# Neural Fraud Detection in Credit Card Operations

José R. Dorronsoro, Francisco Ginel, Carmen Sánchez, Carlos Santa Cruz

J. Dorronsoro and C. Santa Cruz are with the Department of Computer Engineering and the Instituto de Ingeniería del Conocimiento, Universidad Autónoma de Madrid, 28049 Madrid, Spain

C. Sánchez is with the Sociedad Española de Medios de Pago (SEMP), López de Hoyos 151, 28002 Madrid, Spain; F. Ginel, formerly with SEMP, is now with Visa International OCW, San Francisco, California.

J. Dorronsoro and C. Santa Cruz gratefully acknowledge partial support from Spain's CICyT under grant TIC 95–965.

## Abstract

This paper presents an on line system for fraud detection of credit card operations based on a neural classifier. Since it is installed in a transactional hub for operation distribution, and not on a card issuing institution, it acts solely on the information of the operation to be rated and of its immediate previous history, and not on historic databases of past cardholder activities. Among the main characteristics of credit card traffic are the great imbalance between proper and fraudulent operations, and a great degree of mixing between both. To ensure proper model construction, a nonlinear version of Fisher's discriminant analysis, which adequately separates a good proportion of fraudulent operations away from other closer to normal traffic, has been used. The system is fully operational and currently handles more than 12 million operations per year with very satisfactory results.

## Keywords

Credit card fraud detection, neural networks, non linear discriminant analysis.

# I. Introduction

Pattern recognition is certainly one of the most relevant areas of application of neural networks. The range of concrete problems that may fall under this category is very wide. Probably one of the main sources of applications arises from "physical patterns". By this we mean those coming from all sorts of signals, acoustic, graphical or others. But another area of great practical interest and active research is the classification of what we may call social or economical patterns. A clear instance are the long and widely used systems for credit scoring. What they essentially have to decide is whether or not a given petition is credit worthy or, instead, should be rejected. The pattern here is a set of characteristics, financial, job related, familial or other, of the credit applying person. This problem model is extendible to many other interesting instances: insurance policy decisions, financial ratings, or to the topic of this work, acceptance or rejection of credit card operations.

This last problem has, of course, received a great deal of attention, under many different approaches, some of them neural network based [2], [7]. There are even several commercial fraud detection systems with large neural components [9], [1]. In any case, it certainly has some peculiar characteristics of its own, deriving from the different usages that such a card can have. For instance, a customer may choose to take full advantage of his card's credit capabilities. Fraud on the card being used is a possibility, but even if that is not the case, the credit worthiness of the card holder should also be somehow taken into account. We won't deal with this issue here, and will concentrate only on the detection of possibly fraudulent operations. By this we mean the usage of a given card that may have been lost, stolen or falsified, by an unauthorized person against the will of its true owner. Certainly this possibility has to be taken into account even if the card's main use is as a credit device. But in some countries, as it is the case of Spain, most cards, rather that being used to finance purchases, simply provide a certain deferment of the payments due. Fraud is then the paramount risk issue.

Credit card fraud detection also has two other highly peculiar characteristics. The first one is obviously the very limited time span in which the acceptance or rejection decision has to be made. The second one is the huge amount of credit card operations that have to be processed at a given time. The situation in

Spain is fairly typical and self explanatory: more than 1.2 millions of Visa card operations take place in a given day, 98 % of them being handled on line. Of course, just very few will be fraudulent (if not, the entire industry would have soon ended up being out of business), but this just means that the haystack where these needles are to be found is simply enormous.

When considering the information to be used to rate a given operation, two distinct possibilities arise. In the first one, that we may call "by–owner", operations are rated according to the usage history of the card owner. This approach requires the ability to fetch in a very fast way the pertinent owner's information from the usually huge databases of all card holder's historical records. We have to clarify what "fast" means in this setting. Fraud prevention is very important to card issuers, but not to the point of making impractical or simply inconvenient the daily card utilization by hundreds of thousands of customers. When all the time needed for the remote connections and the basic operation processing is taken into account, a fraud detection system usually has no more than a rather small fraction of a second to perform its task. This alloted time most likely will not be enough for large database queries. Of course, this may be alleviated if specially configured and dedicated hardware is available, but it will certainly result in higher start up and maintenance costs.

In any case, it is also true that such systems have additional advantages going for them. The most important is certainly the deep and powerful analysis that the past history of a customer allows when rating a certain operation. Moreover, the information collected in this type of systems may have an added value for other customer management tasks, such as marketing. Furthermore, these systems may very well work in a sort of "deferred on line" mode: even if a given operation has to be authorized before an eventually negative rating has been completed, further incoming operations of its card will be effectively blocked. Notice that fraud may be fought over individual transactions, but it is won on the sum of all of them. Moreover, given the nature of stolen or falsified card "users", they will shift their attention from issuers with globally effective prevention systems to others less prepared.

There are situations, however, where a "by–owner" system is simply impossible to set up. This is case when a detection system is to be installed not at an issuer, but rather in an "operation hub", that is, a central operation processing center that receives transactions from many sellers, distributes them to each particular card issuer, and relays its answer back to the originating sales point. This is precisely the situation of the Sociedad Española de Medios de Pago (SEMP).

SEMP receives more than 60 % of the Visa traffic generated in Spain and, through Visa Net, all the foreign operations due to Visa cards issued by Spanish credit institutions members of Visa España. It is important to point out that more than 95 % of SEMP's Visa traffic is processed on line. The issuers are in principle responsible of authorizing or denying a given operation, but SEMP can also intervene in the authorization process through its National Unified Authorization Center. Moreover, SEMP works on behalf of its members (more than 75 % of Spain's credit institutions) in Visa card fraud prevention, using

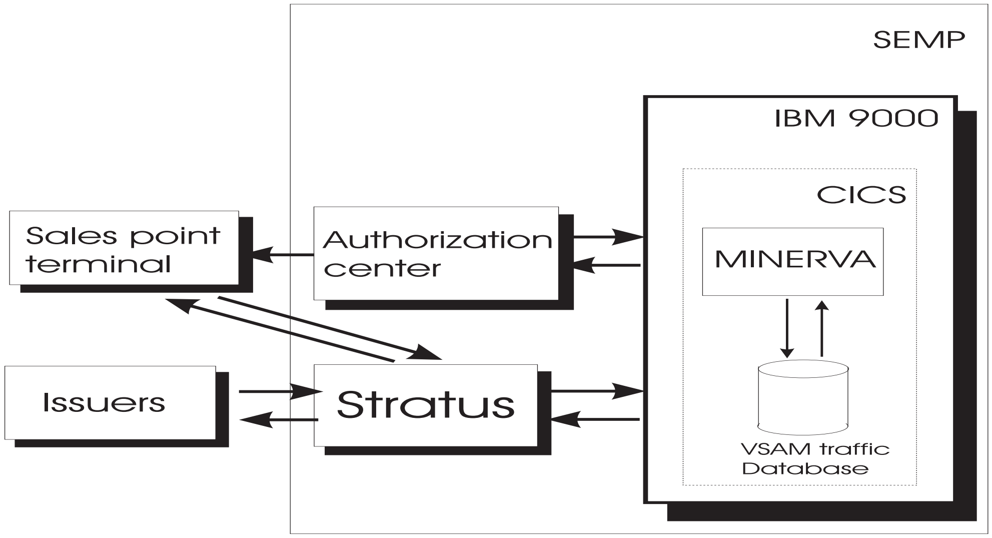

Fig. 1. Minerva's placement in SEMP's transactional environment.

for that purpose several complementary tools.

In any case, SEMP's most important activity is to streamline Visa traffic back and forth between sellers and issuers: therefore, it essentially does not have any owner information. Thus, contrary to what happens for instance to card issuing institutions, SEMP is not a good candidate for a "by–owner" system. An alternative solution is to build a scoring system "by–operation", that is, a system that only uses the information of the operation itself. When complemented with the previous operation history of the associated card over a period of time, variables such as operation frequency, accumulated amounts, acceptance/rejection rates of previous operations, and so on, can be derived. These variables are obviously of great interest when deciding whether the current operation is legal or not.

Certainly, this information is not as complete as the one that could be used in a by–owner system, but this approach has some clear strong points:

- since only the operations of an immediately previous period of time are considered, data querying is done on relatively small databases, and operation scoring and authorization can be performed in real time, without requiring deferred processing;
- the small database sizes make possible to install such a system without having to use large dedicated hardware, thus reducing sensibly its costs;
- its placement on a operation hub allows it to simultaneously service several issuers, without having to install individual systems at each of them.

SEMP's by–operation system, code named Minerva, and developed jointly with IBM Spain and the Instituto de Ingeniería del Conocimiento (IIC), takes advantage of these properties. SEMP uses 3 Stratus mainframe hosts to handle Visa traffic. They are simultaneously connected to a very large number of sales point terminals and to the various issuer's mainframes (total traffic exceeds 250 millions of operations/year). They are also connected to an IBM 9000 mainframe, that keeps an on line track of all operations for different purposes; this machine (see figure 1) also hosts Minerva, under the well known and widely extended MVS–CICS transactional operating environment.

Minerva has several rating modules. Some of them work in a parametric (i.e., rule based) mode; however, Minerva's core modules incorporate neural rating functions. In the rest of this paper we will describe the main issues that were faced in their construction and evaluation.

## II. Credit card operation data and neural model construction

As mentioned before, fraud detection in credit card operation falls neatly in principle within the scope of pattern recognition procedures, and its solution can be sought through the construction of appropriate classifier functions. However, and as it may also be expected, it has certain characteristics of its own that make it a rather difficult problem. The most important is the great imbalance between good operations and fraudulent ones. This is not surprising at all: after all, card issuers intend to make a profit out of credit card use, and fraud directly diminishes that profit. Thus, they will already have implemented a number of fraud prevention methods. In other words, new fraud detection tools, neural or other, are weapons to be used in a war already being fought.

From the point of view of the construction of neural detectors, this imbalance will certainly make model training rather difficult. In fact, typical fraud rates may well be in the one per tens of thousands. This simply means that prior to the model construction, some kind of data segmentation has to be applied in order to lower these rates in the data sets to be used. In Minerva's case, segmentation criteria were defined after a thorough statistical analysis of legal and fraudulent traffic among the different geographical and sector areas for which independent detection modules were to be built.

That segmentation effectively lowered the fraudulent–to–legal operation rate to an average of 1 per 150. In any case, training set sizes are still not balanced and, as it is well known in the neural community, this has to be properly dealt with. In the case of multilayer perceptrons (MLPs), the influence of training set sizes was studied by Webb and Lowe [12]. To build a MLP–based classifier, the most common target labeling for a general $C$ class problem, is to assign to class $i$ patterns the target vector $e_i = (0, \ldots, 1, \ldots 0)$, where the 1 value is assigned to the $i$–th coordinate.

However, as pointed out in [12], the above target labeling implicitly incorporates class sizes, in such a way that larger classes are heavily favored in contrast with smaller ones. To remedy this, they propose the coding of class $i$ patterns by means of the vectors $e_i' = (0, \ldots, 1/\sqrt{N_i}, \ldots 0)$, with $N_i$ the sample number of elements in class $i$. Thus, class size biases can be corrected with the proper incorporation of class size information. The problem is that, in some cases, the information needed, which essentially coincides with class membership prior probabilities, may not be readily available.

Credit card fraud detection is just one of those cases. To begin with, we have just mentioned the possibly large class size differences. Moreover, many external factors affect attempted fraud, which may result in heavy variations among distinct class samples. Prior probabilities are therefore rather difficult to estimate properly, and may result in inadequate target codings and, eventually, in the lack of convergence of MLP training.

Another factor that may handicap conventional MLP training is class overlapping, that is, the occurrence of patterns with different target labelings but very similar inputs. This is rather frequent in fraudulent card operations; in fact, a common technique to lengthen the "operating life" of a bad card is to hide its nature by using it in transactions with "normal" characteristics. After they are detected, these operations are of course considered as fraudulent, but they might otherwise be very similar to a large number of legal ones. If left in training databases, they can "muddle" network training, to the point that resulting MLPs, which of course will not be able to tell them apart from their legal counterparts, neither will be capable of detect fraud operations with more salient features. On the other hand, manual database removal of these operations will probably end up being just too costly, if possible at all.

It is thus clear that markedly different class sizes, difficult target codings and class overlapping, have to be carefully taken into account when building a neural fraud detection system. In particular, the usual error minimizing MLP training procedures may fail entirely over training sets derived from legal and fraudulent traffic; other solutions have to be devised and utilized. In principle, the difficulties easier to remove are those due to target coding. In fact, there are classification procedures that do not need it, the simplest and well known certainly being Fisher's discriminant analysis ([6], [12] discuss the relationship of neural classifiers with Fisher's method).

In Fisher's procedure [4] an error function minimum is not directly sought. Instead it looks for linear projections of input patterns that minimize a certain criterion function, usually the ratio of the determinants of the between and within class variances, $\tilde{\boldsymbol{S}}_B$ and $\tilde{\boldsymbol{S}}_W$ respectively, of target projections; that is, the function $\mathcal{J}(\boldsymbol{W}) = |\tilde{\boldsymbol{S}}_W|/|\tilde{\boldsymbol{S}}_B|$, where $\boldsymbol{W}$ denotes the vector of projecting weights. Notice that this criterion function does not require target coding at all, and therefore prior probabilities do not enter into model construction (of course they do if posterior probabilities are to be computed).

In any case, the linear nature of Fisher's pattern transformation makes it unsuitable for a problem such as fraud detection, which one hardly expects to be linear. However, a natural alternative suggests itself: to combine the coding–independent nature of a procedure such as Fisher's with the nonlinear projecting capabilities of MLPs. This has been precisely the approach followed in the construction of Minerva's neural modules; we will briefly describe it in the following section. As we will also show, besides freeing network training from difficult class size estimations, this approach will also handle adequately the network robustness issues derived from class overlapping.

## III. Non linear discriminant analysis

The non linear discriminant analysis (NLDA) neural models used in Minerva have the usual MLP architecture: one input layer, that receives operation variables, both direct and derived, several hidden

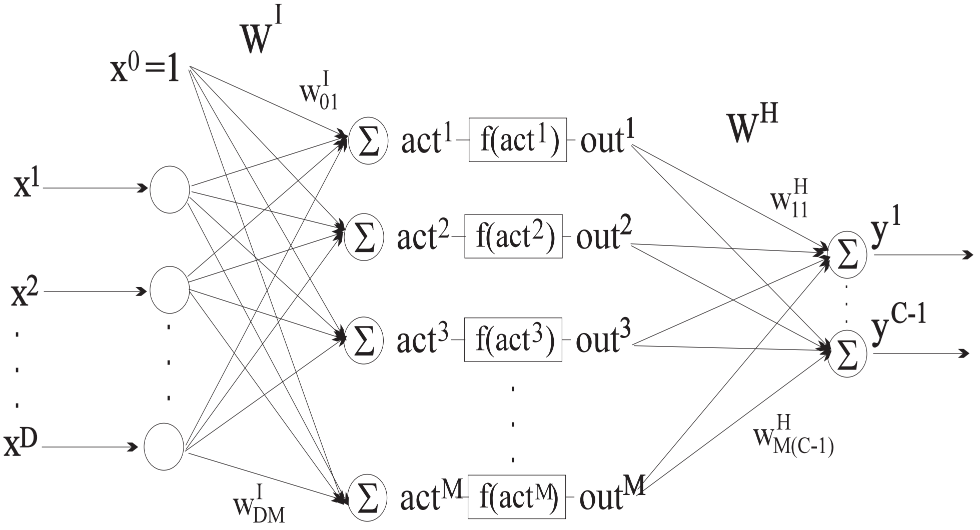

Fig. 2. Architecture of a NLDA net and notational conventions used in the paper.

layers to nonlinearly transform them, and an output layer which, as in classical Fisher's analysis, has $C-1$ units (one less the number $C$ of classes). The unit connections will be sigmoidal for all layers except the linear ones going from the last hidden layer to the output units. This architecture tries to exploit the well known universal approximation properties of MLPs [8].

However, network training is markedly different now. A complete discussion can be found in [11]; we will simply sketch its main points. To begin with, the function to be minimized can be any of those usually applied in Fisher's analysis (see [5], ch. 9). To simplify matters, we will concentrate our discussion upon the most widely used, Fisher's original one:

$$\mathcal{J}(\boldsymbol{W}^H) = \frac{|\tilde{\boldsymbol{S}}_W|}{|\tilde{\boldsymbol{S}}_B|} = \frac{|(\boldsymbol{W}^H)^t \boldsymbol{S}_W \boldsymbol{W}^H|}{|(\boldsymbol{W}^H)^t \boldsymbol{S}_B \boldsymbol{W}^H|} \tag{1}$$

where by $|A|$ we denote the determinant of a matrix $A$ (the rationale for the use of determinants here is that, being the product of the eigenvalues, they measure the volume of the dispersion of the scatter matrices on their eigendirections). Moreover, $\boldsymbol{W}^H$ denotes the weight vector connecting the last hidden and the output layers. We refer to [4] for the concrete definition of these matrices; notice that in this situation, between and within scatter matrices at those layers are then related by the formulae $\tilde{\boldsymbol{S}}_W = (\boldsymbol{W}^H)^t \boldsymbol{S}_W \boldsymbol{W}^H$ and $\tilde{\boldsymbol{S}}_B = (\boldsymbol{W}^H)^t \boldsymbol{S}_B \boldsymbol{W}^H$.

Weight adjustment in network training is done by succesive layer weight optimization. To keep things simple, we will concentrate in a single hidden layer network, such as the one depicted in figure 2. The network transfer function $\boldsymbol{F}(\boldsymbol{X}, (\boldsymbol{W}^I, \boldsymbol{W}^H))$, with $\boldsymbol{W}^I$, $\boldsymbol{W}^H$ denoting weight vectors at the input and hidden layers (see figure 2), can be seen as either one of two functions $\boldsymbol{G}(\boldsymbol{X}, \boldsymbol{W}^I)$, $\boldsymbol{H}(\boldsymbol{X}, \boldsymbol{W}^H)$, where in each of them the weights not written out are assumed to be constant. Now, once a weight set $\boldsymbol{W}_k$ has been obtained, $\boldsymbol{W}_{k+1}$ is computed in a two step fashion. We first compute $\boldsymbol{W}^H_{k+1}$ by optimizing the criterion function (1) with respect to the outputs of $\boldsymbol{H}(\boldsymbol{X}, \boldsymbol{W}^H)$. Notice that since those outputs are given by a linear transformation of the hidden unit outputs with respect to the weights $\boldsymbol{W}^H$) we simply have to perform multiple $C$ class discriminant analysis on the pattern vectors provided by these hidden unit outputs. This is done by the well known Fisher's eigenvalue and eigenvector computations (see [4], pp. 115–121).

Once $\boldsymbol{W}^H_{k+1}$ has been obtained, and the optimal partial transfer function $\boldsymbol{H}(\boldsymbol{X}, \boldsymbol{W}^H)$ at this point defined, we proceed to compute the corresponding optimal weights $\boldsymbol{W}^I_{k+1}$ by a quasi–Newton procedure (see [10]) applied to the criterion function based now in the outputs of $\boldsymbol{G}(\boldsymbol{X}, \boldsymbol{W}^I)$. The required gradient computation for these weights is somewhat more involved; we will describe it immediately. However, we point out that for a general network with several other hidden layers, once $\boldsymbol{W}^I$ is known, the weight gradients of the remaining layers can be obtained in a fashion rather similar to the one used in standard backpropagation.

In order to compute the gradients $\frac{\partial \mathcal{J}}{\partial w^I_{kl}}$, with $k = 0, \ldots, D$, $D$ being input pattern dimension, and $l = 1, \ldots, M$, with $M$ the hidden layer unit number, we will use the hidden layer projections $out^h_{ij}$ of each pattern vector as intermediate variables. It then follows (see figure 2 for concrete variable labelings) that

$$\frac{\partial \mathcal{J}}{\partial w^I_{kl}} = \sum_{i=1}^{C} \sum_{j=1}^{N_i} \sum_{h=1}^{M} \frac{\partial \mathcal{J}}{\partial out^h_{ij}} \frac{\partial out^h_{ij}}{\partial act^h_{ij}} \frac{\partial act^h_{ij}}{\partial w^I_{kl}}, \tag{2}$$

which taking into the account network architecture reduces to

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial w^I_{kl}} &= \sum_{i=1}^{C} \sum_{j=1}^{N_i} \frac{\partial \mathcal{J}}{\partial out^l_{ij}} f'(act^l_{ij}) x^k_{ij} \\ &= \sum_{i=1}^{C} \sum_{j=1}^{N_i} \frac{|\tilde{\boldsymbol{S}}_B| \frac{\partial |\tilde{\boldsymbol{S}}_W|}{\partial out^l_{ij}} - |\tilde{\boldsymbol{S}}_W| \frac{\partial |\tilde{\boldsymbol{S}}_B|}{\partial out^l_{ij}}}{|\tilde{\boldsymbol{S}}_B|^2} \alpha^{lk}_{ij}, \end{aligned} \tag{3}$$

with $\alpha^{lk}_{ij} = f'(act^l_{ij}) x^k_{ij}$. This expression involves in the general case the derivatives of $(C-1) \times (C-1)$ determinants. This fact rather complicates the ensuing gradient computation (concrete results can be found in [11]). Fortunately, in two class problems (as, for instance, credit card fraud detection), things can be considerably simplified. Observe that then the network output layer has a single unit, that is, $C = 2$. Therefore, determinants can be totally avoided, and the above formula for $\boldsymbol{W}^I$ gradients reduces to

$$\frac{\partial \mathcal{J}}{\partial w^I_{kl}} = \sum_{i=1,2} \sum_{j=1}^{N_i} \frac{1}{\tilde{\boldsymbol{s}}_B^2} \left( \tilde{\boldsymbol{s}}_B \frac{\partial \tilde{\boldsymbol{s}}_W}{\partial out^l_{ij}} - \tilde{\boldsymbol{s}}_W \frac{\partial \tilde{\boldsymbol{s}}_B}{\partial out^l_{ij}} \right) f'(act^l_{ij}) x^k_{ij}.$$

where the two scalars $\tilde{\boldsymbol{s}}_B$ and $\tilde{\boldsymbol{s}}_W$ replace now the more general network output scatter matrices. Because of the symmetry properties of the scatter matrices $\boldsymbol{S}_B$ and $\boldsymbol{S}_W$ at the hidden layer, they can be simply computed as

$$\begin{aligned}\tilde{\boldsymbol{s}}_B &= \sum_{k=1}^{M} w_{k1}^H w_{k1}^H (\boldsymbol{S}_B)_{kk} + 2\sum_{k=1}^{M}\sum_{h=k+1}^{M} w_{k1}^H w_{h1}^H (\boldsymbol{S}_B)_{kh} \\ \tilde{\boldsymbol{s}}_W &= \sum_{k=1}^{M} w_{k1}^H w_{k1}^H (\boldsymbol{S}_W)_{kk} + 2\sum_{k=1}^{M}\sum_{h=k+1}^{M} w_{k1}^H w_{h1}^H (\boldsymbol{S}_W)_{kh}.\end{aligned}$$

The within class and between class scatter matrices of the hidden layer projections are now

$$\boldsymbol{S}_W = \sum_{i=1,2}\sum_{j=1}^{N_i} (\boldsymbol{out}_{ij} - \boldsymbol{m}_i)(\boldsymbol{out}_{ij} - \boldsymbol{m}_i)^t$$

$$\boldsymbol{S}_B = \sum_{i=1,2} N_i (\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^t$$

respectively. Here $\boldsymbol{m}_i$, is the mean of the nonlinear projections into the hidden layer of class $i$ elements: $\boldsymbol{m}_i = \frac{1}{N_i}\sum_{j=1}^{N_i} \boldsymbol{out}_{ij}$. Also, $\boldsymbol{m}$ denotes the total mean $\boldsymbol{m} = \frac{1}{N}\sum_{i=1,2}\sum_{j=1}^{N_i} \boldsymbol{out}_{ij}$ of the hidden layer projections of the original patterns. It follows that the partial of the output scatter scalars can be computed now as

$$\begin{aligned}\frac{\partial \tilde{\boldsymbol{s}}_B}{\partial out_{ij}^l} &= 2\sum_{k=1}^{M} w_{l1}^H w_{k1}^H (m_i^k - m^k) \\ &\quad -2\sum_{h=1}^{M}\sum_{k=1}^{M}\sum_{s=1,2} w_{k1}^H w_{h1}^H \frac{N_s}{N}(m_s^k - m^k) \\ \frac{\partial \tilde{\boldsymbol{s}}_W}{\partial out_{ij}^l} &= 2\sum_{k=1}^{M} w_{l1}^H w_{k1}^H (out_{ij}^k - m_i^k) \\ &\quad -\frac{2}{N_i}\sum_{h=1}^{M}\sum_{k=1}^{M}\sum_{r=1}^{N_i} w_{k1}^H w_{h1}^H (out_{ir}^k - m_i^k),\end{aligned}$$

where $m_i^k$ and $m^k$ denote the components of the class means $\boldsymbol{m}_i$ and of the total mean $\boldsymbol{m}$ respectively.

As mentioned before, a full discussion of the properties of NLDA networks can be found in [11]. We will end this section briefly mentioning NLDA complexity and its behavior with respect to local minima. The more complicated criterion function being used in NLDA results in costlier model training. Concrete complexity estimates depend on the particular criterion function being used. For a two class problem, it can be easily derived from the above estimates that the cost of a single full network gradient estimation is O($NDM^2$), where we recall that $D$ denotes input pattern dimension and $M$ the number of hidden units, and $N$ is the total number of patterns. In contrast, gradient computations in backpropagation will have a cost of O($NDM$), that is, will be $M$ times faster. In any case, when tried in problems in which both methods converge, NLDA training tends to require less iterations than backpropagation, partially alleviating its greater cost.

With respect to local minima, it must be first observed that the projecting weights in Fisher's method are not unique at all. Notice that the determinant based criterion function (1) is invariant with respect to translations, dilations and rotations of hidden layer outputs. Thus, a given minimum in NLDA training

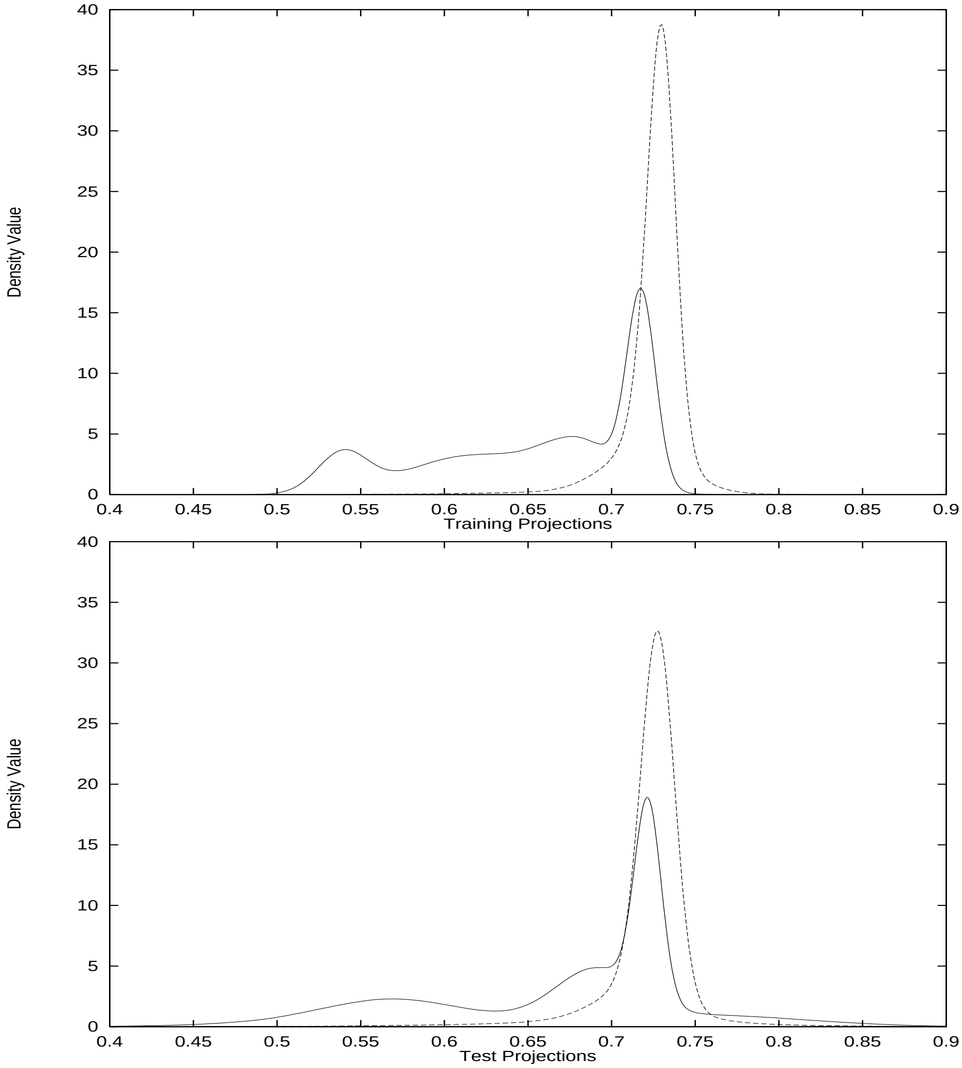

Fig. 3. Profile densities of training and test data sets. Training sets were derived from traffic from January to September 1994, while test sets correspond to traffic from October 1994 to April 1995. The tallest density corresponds to to legal operations, while the shortest, left–skewed one to fraudulent ones.

will correspond to infinitely many different weights. In any case, and as it is done with MLPs, it is the classification power of concrete NLDA networks obtained under similar convergence conditions what should be compared to have a better assessment of performance variations caused by network training. In our experience, both in credit card fraud detection and in other classification problems, different classifiers with similar training convergence behavior have very similar and very good performances. Therefore, local minima, although may be more visible in NLDA networks, have effects rather similar to those present in MLPs.

## IV. Neural fraud detectors construction and testing

In this section we will review the main issues that had to be faced in Minerva's construction. As mentioned in the introduction, the first task was the segmentation of the more than 12 million operations per year which Minerva will have to process. The main objective is to lower the number of good transactions to be considered without much reducing the fraudulent ones to be detected. Although conceivably they could also be used in this task, neural methods have not been applied. Instead, parametric segmentation criteria have been established after analyzing the most relevant traffic statistics. The resulting good–to–bad rates have gone from the tens of thousands to one to a more manageable rate of 150 to one over all the geographical subareas handled by the individual neural modules. Of course, this segmentation also means that Minerva has to let pass a number of bad operations, thus lowering the total fraud it could in principle detect. Concrete amounts vary of course depending on regional areas and activity sectors, but about 40 % of the total amount of fraudulent operations still falls under Minerva's scope.

Once data segmentation has been completed, neural model construction follows a familiar pattern. In fact, training and test sets can be chosen in a natural way: a fixed date is set, and all traffic prior to that date is segmented and used for model construction. The model is then tested with all the traffic occurring after that date. Notice that the date and time recording of actual operations makes sequential traffic emulation very easy; therefore, the model responses thus obtained give a totally realistic view of its performance.

For easier interpretation, Minerva displays not mere numerical profiles but rather what essentially amounts to the posterior probabilities of individual operations of being fraudulent. Their distributions are derived from training data sets. It is interesting to observe that, for legal traffic, the distribution of profiles in training and test sets are nearly equal; of course they vary much more for fraudulent operations. This is shown in figure 3, where actual traffic histograms have been smoothed to density mixtures of 5 gaussians for better visualization. That distribution confidence is quite relevant in the exploitation of Minerva. More precisely, Minerva's ratings could be viewed as absolute measures of the "goodness" of a given operation, and used to set up automatic acceptance or denial criteria. However, a more sensible way of exploiting them is to also consider the prior activity of the underlying card, making a final decision over all this information. This of course means the allocation of human resources for that task, which in most cases will come from the departments already performing related functions (such as the National Unified Authorization Center in the case of SEMP). The similarity of operation profilings in legal training and test traffic means that the number of referrals to be made can be adequately predicted once an exploitation strategy has been decided, and this makes resource assignments easier to decide.

Figure 3 also shows the above mentioned overlapping between legal and fraudulent traffic, as it translates to operation profiles. The figure suggests that a considerable overlapping will also occur at the input level. No direct measurement of that overlapping has been attempted (notice that input dimension is about 20),

| Training | | | |
|---|---|---|---|
| Rating | RFP | oRDT | aRDT |
| 10 | 9 | 99 | 99 |
| 20 | 6 | 93 | 95 |
| 30 | 3 | 79 | 82 |
| 40 | 2 | 74 | 77 |
| 50 | 1 | 70 | 73 |
| 60 | 1 | 64 | 67 |
| 70 | 1 | 58 | 59 |
| 80 | 0 | 54 | 55 |

| Test | | | | |
|---|---|---|---|---|
| Rating | RFP | eRFP | oRDT | aRDT |
| 10 | 19 | 11 | 84 | 73 |
| 20 | 14 | 8 | 73 | 65 |
| 30 | 9 | 5 | 52 | 48 |
| 40 | 7 | 4 | 41 | 38 |
| 50 | 5 | 3 | 36 | 33 |
| 60 | 5 | 2 | 25 | 24 |
| 70 | 4 | 2 | 18 | 17 |
| 80 | 4 | 2 | 14 | 14 |

TABLE I

Evolution with respect to ratings of performance ratios. The traffic from January 1994 to September 1994 was used for training and that from October 1994 through April 1995 for test.

but an indirect confirmation of its occurrence can be derived from the convergence behavior of ordinary perceptrons: it was observed that for various numbers of hidden layers and units, MLP convergence was rather slow and did not reach adequate error levels. Moreover, the resulting MLP performance was very poor, even upon training data sets.

As it has been mentioned, model evaluation is naturally performed over a time period immediately following the date up to which traffic was used for training. However, it is not so clear how to evaluate the actual fraud that a system such as Minerva can prevent. For instance, the money saved when a fraudulent operation has been detected is a first measure but, since that detection will also make impossible subsequent fraud attempts against that card's charge account, the card's credit limit or the money still in the account could also be sensible ways of measuring the prevention capabilities of a given system.

In Minerva's development, however, these values were not available (recall that SEMP does not have information on individual card holders). Instead, for a given probability level, model performance was measured against four values:

- RFP: ratio of false to positive warnings, that is, the quotient between the number of legal and fraudulent operations causing alarms.
- eRFP: effective ratio of false to positive warnings, that is, the quotient between the number of legal cards that generate alarms, and the number of fraudulent operations. The rationale for this is clear: during actual exploitation, a Minerva alert will most likely result in a referral. If it shows that a proper operation has been attempted, the corresponding card will temporarily pass to a "white list" of certified good cards, and won't generate more alerts on a certain time period. Thus, the eRFP ratio reflects more adequately actual performance.
- aRDT: the ratio between the amount of fraud detected and the total amount of the fraud visible to Minerva in the time period under consideration, multiplied by 100.
- oRDT: the ratio between the number of fraudulent operations detected and the total number of fraudulent operations visible to Minerva in the time period under consideration, multiplied by 100.

These values can be easily computed from traffic information. Moreover, they realistically try to capture Minerva's performance under two competing circumstances. Clearly, if high RFP and eRFP values are acceptable, oRDT and aRDT values will most likely be near 100. If, as it is more likely, RFP and eRFP have to be kept instead to relatively low levels, oRDT and aRDT will probably degrade.

An example of this behavior is captured in table I, derived from one of the geographical neural modules in Minerva. Training in that table was done on traffic from January to September 1994, and test with that going from October 1994 to April 1995. Each row corresponds to performance ratios computed over operations with ratings greater or equal to those marked at left; values have been rounded to the nearest integer. Notice how RDT test values are quite good for ratings above 10 and even 20; they decay afterwards, but still manage to be above a fairly good 50 per cent for eRFP test values of 5. Observe also that there is a correlation of ratings and training RFP values, due to the above mentioned fact that Minerva ratings essentially correspond to posterior probabilities of training data. Of course, fraud data in test sets is likely to be different from that in training; this is shown by the loss of that correlation in RFP test values.

## V. Conclusions

The Minerva fraud detection system for credit card operation was installed in SEMP's transactional systems in July 1996 and has been run in a continuous fashion since mid August, providing operation ratings against which actual queries are being made. It is by now totally integrated in SEMP's operation. The average rating time is about 60 milliseconds, largely dominated by disk accesses, and its detecting performance is so far are very positive, both with respect of human intervention those queries are needing and with the amounts saved through fraud prevention.

Systems such as Minerva show that neural, operation based, real time fraud detection systems are not only technically feasible, but highly interesting from a purely economical point of view. However, their development has to overcome certain hurdles. First, rather extensive data analysis has to be performed on traffic information to obtain a meaningful set of detection variables, and also to effectively segment that data in such a way that enormous imbalances between legal and fraudulent traffic do not overwhelm the latter to the point of making detection impossible.

Moreover, difficult prior probabilities estimation and class overlapping may make ordinary MLP training essentially useless; it is thus necessary to devise new model building approaches. The one adopted in Minerva's development, non linear discriminant analysis (NLDA), was primarily designed to avoid difficult to implement target codings. It has also resulted in considerable network robustness with respect to class overlapping. This is a property that NLDA may share with other network training procedures that do not rely on target codings such as, for instance, support vector machines [3].

It is certainly possible to use the ratings of a system such as Minerva as the basis for absolute, totally automated, operation acceptance or rejection criteria. However, an alternate, more realistic approach in by–operation systems is to use those ratings jointly with a card's immediate operation history to make referral decisions. This may certainly be impossible if the number of warnings is excessively high. In Minerva's case, the traffic segmentation being performed, while letting pass some bad operations, puts the requirements of its referral handling well under reach of SEMP's Authorization Center. The final system can still attack a significant portion of attempted fraud while making very good sense from a cost–effectiveness point of view.

## References

[1] "Canada Trust inks Falcon pact with HNC", AI Expert September 1994; "Nestor inks pact with financial firms", AI Expert, December 1993.

[2] A. Classe, "Caught in the neural net (Credit card fraud detection)", Accountancy **115** (1995), 58–59.

[3] C. Cortes and V. Vapnik, "Support vector networks", Machine Learning **20** (1995), 273–297.

[4] R.O. Duda, P.E. Hart, "Pattern classification and scene analysis", Wiley, 1973.

[5] K. Fukunaga, "Introduction to Statistical Pattern Recognition", Academic Press, 1972.

[6] P. Gallinari, S. Thiria, F. Badran, F. Fogelman–Soulie, "On the relations between discriminant analysis and multilayer perceptrons", Neural Networks **4** (1990), 349–360.

[7] S. Ghosh, D.L. Reilly, "Credit card fraud detection with a neural network", Proceedings of the 27th Hawaii International Conference on System Sciences, IEEE Computer Society Press, 1994, 621–630.

[8] K. Hornik, M. Stinchcombe, H. White, "Universal Approximation of an Unknown Mapping and Its Derivatives Using Multilayered Feedforward Networks", Neural Networks **3** (1990), 551–560.

[9] "Visa cracks down on fraud: credit card ID systems get added muscle", Information Week, August 1996.

[10] W. H. Press, B. P. Flannery, S. A. Teukolski, W. T. Vetterling, "Numerical Recipes in C", Cambridge U. Press, 1992.

[11] C. Santa Cruz, J. Dorronsoro, "A nonlinear discriminant algorithm for data projection and feature extraction", IIC Technical Report 2/96 (submitted to IEEE Trans. in Neural Networks).

[12] A.R. Webb, D. Lowe, “The optimized internal representation of multilayer classifier networks performs nonlinear discriminant analysis”, Neural Networks **3** (1990), 367–375.

**José R. Dorronsoro** received the Licenciado en Matemáticas degree from the Universidad Complutense de Madrid in 1977 and the Ph.D. degree in Mathematics from Washington University in St. Louis in 1981. Currently he teaches in the Computer Engineering Department of the Universidad Autónoma de Madrid, of which he was head from 1993 to 1996, and currently heads the Quantitative Methods Group of the Instituto de Ingeniería del Conocimiento, a Research and Development center of that university. His research interest are in Neural Networks and Pattern Recognition.

**Francisco Ginel** is currently working at Visa International OCW in San Francisco Bay area as a Director for International Risk Management. He was previously working at SEMP in Madrid as head of the Risk Management and Security Department and as an Analyst for Organization and Delivery Systems Departments.

**Carmen Sánchez** is Head of the Risk Management Department at SEMP in Madrid. Previously she was in charge of the full Visa product range also at SEMP, where she has been working since 1991.

**Carlos Santa Cruz** received the degree physics from the Universidad Autónoma of Madrid (UAM) in 1987. He joined Hamburg University (Germany) in 1989 as a post-graduate student. He received the Ph.D. in physics from the UAM in 1991. Currently he is a professor at the Computer Science Department of the UAM and member of the Quantitative Methods Group of the Instituto de Ingeniería del Conocimiento. His research interest are in Pattern Recognition, Model Building and Time Series Forecasting using Neural Networks.