COMPARING THE PERFORMANCE OF SOME NEURAL FRAUD DETECTORS IN TELECOMMUNICATIONS

M.R. Arahal*, M. Berenguel†, F. Pavón*, E.F. Camacho*

* Universidad de Sevilla. Depto. Ingeniería de Sistemas y Automática. Escuela Superior de Ingenieros. Camino de los Descubrimientos s/n, E 41092, Sevilla, Spain, arahal@esi.us.es

† Universidad de Almería. Depto. de Lenguajes y Computación. Área de Ingeniería de Sistemas y Automática. Ctra. Sacramento s/n, La Cañada, E 04120, Almería, Spain, beren@ual.es

Keywords: neural networks, fraud detection, pattern recognition, telecommunications.

Abstract

Fraud detection in telecommunications is an area where pattern recognition and so called ”intelligent” techniques have found widespread use. Due to fraud, companies suffer not only direct economic losses but also the risk of bad publicity. In this paper real cases of fraud are being treated in order to develop a detection system with low number of false alarms and good sensitivity. Call data records provide a number of measures that can be used to discriminate fraudulent activities from correct ones. Three neural networks schemes have been applied to such data comparing latter their results with new cases.

1 Introduction

Telecommunication companies are often faced with fraudulent activities on the part of some clients that use their services to yield an illegal profit. Automatic fraud detection is based on data from call records handled by pattern recognition techniques. A survey of data mining techniques can be found in [3].

An automatic fraud detection system (AFDS) should maximize the number of fraudulent cases detected and minimize the number of false alarms. The cost associated with non detected fraud and with false alarms should be known in order to optimize the performance of the AFDS as is proposed in [2]. In this particular case no quantitative measure has been given about these costs, the only indication being that false alarms should be kept as low as possible with a ratio below five percent.

Fraud amounts to a very small percentage of the total number of cases. For this reason segmentation of data is often a prerequisite in order to overcome the imbalance between legal and fraudulent traffic (see [1] for a longer discussion). In our case, data comes after a previous screening performed by the company. This filtering eliminates most obvious cases of non fraud activity.

Commercial packages such as HNC Fraud Manager (from HNC software) cover different kinds of fraud such as Subscription Fraud, Internal Fraud and Dealer/Agent Fraud. In this paper the type of fraud considered is Call Sell Operation (CSO). This amount to the illegal re-selling of telephony services. The only data used is obtained from Call Data Records (CDR), that means that no use is made of the private customer data.

In the real application considered in this paper, some data mining neural techniques have been tested in order to develop a detection system with low number of false alarms and good sensitivity. For each technique a number of models have been developed and tested with data spared for this purpose. The models are later compared using new data not previously available. The comparison points out that model selection is far from being straightforward. Good generalization is hard to achieve even with the aid of a large data base.

The fraud problem is presented in the next section together with the available data. The different neural techniques are shown in section 3, followed by the selection of neural fraud detectors. The comparison of the models at work with fresh data is presented in section 4. The conclusions derived from this work end the paper.

2 Fraud detection problem

The use of phone services generates large amount of data with information about calls in basic telephony and traffic in modern information networks. Millions of records are created each day containing the activities made by the users. Most of these records correspond to non-fraudulent activities. Due to this, the fraud detection task has to deal with two basic characteristics: large amount of information and a small ratio of fraudulent activities.

Fraud detection systems (FDS) use well defined criteria for analyzing all information coming out of Call Data Records (CDR) together with information about the user ([7]). Human analysts define a set of rules for FDS. As a result, the performance of the FDS is largely determined by subjective criteria and too depending on human knowledge.

Automatic detection systems try to discover relationships among data and use them to classify new cases. Many variables are usually at the disposal of such systems. In this paper the type of fraud considered is Call Sell Operation, this is the illegal re-selling of telephony services. The only data used is obtained from Call data records, that means that no use is made of the private customer data.

The data base used for the tests consists of thousands of records previously screened. Each record contains the following fields:

- Phone number
- Total number of calls
- Total cost of the calls
- Cellular/conventional phone
- Technique used in screening
- Cause that motivated the selection of the record
- Severity of the case
- Length of the temporal window used in the screening of the record

These fields are numerically modified in a certain way to prevent sensible information from spreading. This should not affect the automatic detector in any significant way since they look after patterns and not for particular values of the variables. Of all records used, just about a 20 % correspond to fraudulent activities. This percentage is obviously larger than the normal rate of fraud in non-screened data. The used non-fraud records are still sufficiently informative and numerous.

The above listed fields can be accommodated in a vector $\mathbf{x}$ whose components are all normalized to have zero mean and variance unity as is usually done in the artificial neural networks realm. Each record in the database contains also an indicator y that can take two values: "1" if the call was used for fraudulent purposes or "0" otherwise.

From the neural perspective the data base is seen as a collection of pairs ($\mathbf{x}_i$, y_i) with $i = 1, 2, \cdots N$, being N the number of available (labelled) records . A neural network can be trained using some input/output pairs to produce at its output correct values of y when an instance of vector $\mathbf{x}$ is presented at its input.

A fraud detection system has to use the labelled data to provide a classification of new cases. The possible outcomes are summarized in the contingency table 1. As can be seen, a case of fraud is called a "positive". A correctly classified positive is so called a "true positive" (TP). A case of fraud that is not signaled by the detector is a "false negative" (FN). Similarly, a case of legal activity is denominated a "negative", if the detector classifies it as legal the result is labelled as "true negative" (TN). Finally, a "false positive" (FP) or false alarm is a case of legal operation misclassified as fraud.

Classification	Case	Denomination
Fraud	Fraud	TP
Legal	Legal	TN
Fraud	Legal	FP
Legal	Fraud	FN

Table 1. Contingency table.

The goodness of a fraud detector can be assessed in terms of sensitivity and susceptibility. The first property refers to the capacity of the detector to signal true cases of fraud, the second is the tendency to produce false alarms. These properties can be better defined as functions of the number of "true" and "false" positives and negatives as shown in Table 2.

Indicator	Symbol	Expression
sensitivity	TPR	$100\frac{TP}{TP+FN}$
susceptibility	FPR	$100\frac{FP}{FP+TN}$

Table 2. Definition of sensitivity and susceptibility.

It can be seen that the TPR or True Positives Ratio is obtained as the percentage of true positives over the total number of positives ($TP + FN$). In the same way, the FPR of False Positives Ratio is obtained as the percentage of false alarms over the total number of legal cases ($FP + TN$).

The available data has been split in three sets. A training set (TS) that will be used for adjusting the models parameters, usually trough a gradient based adjusting algorithm such as backpropagation or other similar. A second set denominated validating set (VS) will be used to compare the performance of different networks (networks with different parameters or different structures) and select the one that provides the best results without overfitting the data. Finally a third set (MVS) will be used to compare different model building techniques. All the three sets are composed of past data. However, the resulting models will be compared in the more realistic scenario composed of new data that will be designated as (NS).

Of all historical data, TS contains 29 % of the records, VS contains 43 % and the rest (28 %) forms the MVS. The results will be given based on NS which is as large as VS.

The size of the stored data base is $N = 10000$ labelled cases. The NS consists of 5000 new cases.

3 Neural detectors

Neural networks are nowadays usual instruments for approximating nonlinear mappings from examples. In this particular application they will be used to classify cases of phone services based on the features included in vector $\mathbf{x}$ that will be used as the input vector of the networks.

After adequate training, the output of the network NN($\mathbf{x}$) will be considered to signal a case of fraud when it is larger than 0.5, else the case will be classified as a legal activity one.

On top of the requirements of high sensitivity and low susceptibility, the specifications for an automatic fraud detection system should include the ability to treat large amounts of data and ease for incorporating new labelled cases. Radial basis functions (RBF) networks with local basis are appalling to this task since there are methods for constructing and adapting them at

the same rate of arrival of new information.

On the other hand, multilayer perceptrons are considered to achieve more flexibility with fewer nodes, specially in high dimensional spaces, where RBF networks are victims of the curse of dimensionality.

In the following some tailored techniques for constructing neural fraud detectors will be presented.

3.1 Homogeneous RBF networks

The first neural structure used in this paper corresponds to the RBF type with linear output node (see Figure 1). The output of the network is the weighted sum of the outputs of all nodes:

$$\mathrm{NN}(\mathbf{x}) = \sum_{n=1}^{n=nn} w_n e^{-d(\mathbf{x})^2/\sigma_n^2} \qquad (1)$$

being w_n the output weight for the n-th node, $d(\mathbf{x})$ the distance from input vector $\mathbf{x}$ to the n-th basis function calculated as $d(\mathbf{x})^2 = \|\mathbf{x} - \mathbf{c}_n\|^2$, nn the number of nodes of the network and $\mathbf{c}_n$ the center of the n-th node.

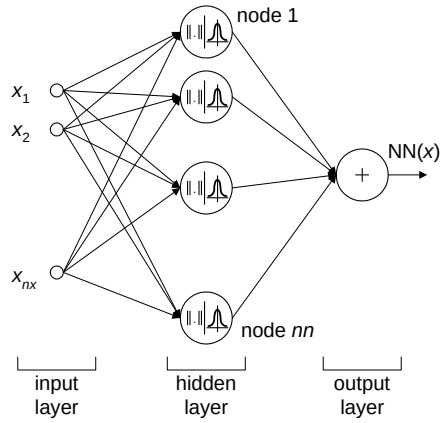

Figure 1. A radial basis function network with linear output node.

The number of nodes nn, the centers $\mathbf{c}_n$, the widths σ_n and output weights w_n can be selected in different ways. Usually some parameters (such as nn, σ_n, etc.) are selected *a priori* leaving to the training procedure the task of adjusting the rest of the parameters in such a way that the output of the network takes appropriate values in a controlled set of data.

For the sake of comparison with more complex models we begin with a RBF network with nodes of equal widths disposed at the locations of high density of fraud.

The placement of centers can be accomplished in this way: whenever a labelled case of fraud is misclassified a new basis (node) should be placed taking the current input vector $\mathbf{x}$ as center. Please note that not all cases of fraud in the database receive a node, just those that happen to lie far away from other cases in the input space.

The output weights of the nodes are obtained through a simple gradient-based algorithm using the TS (see for instance [4]). The VS is used to stop training before overfitting occurs.

With these simplifications the only design parameter that has to be chosen is the width of the basis (σ). We will later see how the choice of σ affects the performance in the MVS. This algorithm is quite simple to program, test and implement and is able to cope with new cases of fraud adding new nodes when needed. In fact, this approach achieves a solution many times faster than backpropagation does with feedforward networks of various hidden layers.

Typical problems of this approach are that the placement of centers can allocate far more resources than needed in regions having dense data and small variations in the output. This first problem can be alleviated using an extended input-output metric for the center-allocating phase (see [5]), and allowing the centers to be moved in the training phase as done in [6]. However this is not an issue in our present application since the number of fraud cases are small.

After training it was found that the performance of the algorithm in the TS is quite good, achieving almost 100 % sensitivity with less than 10 % of false alarms. This is somehow expectable of a clean data base and does not relate well to the performance in the real use of the fraud detector. Table 3 shows the results obtained applying the algorithm to the same data set used to adjust the w_n parameters, that is, to TS. It can be seen that for small values of σ the sensitivity is low, larger values provide better sensitivity at the cost of an increase in the number of false alarms.

σ	FPR	TPR
0.010	0	89.0
0.015	0.9	98.5
0.020	4.5	99.4
0.030	6.2	99.6
0.060	8.6	100
0.100	11	100

Table 3. Results obtained in the TS by the homogeneous RBF construction algorithm.

The MVS data is now used to assess the effectiveness of this technique. In Table 4 the susceptibility and sensitivity are shown for the same values of the parameter σ. As expected the results are not as good as with the TS. Again, for large values of σ the TPR goes up at the expense of worse FPR.

σ	FPR	TPR
0.010	0	57.0
0.015	0.9	59.3
0.020	4.5	63.1
0.030	6.2	76.0
0.060	8.6	80.1
0.100	11	89.9

Table 4. Results obtained in the MVS by the homogeneous RBF construction algorithm.

A graph can help analysing the results. Figure 2 shows the susceptibility or False Positive Ratio FPR in the horizontal axis and the sensitivity of True Positive Ratio TPR in the vertical axis. The results correspond to homogeneous widths RBF networks for different choices of σ. This plot (known as Lorenzt diagram) allows to decide which parameter of a classifier yields better results once an optimization criterion has been stated (in terms of acceptable FPR and desired TPR). The upper curve (circles) shows the results for the TS. Each circle correspond to a different value of the parameter (σ in this case). The values are the same listed in Table 3. The lower curve (x-marks) shows the results for the MVS. The values of σ are the same than in the TS curve.

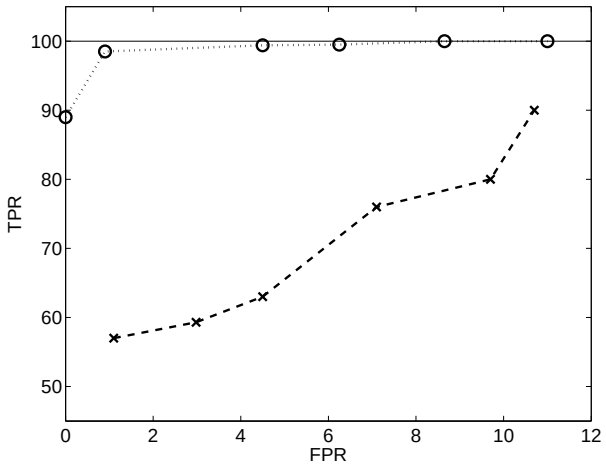

Figure 2. Lorenzt diagram obtained for homogeneous RBF networks on the MVS data (x-marks) and the TS data (circles). Just three values of the varying parameter σ are given in the graph. See text for more details.

This graph tells us that if a 5 % is the upper limit tolerable for false alarms, then values of $\sigma > 0.02$ should not be used. Also it warns us against making optimistic claims about the performance of the classifier based only in the data used for training. The performance in data not used for training is lower and this is an indication of the generalization capacity of the classifier.

3.2 Variable width RBF networks

A second and more elaborate algorithm for constructing RBF networks makes use of the idea of the nearest neighbor of a basis function. Again the nodes are located at points of the input space that represent fraudulent cases. For each basis the width is selected so as to cause minimum disturbance on the closest case of non-fraud activity. A reduction of the number of false alarms is seek in this way.

To make this clearer suppose that a basis is to be situated at location $\mathbf{x}_c$ corresponding to a labelled case of fraud that is currently misclassified. Suppose also that the closest labelled record in the input space using Euclidean metric that correspond to non-fraud activity is $\mathbf{x}_v$. The value that the neural network should provide for $\mathbf{x}_v$ is zero. However, if a basis with $w = 1$ is placed at $\mathbf{x}_c$ it would have the effect of providing a value for $\mathbf{x}$=$\mathbf{x}_v$ of $\kappa = e^{-\|\mathbf{x}_c - \mathbf{x}_v\|^2/\sigma^2}$

The value κ is designated as overlap. An overlap less than 0.5 could be considered tolerable since only values greater than 0.5 are considered to signal positives. It has to be noted however that the basis at $\mathbf{x}_c$ might not be the only one in the vicinity of the negative at $\mathbf{x}_v$. This causes a problem since the output of the network is the sum of the output of all nodes. A correct value for κ has to be selected.

It has to be noted that once κ has been selected, the width of each base is determined. Using this approach the only parameter to be chosen is the amount of overlap that is permitted.

This algorithm is also easy to implement and the parameter κ can be tuned using the Lorenzt diagram as in the previous case. The results obtained for the TS and MVS are shown in Table 5.

	TS			MVS	
κ	FPR	TPR	κ	FPR	TPR
0.001	0	96.4	0.001	0	53.0
0.005	1.1	99.3	0.005	1.2	64.2
0.010	2.0	99.8	0.010	4.9	67.1
0.020	4.3	100	0.020	7.2	81.8
0.050	7.1	100	0.050	9.8	83.8
0.100	10.8	100	0.100	10.5	94

Table 5. Results obtained in the TS (left) and in the MVS (right) by the variable width RBF construction algorithm.

It can be seen in Figure 3 that the performance of the algorithm in the TS (circles) shows high sensitivity but the ratio of false alarms is considerably reduced with respect to the previous technique. In the MVS (x-marks) the results are again poorer than in the TS but some improvement has been made over the first technique. The values of the varying parameter κ are the same as in Table 5.

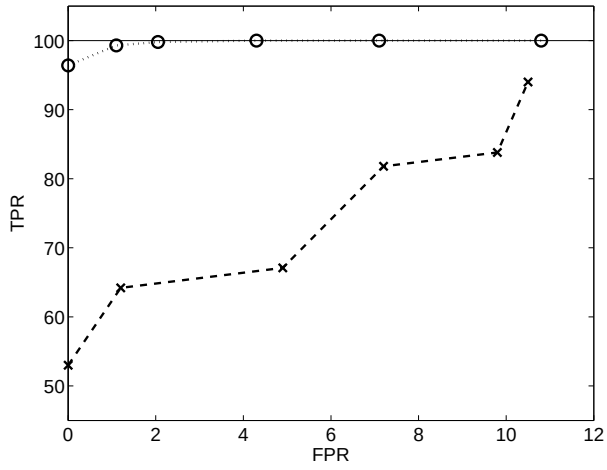

Figure 3. Lorenzt diagram obtained for variable width RBF networks on the MVS data (x-marks) and the TS data (circles).

3.3 Multilayer perceptrons

The last technique is the well known multilayer perceptron (see Figure 4). In this paper we have restricted ourselves to the use of one-hidden layer perceptrons with sigmoidal activation function in the hidden layer and a single linear output node.

The network output is calculated as a function of its input vector as:

$$\mathrm{NN}(\mathbf{x}) = \sum_{n=1}^{n=nn} w_n^o s_n(\mathbf{x}) + b^o \qquad (2)$$

being nn the number of nodes in the hidden layer and $s_n(\mathbf{x})$ the output of the $n-$th node, obtained as:

$$s_n(\mathbf{x}) = \sum_{k=1}^{k=n_x} w_k^i \mathbf{x}_k + b^i \qquad (3)$$

where n_x is the dimension of the input vector $\mathbf{x}$. Coefficients w_n^o, b^o, w_k^i and b^i are the adjustable parameters of the network and are referred to as weights. Training is the procedure (gradient based in most cases) for assigning a value to weights so that the approximation error is made small. The VS is used to control the number of training cycles in order to avoid overtraining.

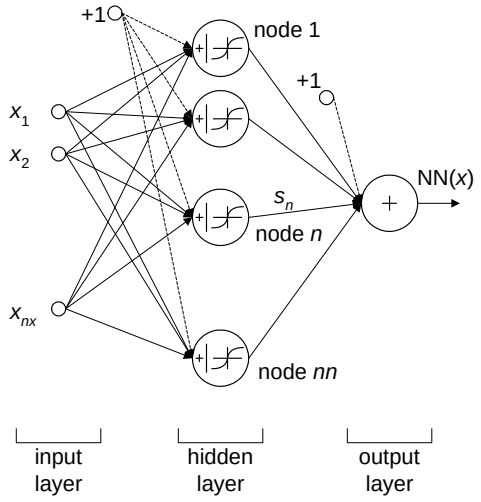

Figure 4. A perceptron with a single hidden layer and a linear output node.

For this restricted class of perceptrons the only parameter to be chosen is the number of nodes nn. As in the previous cases, the performance will be shown using data from MVS for different values of nn.

Results in the TS and VS are a little poorer than those of the RBF networks, but contrary to those they do not degrade much when the MVS is used. However the results happen to be very much dependent on the number of training cycles and the initial value given to weights before training. For instance, in Figure 5 the results obtained with ten networks of $nn = 5$ are shown. The networks are different because they have been obtained using a training algorithm that provides random initial values to weights.

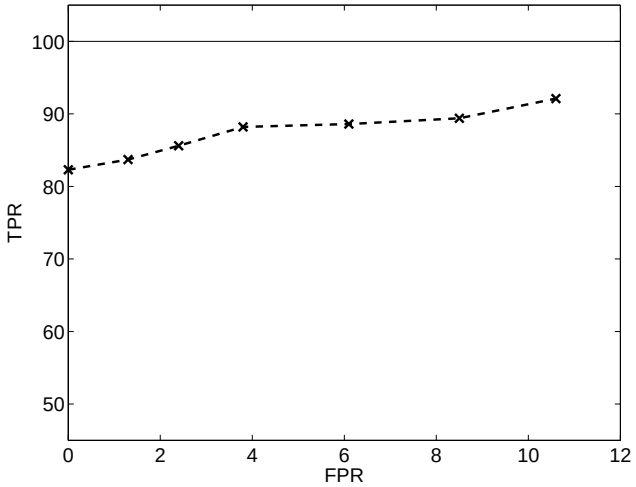

Figure 5. Lorenzt diagram obtained for a perceptron with one hidden layer of five nodes using the MVS data.

The number of nodes also plays a role in this classifier. Figure 6 shows different curves each one obtained in a similar way as the one in Figure 5 but using different values for nn. Each line joins result points obtained with different neural networks with the same number of nodes, that is, networks with different number of training iterations and different initial value of weights but with the same number of nodes.

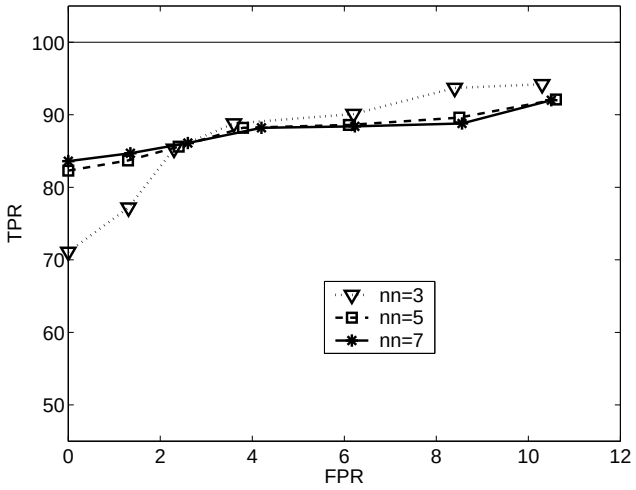

Figure 6. Curves for networks of different number of nodes using the MVS data.

It appears that small networks ($nn < 4$) have less sensitivity at smaller susceptibility but perform better than larger networks if ratios of false alarms over 5 % were allowed. Also, since curves for $nn = 5$ and higher looks very similar, it seems that there is little difference in performance for different number of nodes once a certain degree of flexibility has been reached.

4 Results

According to the results obtained with the controlled sets of data TS, VS and MVS a single model has been selected from each of the three classes. All three models have a FPR less than 4 % and TPR greater than 60 %.

The most realistic test for a classifier is when it is faced with totally new data. This has been carried out using data from the NS. These labelled cases have been obtained after prescreening obvious cases of legal activities, so that the fraudulent cases in this new set of data are about 30 %.

Table 6 shows the TPR and FPR obtained for the three models. It can be seen that the homogeneous width RBF network did perform acceptably well despite its simplicity. Better results are given by the variable width RBF network, that retains in this test the low value of false alarms although the sensitivity has lowered a bit. Finally, the perceptron detector is the best combining the higher sensitivity with a medium and very acceptable false alarm level.

Model	FPR	TPR
Homogeneous width RBF network with $\sigma = 0.02$	6.25 %	74.4 %
Varying width RBF network with $\kappa = 0.01$	2.23 %	71.2 %
One hidden-layer perceptron with seven hidden nodes	3.20 %	77.0 %

Table 6. Results of the three selected models using fresh data (NS).

It can be seen that all three models have shown a worse performance with the new data than with the historical one used (in one way or another) to generate them. This is the classical problem of generalization.

5 Conclusions

The comparison of three models have shown that one hidden layer perceptrons perform better than radial basis function networks constructed with the algorithms presented in this particular problem. However, RBF are easy to adapt to new cases of fraud due to the local nature of the approximation they offer. For instance, removing or adding a new basis produces changes in the overall classification just in cases whose vector $\mathbf{x}$ lies close enough to the altered basis. The classification is thus changed only locally and the extension of the changes can be controlled. This is a great advantage when one is faced with patterns changing over time as can be the case with fraud.

On the other hand, a perceptron has not easy incremental learning capabilities. Retraining affects not only new patterns but also every other pattern already learned due to the global nature of the nonlinear activation functions used by its nodes.

Changing the metric of the basis has been proposed (see [5]) as a means to give more flexibility to RBF networks and attaining better results. This however has the problem that the construction of the network becomes as involved as a perceptron, being then no difference between both approaches.

The comparison also points out that model selection is far from being straightforward. First of all, performance in the labelled data set used for training is not indicative of final performance. The use of a validation set to avoid overfitting and a model validation set to select the most appropriate parameters and/or structures provides a simple way for dealing with the problem of generalization, however many cases have to be spared for this purpose. Finally, the gap between performance in the cases contained in the database and in the NS shows that good generalization is hard to achieve even with the aid of a large database.

References

[1] Dorronsoro, J., F. Ginel, C. Snchez and C. Santa Cruz (1997). Neural fraud detection in credit card operations. IEEE Trans. Pleural Networks 8, 827-834.

[2] Fawcett, Tom and Foster Provost (1997). Adaptive fraud detection. Data Mining and Knowledge Discoverij 1(3), 291-316.

[3] Mitra, Sushmita, Sankar K. Pal and Pabitra Mitra (2002). Data mining in soft computing framework: A survey. IEEE Trans. Neural Networks 1, 3-14.

[4] Moody, J. and C. Darken (1989). Fast learning in networks of locally-tuned processing units. Neural Computation 1, 281-294.

[5] Saha, A. and J.D. Keeler (1990). Algorithms for better representation and faster learning in radial basis function networks. In: Advances in Neural Information Processing Systems (D. S. Touretzky, Ed.). Vol. 2. Morgan Kaufmann, San Mateo. Denver 1989. pp. 482-489.

[6] Wettschereck, D. and T. Dietterich (1992). Improving the performance of radial basis function networks by learning center locations. In: Advances in Neural Information Processing Systems (D. S. Touretzky, Ed.). Vol. 4. Morgan Kaufmann, San Mateo. pp. 1133-1140.

[7] Yuhas, B. (1993). Toll-fraud detection. In: Proceedings of the International Workshop on Applications of Neural Networks to Telecommunications, 239–244.