01 Jan 2001

Using a Neuro-Fuzzy-Genetic Data Mining Architecture to Determine a Marketing Strategy in a Charitable Organization's Donor Database

Korakot Hemsathapat

Cihan H. Dagli
Missouri University of Science and Technology, dagli@mst.edu

David Lee Enke
Missouri University of Science and Technology, enke@mst.edu

Recommended Citation

K. Hemsathapat et al., "Using a Neuro-Fuzzy-Genetic Data Mining Architecture to Determine a Marketing Strategy in a Charitable Organization's Donor Database," *Proceedings of the Change Management and the New Industrial Revolution (2001, Albany, NY)*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2001.
The definitive version is available at https://doi.org/10.1109/IEMC.2001.960482

USING A NEURO-FUZZY-GENETIC DATA MINING ARCHITECTURE TO DETERMINE A MARKETING STRATEGY IN A CHARITABLE ORGANIZATION'S DONOR DATABASE

Korakot Hemsathapat, Cihan H. Dagli, and David Enke*
Smart Engineering Systems Laboratory
236 Engineering Management Building
University of Missouri – Rolla
Rolla, MO 65409-0370
Email: korakot@umr.edu, dagli@umr.edu, enke@umr.edu*

Abstract

This paper describes the use of a neuro-fuzzy-genetic data mining architecture for finding hidden knowledge and modeling the data of the 1997 donation campaign of an American charitable organization. This data was used during the 1998 KDD Cup competition. In the architecture, all input variables are first preprocessed and all continuous variables are fuzzified. Principal Component Analysis (PCA) is then applied to reduce the dimensions of the input variables in finding combinations of variables, or factors, that describe major trends in the data. The reduced dimensions of the input variables are then used to train probabilistic neural networks (PNN) to classify the dataset according to the groups considered. A rule extraction technique is then applied in order to extract hidden knowledge from the trained neural networks and represent the knowledge in the form of crisp and fuzzy If-Then-rules. In the final stage a genetic algorithm is used as a rule-pruning module to eliminate weak rules that are still in the rule base while insuring that the classification accuracy of the rule base is improved or not changed. The pruned rule base helps the charitable organization to maximize the donation and to understand the characteristics of the respondents of the direct mail fund raising campaign.

1. Introduction

The low response rate of a direct mail fund raising campaign is a typical problem. In one study [1], gifts were used as incentives to increase the response rate of the public. This is the marketing strategy implemented by an American charity in the June 1997 donation campaign. The mailing included a gift comprised of personalized address labels with 10 note cards and envelopes. Each mailing cost the charity 0.68 dollars and resulted in a response rate of about 5% from the lapsed donors who made their last donation more than a year before the June 1997 donation campaign. Because the donations received from the respondents varied between 0.32 and 500 dollars, and the mean of donations was about 15 dollars, the charity needs to decide when it is worth sending the donation mail to a donor based on the information from the database. Furthermore, the charity is interested in finding a marketing strategy to regain lapsed donors. Hence, building a classification model to classify the lapsed donor database into two groups, respondents and non-respondents, can be used to understand the characteristics of the lapsed donors in each group.

Some analysts are reluctant to use neural networks in a direct marketing campaign because they usually want to understand the characteristics of the respondent group, which is targeted for mailing. Section 2 introduces the neuro-fuzzy-genetic data mining architecture, which can be used to classify the data, extract the knowledge or the characteristics of the lapsed donor in each group from the trained neural networks, and represent the knowledge in the form of crisp and fuzzy *If-Then* rules. Section 3 describes the steps of preprocessing and cleaning of the database. The architecture used this database to generate the model. The implementation of the architecture and the results are explained in Section 4. Section 5 is the conclusion.

2. Neuro-Fuzzy-Genetic Data Mining Architecture

Data mining is an iterative process involving several stages, beginning with domain understanding and ending with reporting and using of the discovered knowledge [2,3]. The neuro-fuzzy-genetic data mining architecture is also processed in an iterative manner as shown in Figure 1. The outputs of some modules are used as feedback to the previous modules to improve the outputs of the subsequent modules. Knowledge in a trained neural network is encoded in its connection weights, which are distributed throughout the entire neural network structure. In Figure 1, all of the inputs are first preprocessed and cleaned. The missing values of the continuous variables are replaced with the mean of

0-7803-7260-3/01/$10.00©2001 IEEE

the considered variables. The missing values of the discrete variable are replaced with the mode of the considered variables. The discrete input variables are coded by using one-of-m coding [12].

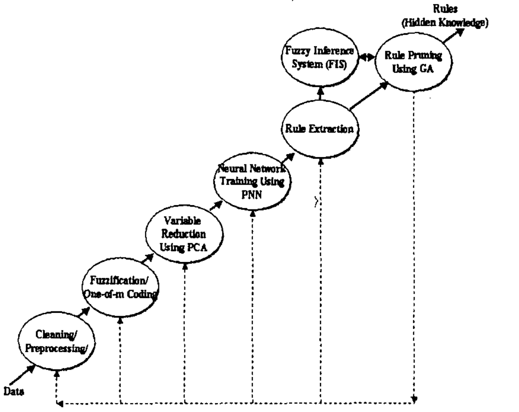
Rules
(Hidden Knowledge)
Fuzzy Inference System (FIS)
Rule Pruning Using GA
Rule Extraction
Neural Network Training Using PNN
Variable Reduction Using PCA
Fuzzification/ One-of-m Coding
Cleaning/ Preprocessing/
Data

Figure 1. The Neuro-Fuzzy-Genetic Data Mining Architecture

The three triangular membership functions (Low, Average, High) for fuzzification are designed based on the mean and standard deviation of each continuous variable, as shown in Figure 2.

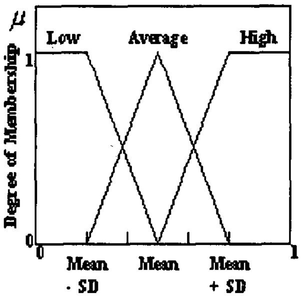

Figure 2. Triangular Membership Functions

The fuzzification and one-of-m coding increase the number of variables inputted to the next module. The increased number of inputs will result in increased training time of the network. Principal Component Analysis (PCA) is then applied for variable reduction. Mathematically, PCA relies upon an eigenvector decomposition of the covariance or correlation matrix of the variables. The objective of a PCA is to transform correlated random variables to an orthogonal set, which produces the original variance/covariance structure. In this paper the minimum fraction variance component to the database was set to 0.01. The details of the PCA are given in [4].

The neural network model used was a Probabilistic Neural Network (PNN). It has the capability to quickly classify a categorized dataset. The detail of the network structure are given in [5,6]. A PNN is first trained on a set of training data until it reaches the desired accuracy level. Then the rule extraction technique is applied to extract hidden knowledge from the trained neural network and represents it in the from of crisp and fuzzy *If-Then* rules:

If X_1 *is* Y_1, *and* X_2 *is* Y_2,.., *and* X_n *is* Y_n *then* C, WT; (1)

where X_i represents an input variable, Y_i represents a fuzzy membership function derived from X_i or one of the categories in one-of-m codes, C represents classes, and WT represents a weight of the rule. The rule extraction module extracts crisp and fuzzy *If-Then* rules from the weights of the trained neural network. Different techniques for rule extraction from neural networks can be used [7,8,9].

The Fuzzy Inferences System (FIS) was developed to test the rule base performance. It evaluates fuzzy membership values for each input variable, fires the rules sequentially to calculate the degree of membership for each species type, and declares the highest membership value rule the winner. The extracted rules that are evaluated by the FIS are not optimized. There are still some rules deemed not important to the rule base. The genetic algorithm for rule pruning is then applied to find optimal sets of rules in the rule base while the classification accuracy of the pruned rule base is still maintained or improved. The genetic algorithm rule pruning is explained later and was modified from [10].

2.1. Rule Extraction Technique. The rule extraction technique used in this paper has been modified from [9]. It calculates an effect of each fuzzified input neuron on each output by the multiplication of weight matrices. For a network (see Figure 3) with $(I_1, I_2,..., I_i)$ fuzzified input neurons, $(P_1, P_2,... ,P_j)$ neurons in the PCA layer, $(H_1, H_2,... ,H_k)$ hidden layer neurons, and $(O_1, O_2,..,O_l)$ output neurons, the technique can be described with the following steps:

Step 1: Calculate the effect measure matrix.

- Let P equal an $i \times j$ dimensional matrix, where P_{ij} is the weight connection between the i^{th} fuzzified input neuron and the j^{th} PCA layer neuron.
- Let W equal a $j \times k$ dimensional matrix, where W_{jk} is the weight connection between the j^{th} PCA layer neuron and the k^{th} hidden layer neuron.

- Let V equal a $k \times l$ dimensional matrix, where V_{jk} is the weight connection between the k^{th} hidden layer neuron and the l^{th} output layer neuron.

The $i \times l$ dimensional effect measure matrix, A, is given by

$$A = P \times W \times V = \begin{bmatrix} e_{11} & \cdot & \cdot & e_{1l} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ e_{i1} & \cdot & \cdot & e_{il} \end{bmatrix}$$

where e_{il} is the effect measure of fuzzified input neuron I_i on output neuron O_l.

Step 2: Extract the rules.
For each $e_{il} > 0$, write a rule of the form:

If x is X then y is C,

where x is X and y is C are the descriptions for the fuzzified input neuron I_i and output neuron O_l, respectively.

Step 3: Calculate the rule weighting for each rule.
For each $e_{il} > 0$, the weighting for the rule "if x is X then Y is C" is given by

$$Weight = \frac{e_{il}}{\max\{e_{11}, \ldots\ldots e_{il}\}} \times 100$$

Rules high and low in weight (*Weight*) are called strong and weak rules, respectively. This technique was previously implemented without a PCA layer and provided good results [9, 11]. However, this is an attempt to modify a rule extraction technique by adding the PCA layer to reduce the dimension of the fuzzified input layer, which results in a decreased training time for the neural network.

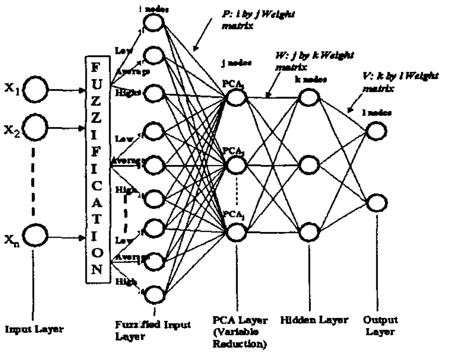

Figure 3. The Structure of the Architecture

2.2. Rule Pruning Using Genetic Algorithm. The objectives of the genetic algorithm are to find a small number of significant rules from a large number of extracted rules in the previous stage and to maximize the number of correctly classified data by the selected rules. In order to apply a genetic algorithm to the rule pruning module, a subset S of the extracted rules are denoted by a gene string as $S = s_1 s_2 s_3 \ldots s_r$, where $r = i \times l$ is the total number of extracted linguistic rules from the trained neural network, $s_p = 1$ means that the *p-th* rule is included in the rule set S, and $s_p = 0$ means that the *p-th* rule is not included in S. Rows (i) in the matrix represent all the possible premises of a targeted class. Each column (l) represents a class attribute of a target variable.

$$Weight = \begin{bmatrix} s_1 & \cdots & s_{1 \times l} \\ s_2 & \vdots & \\ \vdots & \vdots & \vdots \\ s_{i \times 1} & \cdots\cdots & s_r \end{bmatrix} \rightarrow S = s_1 s_2 \ldots\ldots s_r \qquad (2)$$

The fitness of each gene string is evaluated by

$$Fitness(S) = NCP(S) \qquad (3)$$

where *NCP* is the number of correctly classified data by S. The population is randomly initialized from the extracted rule bases in the previous stage. Starting with a randomly generated population of n r-bit chromosomes. Every chromosome string is evaluated by FIS and ranked by its fitness. Half of the chromosome strings, with the highest fitness in the current population, are selected as parents. Then, 1-point cross over is employed for generating child strings from their parents. Mutation is then applied to the newly generated strings and is kept at the last chromosome of the population. The roulette wheel operator is used to create a new population for the next generation. In this application, the genetic operations were repeated for 1000 generations. The following parameter settings were used:

Number of generations:	1000
Population size:	21
Crossover:	1-point uniform
Mutation rate:	≈ 0.007 to 0.004
Reproduction:	roulette wheel

3. Data Cleaning and Preprocessing

The 1998 KDD Cup database was divided into training and testing sets. The training set contains 95,412 records with 481 variables. The testing set contains 96,367 records with 479 variables. The two left out fields in the testing set are indicators for response (TARGET_B) and donation amount (TARGET_D). These two variables were used as an evaluation tool for the competition. It is quite difficult to fit all the variables into the model and see how they interact. There are 285 continuous variables, collected from the 1990 US Census, which reflect the characteristics of the donor neighborhood. The correlation analysis was applied to eliminate highly correlated variables from the 1990 US Census database.

Let X be a data variable with n records and p variables and let μ_{ij}, $i,j = 1...p$, be the mean of all records for each variable, then the elements in the covariance matrix are calculated as

$$C_{i,j}=\frac{\sum_{k=1}^{n}(X_{k,j}-\mu_i)(X_{k,j}-\mu_j)}{n-1}, \quad i,j=1...p \qquad (4)$$

The correlation matrix is defined from the covariance matrix as

$$S_{i,j}=\frac{C_{i,j}}{\sqrt{C_{i,i}C_{j,j}}}, \qquad i,j=1....p \qquad (5)$$

The correlation matrix (285 by 285 matrix) of the 1990 US Census was calculated from 10,000 randomly picked records (n = 10,000) from the training set (see Figure 4).

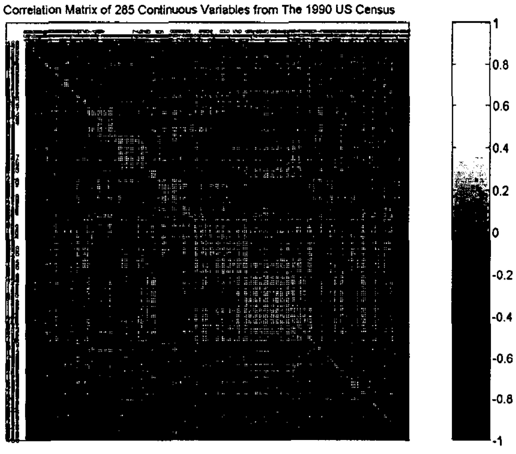

Figure 4. Correlation Matrix of 285 Variables from the 1990 US Census

The above correlation matrix was used to eliminate positively and negatively correlated continuous variables. Variables that correlated more than 80% positively or negatively to other variables were eliminated. After the elimination, only 28 variables were left in the correlation matrix (see Figure 5). The 28 variables are numbered as (1) POP90C1, (2) POP90C3, (3) AGE903, (4) AGE904, (5) AGEC6, (6) AGEC7, (7) MARR3, (8) HU3, (9) HHD9, (10) HVP1, (11) HVP6, (12) RP1, (13) RP4, (14) IC13, (15) TPE5, (16) TPE13, (17) OCC1, (18) EIC1, (19) EIC16, (20) OEDC1, (21) EC4, (22) EC7, (23) AFC2, (24) AFC3, (25) VC1, (26) HC1, (27) HC9, and (28) AC2.

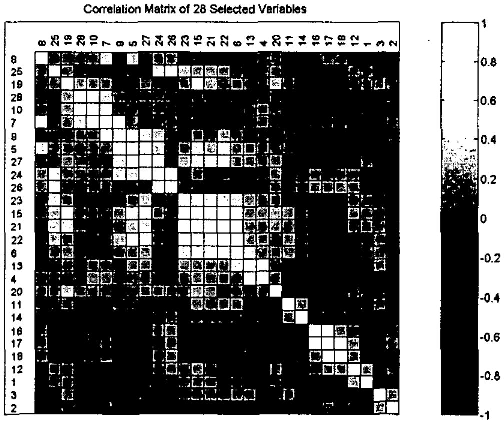

Figure 5. Correlation Matrix of 28 Picked Variables from the 1990 US Census

Besides the 285 variables from the 1990 US Census, there are still 194 variables in the KDD-CUP98 database. The criteria to pick the variables from 194 variables are quite simple. The variables that have more than 50% missing values were eliminated from model. The categorical variables that have more than 10 categories were also eliminated from the model. Consequently, 36 variables were picked from the 194 total variables. A total of 64 variables have been selected in the model.

The 36 variables are ODATEDW, DOMAIN, CLUSTER, AGE, INCOME, GENDER, WEALTH1, MALEMILI, MALEVET, VIETVETS, WWIIVETS, LOCALGOV, STATEGOV, FEDGOV, MAJOR, WEALTH2, CARDPROM, MAXADATE, NUMPROM, CARDPM12, NUMPRM12, RAMNTALL, NGIFTALL, CARDGIFT, MINRAMNT, MINRDATE, MAXRAMNT, MAXRDATE, LASTGIFT, LASTDATE, FIRSTDATE, NEXTDATE, TIMELAG, AVGGIFT, CONTROLN, and HPHONE_D. The descriptions of all 64 variables are described in the data documentation.

4. Implementation and Results

Because the proportion of the donor to non-donor data is quite different, approximately 5% to 95%, the architecture used all the donor records (4,843 records) from the training set and sampled the same amount of records from the non-donor records of the training set to classify the data into two groups: respondent and non-respondent. The architecture used a total of 9,686 records to train the neural network and generate rules. The original rule base generated from the architecture has a classification accuracy of 82.18% on the 9,686 records of the training set. After the genetic algorithm rule-pruning module is applied, the classification of the rule base increased to 82.81% (see Figure 6). The original rule base consists of 105 rules. When rule pruning was applied, only 55 rules were selected from the rule base. Below is the pruned rule base generated from the architecture.

IF ODATEDW IS AVERAGE THEN RESPONDENT (11)
IF DOMAIN IS TOWN THEN RESPONDENT (13)
IF AGE IS AVERAGE THEN RESPONDENT (34)
IF AGE IS HI THEN RESPONDENT (88)
IF INCOME IS HI THEN RESPONDENT (69)
IF GENDER IS FEMALE THEN RESPONDENT (75)
IF MALEVET IS HI THEN RESPONDENT (15)
IF VIETVETS IS HI THEN RESPONDENT (19)
IF FEDGOV IS HI THEN RESPONDENT (11)
IF POP90C3 IS LOW THEN RESPONDENT (11)
IF AGEC6 IS LOW THEN RESPONDENT (11)
IF MARR3 IS LOW THEN RESPONDENT (21)
IF HHD9 IS LOW THEN RESPONDENT (21)
IF HVP1 IS HI THEN RESPONDENT (18)
IF HVP6 IS HI THEN RESPONDENT (14)
IF EIC1 IS LOW THEN RESPONDENT (17)
IF EC7 IS HI THEN RESPONDENT (25)
IF NUMPROM IS HI THEN RESPONDENT (26)
IF CARDPRM12 IS AVERAGE THEN RESPONDENT (11)
IF RAMNTALL IS HI THEN RESPONDENT (12)
IF CARDGIFT IS HI THEN RESPONDENT (26)
IF MINRAMNT IS AVERAGE THEN RESPONDENT (19)
IF LASTGIFT IS LOW THEN RESPONDENT (52)
IF LASTDATE IS HI THEN RESPONDENT (42)
IF FIRSTDATE IS AVERAGE THEN RESPONDENT (11)
IF CONTROLN IS LOW THEN RESPONDENT (29)
IF HPHONE_D IS HI THEN RESPONDENT (94)

IF DOMAIN IS RURAL THEN NON-RESPONDENT (54)
IF AGE IS LOW THEN NON-RESPONDENT (61)
IF INCOME IS LOW THEN NON-RESPONDENT (94)
IF WEALTH1 IS LOW THEN NON-RESPONDENT (100)
IF WEALTH1 IS AVERAGE THEN NON-RESPONDENT (30)
IF VIETVETS IS AVERAGE THEN NON-RESPONDENT (13)
IF WEALTH2 IS LOW THEN NON-RESPONDENT (17)
IF WEALTH2 IS AVERAGE THEN NON-RESPONDENT (12)
IF POP90C1 IS LOW THEN NON-RESPONDENT (13)
IF AGE903 IS AVERAGE THEN NON-RESPONDENT (10)
IF AGEC6 IS AVERAGE THEN NON-RESPONDENT (20)
IF HU3 IS AVERAGE THEN NON-RESPONDENT (13)
IF HVP1 IS AVERAGE THEN NON-RESPONDENT (11)
IF RP4 IS LOW THEN NON-RESPONDENT (26)
IF IC13 IS LOW THEN NON-RESPONDENT (19)
IF TPE13 IS LOW THEN NON-RESPONDENT (11)
IF OCC1 IS LOW THEN NON-RESPONDENT (26)
IF EIC1 IS HI THEN NON-RESPONDENT (11)
IF AC2 IS AVERAGE THEN NON-RESPONDENT (16)
IF CARDPM12 IS LOW THEN NON-RESPONDENT (32)
IF RAMNTALL IS LOW THEN NON-RESPONDENT (16)
IF MINRAMNT IS HI THEN NON-RESPONDENT (36)
IF MINRDATE IS LOW THEN NON-RESPONDENT (12)
IF LASTGIFT IS HI THEN NON-RESPONDENT (23)
IF FIRSTDATE IS HI THEN NON-RESPONDENT (11)
IF NEXTDATE IS LOW THEN NON-RESPONDENT (31)
IF TIMELAGE IS HI THEN NON-RESPONDENT (11)
IF CONTROLN IS HI THEN NON-RESPONDENT (16)

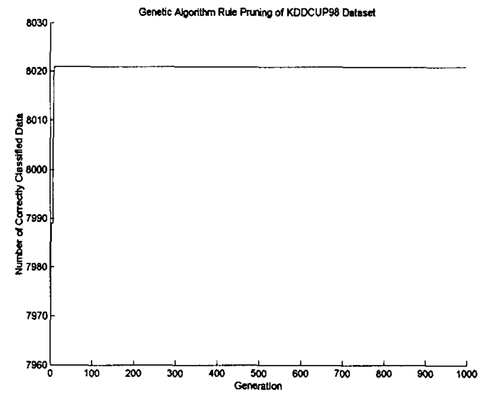

Figure 6. Plot Between Generation and the Number of Correctly Classified Data of KDD-CUP98 Database

To evaluate model efficiency, the classification results from the pruned rule base were compared with the actual donor response results (TARGET_B). By using TARGET_D, the actual donation amount was correctly classified by the pruned rule base as respondents were summed. Profit generated from the model can be calculated by the summation of the donation amounts, which were correctly classified by the pruned rule base minus the number of records selected as donors from the pruned rule base multiplied by the cost of the mailings ($0.68). The pruned rule base or model has been evaluated with the entire training and testing sets. For the training set, sending mail to everyone resulted in a profit of $10,788.55. The model sent out only 68,088 mailers (out of 95,412 cases) resulting in a profit of $12,641.23. For the testing set, sending mail to everyone resulted in a profit of $10,560.28. The model sent out only 72,298 mailers (out of 96,367 cases) resulting in a profit of $13,001.21, which is about a 23.27% increase in revenue comparing to the "sent all" method.

From the above rule base, some strong rules have shown some explicit patterns in each group. The respondents are normally female (GENDER IS FEMALE (75)), elderly (AGE IS HI (88)), wealthy (INCOME IS HI (69)), and have listed home phone numbers (HPHONE_D IS HI (94)). The non-respondents are normally young (AGE IS LOW (61)), and not wealthy (WEALTHY1 IS LOW (100)), live in rural areas (DOMAIN IS RURAL (54), and have low income (INCOME IS LOW (94)).

The model or rule base can help the organization to determine a marketing strategy to target the potential donors for future donation campaigns.

5. Conclusions

This paper has shown a successful application of a neuro-fuzzy-genetic data mining architecture in an American charitable organization's donor database. The goal is to increase donations and to help the organization understand the reasons behind the lack of response to renewal mail sent to donors who had made a donation in the past. The generated rule base from the architecture has proven it usefulness in making more profit than the "sent all" method in both the training and testing set. The characteristics of the donors in each group are shown from the rule base.

References

[1] DeSarbo, W.S., Ramswamy, V., 1994, *CRIPS: Customer Response Based Iterative Segmentation Procedures for Response Modeling in Direct Marketing*, Journal of Direct Marketing, Vol. 8, pp. 7-20.

[2] Fayyad, U, Piatetsky-Shapiro, G, Smyth, P, 1996, *Knowledge Discovery and Data Mining: Towards a Unifying Framework*, in Proceedings of The 2nd International Conference on Knowledge Discovery & Data Mining, KDD-96, Oregon 1996.

[3] Piatetsky-Shapiro, G, Brahman, R. Khabaza, T, 1996, *An Overview of Issues in Developing Industrial Data Mining and Knowledge Discovery Applications*, in Proceedings of The 2nd International Conference on Knowledge Discovery & Data Mining, KDD-96, Oregon 1996.

[4] Haykin, Simon, 1999, *Neural Networks, A Comprehensive Foundation*, Prentice Hall, 2nd Edition, pp. 392-440.

[5] Demuth, Howard, Beale, Mark, 1998, *Neural Network Toolbox for Use with MATLAB*, Version 3.0

[6] Wasserman, P.D., 1993, *Advanced Methods in Neural Computing*, New York: Van Nostrand Reinhold, pp. 35-55.

[7] Kasabov, N, 1996, *Learning Fuzzy Rules and Approximate Reasoning in Fuzzy Neural Networks and Hybrid Systems*, Fuzzy Sets and Systems 82(1996) pp.135-149, Elseveier.

[8] Towell, G, Shavlik, J, 1993, *Extracting Refined Rules from Knowledge-Based Neural Networks*, Machine Learning, 13, pp.71-101.

[9] Matthews, C., Jagielska, I., 1995, *Fuzzy Rule Extraction From Multilayered Neural Network*, Proceedings of IEEE International Conference on Neural Networks (ICNN'95), Perth.

[10] Ishibushi, H., Nozaki K & Yamamot N, 1993, *Selecting Fuzzy Rules by Genetic Algorithm*, IEEE Int Conf Fuzzy System (ICFS), San Francisco.

[11] Hemsathapat, K., Dagli, C.H., *Neural Network Model for Detecting Rare Patterns in Data: A Telecommunication Example*, 1999, in Intelligent Engineering Systems through Artificial Neural Networks, eds. Dagli et. al, Vol.9, ASME Press, pp. 525-530.

[12] Bigus, Joseph P., 1996, *Data Mining with Neural Networks,* McGraw Hill.