}, W_{cd})$ of transaction x^{n+1} to profile W_{cd}.
3. Hypothesis H_l is accepted if the similarity rate $\delta(x^{n+1}, W_{cd})$ satisfies the condition
4. $\delta(x^{n+1}, W_{cd}) \leq e_l$ where e_l is some set parameter (that is, the threshold/boundary value for the degree of similarity of the transactions on call data c_d to the profile W_{cd}, making it possible to cut off call transactions that deviate from the early established norm and to control the accuracy of fraud detection).

Testing the hypothesis H_f for transaction x^{n+1} is performed as follows:

1. Create a typical fraudster's behaviour pattern model W_f on the basis of fraudulent transactions $X^f{}_n$ executed earlier in call data transaction in mobile communication and determined as fraudulent. This model W_f was also represented as a SOM, which is the fraudster's profile.
2. Determine the similarity rate $\delta(x^{n+1}, W_f)$ of transaction x^{n+1} to profile W_f.
3. Hypothesis H_f is accepted if the similarity rate $\delta(x^{n+1}, W_f)$ satisfies the condition $\delta(x^{n+1}, W_f) \leq e_f$, where e_f is some parameter.

The proposed method for call transaction analysis is represented as a block diagram in Figure 6. The process of call data transaction monitoring consists of three stages: data accumulation, training/building of subscriber's profile and call activity control.

At the stage of data accumulation, the call data about the transactions on subscriber c_d are collected in the database DB. If the size of X_{cd} exceeds some predefined level, sufficient to build an adequate profile, then the monitoring process goes to stage two.

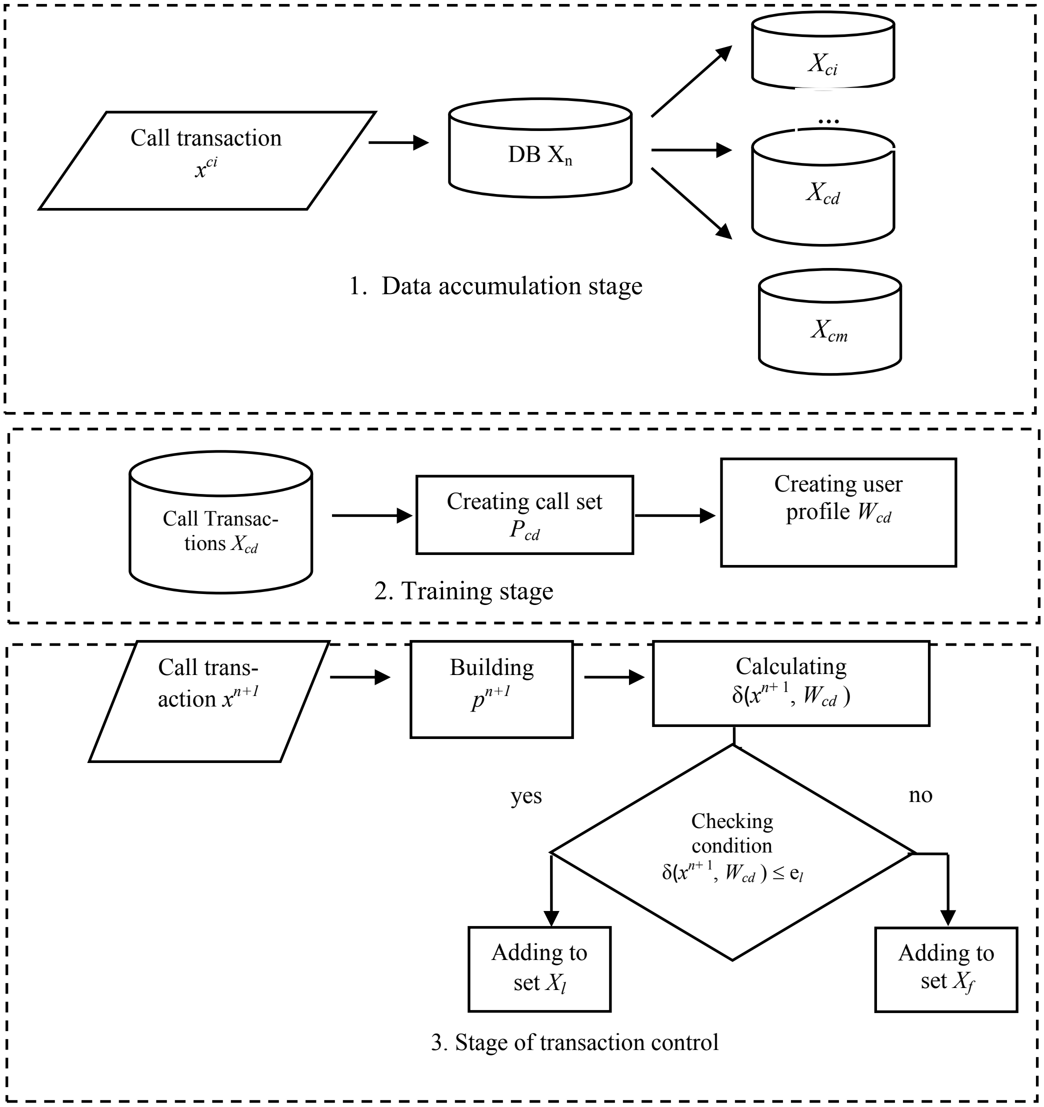
At stage two, the training stage, the subscriber's profile W_{cd} is created as follows:

- The set P_{cd} is built using the function φ;
- The neural network is trained on the basis of set P_{cd};
  The profile
  $W_{cd} = \| w^s{}_d \|_{s=1;\, d}$
  ${}_{d=1;M}$ is built as a result of training.

After the training stage, the process goes to the stage of call transaction control, which consists of the following:

- The vector p^{n+1} is built applying the function φ to every new call activity x^{n+1}: $p^{n+1} = \varphi(x^{n+1})$;
- The deviation of the current call transaction x^{n+1} from the profile W_{cd} created at the training stage was calculated as $\delta_0 = \delta(x^{n+1}, W_{cd})$
- The value δ_0 is compared with the threshold ε_l fixed for the profile W_{cd} where ε_l is a boundary value for the degree of similarity of the call transactions on subscriber c_d to the profile W_{cd}. This made it possible to cut off call transactions that deviate from the established norm and to control the accuracy of fraud detection.
- If $\delta_0 \leq \varepsilon_l$ then call transaction x^{n+1} is considered as legal and the vector x^{n+1} is added to the set $X_l = X_{cd}$;
- If $\delta_0 \leq \varepsilon_f$ then call transaction x^{n+1} is considered suspicious for fraud and is added to the set X_f for further expert analysis.

The principles underlying detection software are grounded in classical statistical decision theory (Cavusoglu and Raghunathan, 2004). There are two sources that generate inputs to the detection software: normal (H_0) and fraudulent (H_1). The normal source generates legally authorized call activity. The fraudulent source generates illegal or fraudulent call activity. In this study, a large percentage of transactions are legal. The skewed nature of the frequency distribution makes detection of illegal transactions difficult. The detection software observes the transaction but does not know whether it came


Call transaction x^{ci}
DB X_n
X_{ci}
...
X_{cd}
X_{cm}
1. Data accumulation stage
Call Transactions X_{cd}
Creating call set P_{cd}
Creating user profile W_{cd}
2. Training stage
Call transaction x^{n+1}
Building p^{n+1}
Calculating $\delta(x^{n+1}, W_{cd})$
yes
no
Checking condition $\delta(x^{n+1}, W_{cd}) \le e_l$
Adding to set X_l
Adding to set X_f
3. Stage of transaction control

from a normal or fraudulent source. The goal of the detection software was to classify each transaction as legal or fraudulent. The types of errors that can occur in this classification are:

i) Classification of a fraudulent transaction as legal (false negative); and

ii) Classification of a legal transaction as fraudulent (false positive)

- Probability of detection = $P_D = P_r$ (classify into $H_1 \mid H_1$ is true) or
- Probability of false negative = $1 - P_D$
- Probability of false positive = $P_F = P_r$ (classify into $H_1 \mid H_0$ is true)

If the numerical values for the normal and fraudulent call activity follow exponential distributions with parameters λ_N and λ_F, $\lambda_N > \lambda_F$ respectively, then the probability of detection P_D and probability of false positive P_F will be expressed as

$$P_D = \int^{\infty} \lambda_F e^{-(\lambda_F x) dx = e^{-\lambda_F t}} \quad (4)$$

$$P_F = \int^{\infty} \lambda_N e^{-(\lambda_N x)_{dx = e^{-\lambda_N t}}} \quad (5)$$

Then, P_D can be expressed as a function of P_F as

$$P_D = P_F^r \quad (6)$$

where r = λ_F/λ_N is between 0 and 1.

Trees (2001) stated that the quality profile of most detection software is characterized by a curve that relates its P_D and P_F known as the receiver operating characteristic curve (ROC). ROC is a function that summarizes the possible performances of a detector. It visualizes the tradeoff between false alarm rates and detection rates, thus facilitating the choice of a decision function. The effectiveness of this detection software is measured in terms of the classification errors, which consist of system detection rate and false alarm rate. The data used in the application were collected from a telecommunication firm, which consist of call data made per day during the observed period. The performance analyses of the respective detection algorithms are carried out using MATLAB software package and the results compared with the collected data are as shown in Figure 7. In Figure 7, it can be seen that the model results compared satisfactorily well with the collected call data results.

All data held by the system was kept in an embedded database that cannot be connected from outside of the application. All call data account specific information was anonymous and encrypted. Access to the system was controlled by an authentication system that allows the telecommunication firm to control access to information. Users and workstations are subjected to authentication. All user activity was logged.

The use of user profiling, probabilistic and neural network models reduced the classification of legitimate transactions as fraudulent, and ensured accurate and reliable results. The baseline model generated represents legitimate behaviour and then attempt to detect observations/

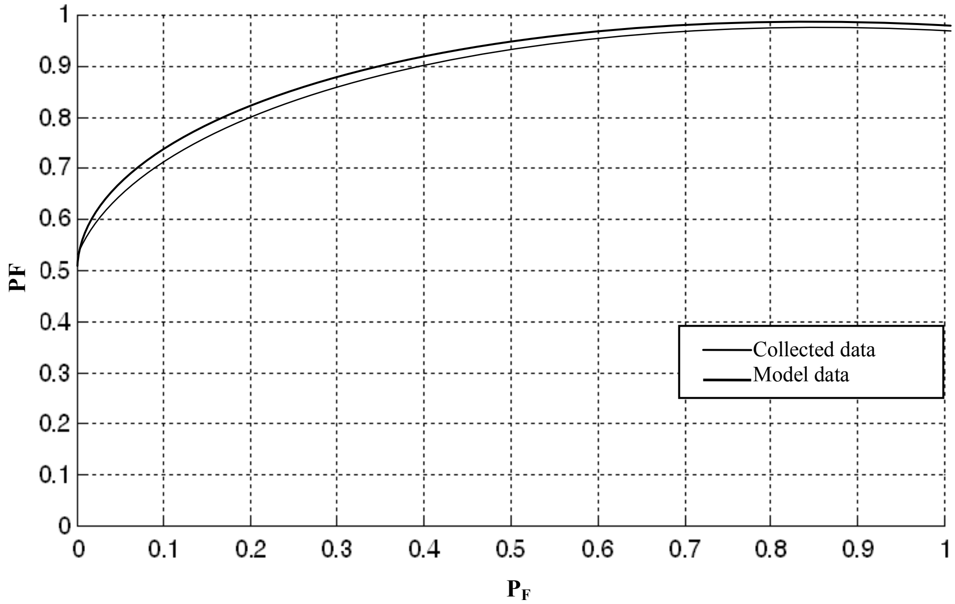


Fig. 7: ROC for mobile communication fraud detection

transactions that show greater departure from this normal transaction helping in the effectiveness of the mobile communication fraud detection system developed. All noted suspicious transactions were accurately established to be fraudulent during testing and evaluation.

The data used was a mixture of legitimate and fraudulent with an unknown mixing mechanism. The mobile communication detection system detected most of the fraudulent transactions. The probability of detection, that is, probability of false positive was below 3 percent. The methods are shown to be effective in detecting system with data from real telecommunication environment. This study reinforced the validity and efficiency of detecting fraud in mobile communication using probabilistic and neural network model as a research tool and laid a solid groundwork for intelligent detection methodologies to be used in an operational fraud detection system.

CONCLUSION

User profiling and classification are important tasks in data intensive environments where computer assisted decision-making is sought for. The calling behaviour is described by the subscriber's call data and was used in this study as a basis for modeling. The goal for the learning methods in this study was to learn user profiles from the call data in order to make decisions about fraud occurrence. The methods presented in this study learn to detect fraud from partially labeled data, in that a call account was known to be defrauded but not exactly when. The data is therefore a mixture of legitimate and fraudulent data with an unknown mixing mechanism.

The models used in fraud detection were probabilistic models and neural networks. The ability to learn from data was considered an important asset of these models, as was the capability to process uncertainty, which was present in the fraud domain. Modeling was performed on user level, user profile level and class level, of which the user profile level was seen to be the most appropriate. Discriminative training was utilized for tuning the models for best diagnostic accuracy.

REFERENCES

Alhoniemi, E., Hollm´en, J., Simula, O., and J. Vesanto (1999). Process monitoring and modeling using the self-organizing map. *Integrated Computer Aided Engineering 6* (1), 3–14.

Barson, P., Field, S., Davey, N., McAskie, G., and Frank, R. (1996). The detection of fraud in mobile phone networks. *Neural Network World 6*(4), 477–484.

Burge, P. and Shawe-Taylor, J. (1997). Detecting cellular fraud using adaptive prototypes. In *Proceedings of AAAI-97 Workshop on AI Approaches to Fraud Detection & Risk Management*, AAAI Press. 9–13.

Cavusoglu, H. and Raghunathan, S. (2004) Configuration of Detection Software: A Comparison of Decision and Game Theory Approaches. Decision Analysis Vol. 1, No. 3, Pp. 131–148

Cowell, R., Dawid, A., Lauritzen, S., and Spiegelhalter, D. (1999). *Probabilistic Networks and Expert Systems*. Statistics for Engineering and Information Science. Springer-Verlag.

Davis, A. B. and Goyal S. K. (1993). Management of cellular fraud: Knowledge-based detection, classification and prevention. In *Proceedings of the 13th International Conference on Artificial Intelligence, Expert Systems and Natural Language, Avignon, France*, Vol. 2, 155–164.

Dempster, A. P., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B 39*, 1 –38.

Fawcett, T. and Provost F. (1997). Adaptive fraud detection. *Journal of Data Mining and Knowledge Discovery 1*(3), 291–316.

Hollm´en J (2000). User profiling and classification for fraud detection in mobile communications networks. Dissertation for the degree of Doctor of Science in Technology. Helsinki University of Technology.

Hollm´en, J. and V. Tresp (1999). Call-based fraud detection in mobile communication networks using a hierarchical regime-switching model. In M. Kearns, S. Solla, and D. Cohn (Eds.), *Advances in Neural Information Processing Systems 11: Proceedings of the 1998 Conference (NIPS'11)*, pp. 889–895. MIT Press

Johnson, M. (1996). Cause and effect of telecoms fraud. *Telecommunication (International Edition)30* (12), 80–84.

Kaski, S., Kangas, J., and Kohonen, T., (1998). Bibliography of selforganizing map (SOM) papers: 1981-1997. *Neural Computing Surveys 1*, 102–350.

Kohonen, T. (1995). *Self-Organizing Maps*. Springer-Verlag.

Kumar, S. (1995). *Classification and detection of computer intrusions*. Ph. D. thesis. Purdue University.

McLahlan, G. J. (1996). *The EM Algorithm and Extensions*. Wiley & Sons.

Ogwueleka, F.N and Inyiama, H.C (2009). Credit card fraud detection using artificial neural networks with a rule-based component. The Icfai University Journal of Science and Technology, Vol. 5. No. 1.

O'Shea, D. (1997). Beating the bugs: Telecom fraud. *Telephony 232*(3), 24.

Shortland, R. and Scarfe, R. (1994). Data mining applications in BT. *BT Technology Journal 12*(4), 17–22.

Taniguchi, M., Haft, M., Hollm´en, J., and Tresp, V. (1998). Fraud detection in communications networks using neural and probabilistic methods. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'98)*, Volume II, pp. 1241–1244.

Trees, H. V. (2001). Detection, Estimation and Modulation Theory-Part I. John Wiley, New York.

Xu, L. and Jordan, M. (1996). On convergence properties for EM algorithm for gaussian mixtures. *Neural Computation 8*, 129–151.