NRC: A Neuro-Rough Classifier for Landmine Detection

Ajay Choudhari[1], G. C. Nandi[2] and R. D. Choudhari[3]

Abstract— **Landmines are a significant barrier to financial, economic and social development in various parts of the world. The demand for dependable, trustworthy intelligent diagnostic systems in the field of landmine detection has been increasing rapidly. Metal detectors used in mine-decontamination, cannot differentiate a mine from metallic debris where the soil contains large quantities of metal scraps and cartridge cases, so a device is required that will reliably confirm that the ground being tested does not contain an explosive device, with almost perfect reliability. Human experts are unable to give belief and plausibility to the rules devised from the huge databases.**

In this paper a hybrid classifier has been developed that uses rough sets theory and neural networks architecture to classify mines from the non-mines with better results. Rough sets have been applied to classify the landmine data because in this theory no prior knowledge of rules are needed, these rules are automatically discovered from the database. The rough logic classifier uses lower and upper approximations for determining the class of the objects. The neural network is used for training the data, and has been used especially to avoid the boundary rules given by the rough sets that do not classify the data with cent percentage probability. Moreover, the algorithms based on the rough set theory are particularly suited for parallel processing architecture.

Keywords - **decision matrix, neural networks, rough sets, soft computing.**

I. Introduction

Modern mines have minimal metal content to make them harder to detect. A considerable challenge exists with the processing of noisy signals and images sent by the sensory system. Useful information can only be derived with a carefully designed sensor fusion system backed by strong signal and image processing algorithms. The control algorithms must make sure the mine detector robot is able to manipulate in unknown, uncertain environments, it should be stable, and it should be able to learn from interaction and can improve performance over time. Complex application problems, such as reliable monitoring and diagnosis of industrial plants, are likely to present large numbers of features, many of which will be redundant for the task at hand [3,4]. Additionally, inaccurate and uncertain values cannot be ruled out. The landmine databases are large, and tracing general knowledge from databases is known to be the most difficult part of creating a knowledge-based system. The most common approach to developing expressive and human readable representations of knowledge is the use of if–then production rules [5]. Yet, real-life problem domains usually lack generic and systematic expert rules for mapping feature patterns onto their underlying classes.

The paper aims to induce low-dimensionality rule sets from historical descriptions of domain features which are often of high dimensionality. A common disadvantage of techniques applied is their sensitivity to high dimensionality, i.e., PCA [6] irreversibly destroys the underlying semantics of the feature set. Most semantics-preserving dimensionality reduction (or feature selection) approaches tend to be domain specific, i.e., relying on the use of well-known features of the particular application domains.

Given a dataset with some feature values, it is possible to have a subset (termed as reduct) of the original features using Rough Set Theory (RST) [7,8] that are the most informative; all other features can be removed from the dataset with minimal information loss. RST is an alternative approach that preserves the underlying semantics of the data while allowing reasonable generality. It is, therefore, desirable to develop this technique to provide the means of data reduction for crisp and real-valued datasets which utilizes the extent to which values are similar. And the neural networks have been used in the training of data and the classification of objects which came under the boundary rules of the rough sets.

II. State Of Art

Presently the different algorithms have been used for a classification and pattern recognition. Soft computing is a consortium of methodologies that provides information processing capabilities for handling real life ambiguous situations. The soft computing comprises of Fuzzy Logic, 'FL' [14]; Rough Sets , 'RS' [7]; Artificial Neural Networks , 'ANN' [13] and Genetic Algorithms , GA.

A. Fuzzy Logic

Fuzzy sets is a way to represent vagueness. They are generalizations of conventional crisp theory. FL provides a simple way to arrive at a definite conclusion based upon vague, am-

[1,2] Indian Institute of Information Technology-Allahabad, India.
[3] Indian Institute of Technology (IIT-Delhi), New Delhi, India.

0-7803-9503-4/05/$20.00 ©2005 IEEE

biguous, imprecise, noisy, or missing input information. Fuzzy logic provides the algorithm for dealing with imprecision and uncertainty arising from vagueness rather than randomness.

B. Rough Sets

Rough Sets is used for handling uncertainty arising from the limited discernibility of objects. RS methodology provides definitions and methods for finding which attributes separates one class or classification from another. The results from a training phase when using the Rough Sets approach will usually be a set of propositional rules which may be said to have syntactic and semantic simplicity for a human.

C. Artificial Neural Network

ANN is said to be the machinery for learning and adaptation. It is composed of a large number of highly interconnected processing elements i.e., neurons working in unison to solve specific problems. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends.

D. Genetic Algorithm

Genetic algorithms have been established as a viable technique for search, optimization, machine learning and other problems. GAs are executed iteratively on a set of coded solutions, called population, with three basic operators: selection/reproduction, crossover and mutation. They use only the objective function and probabilistic transition rules for moving to the next iteration.

III. System Architecture

The system architecture shown in figure 1 consists mainly four components

1. Initial Categorizing System
2. Rough Logic System
3. Neural Networks System
4. A Decision Subsystem

The database has been given to the system, which is in the form of the redundant and inconsistent tuple. It's first given to the categorizing system which categorizes the database into "very small", "small", "average", "high" and "very high" classes. The database now converted into discrete database. This discrete database is now act as input into "Rough Logic System" where it's divided into different sets like "lower bound", "upper bound", "boundary set" etc.

After this, the outputs from "Rough Logic System" goes into

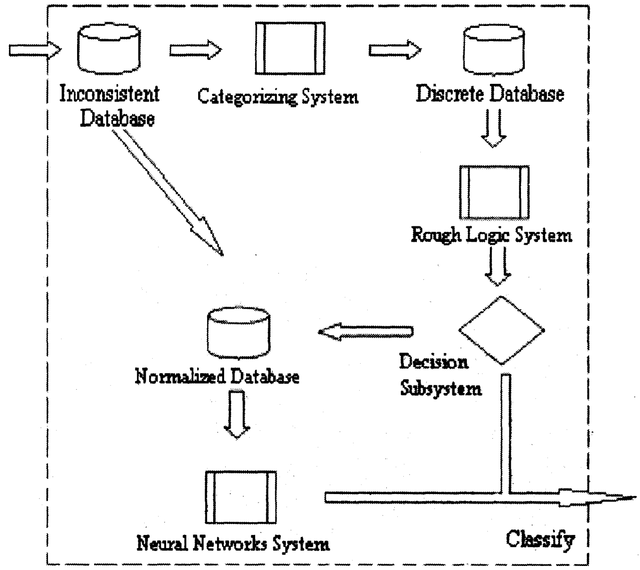

Fig. 1 The system architecture

"Decision Sub-system" and a decision has been made whether to send the inputted query tuple to the "Neural Network System" or just classify it based on rough sets classifier itself.

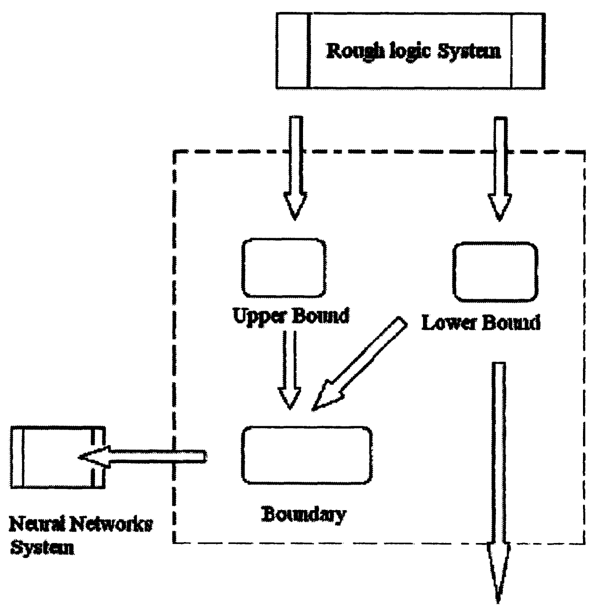

Fig. 2 The Flow diagram of decision sub-system

The detail architecture of decision sub-system has been shown in fig. 2.

IV. Implementation Details

A conventional technique for landmine detection has been metal detector. But this technique fails to catch plastic mines. Hence, infra red images of suspected areas are taken. The analysis of landmines begins after feature extraction from various sensors.

A. Feature selection

The features which are taken into consideration are:

Blob Aspect Ratio: Firstly mines are more likely to be circular while the metal scrap in ground is more likely to be tube, cylindrical or cubical.

Gray scale value: Secondly, mines are more likely to be metallic and hence grayscale value of the blob can be used to differentiate them from wooden pieces in the ground.

Size: The mines are more likely to be larger in size as compared to simple material in ground.

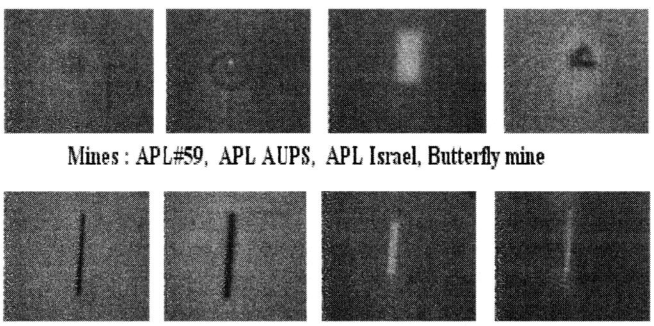

Fig 3. Sensor images of different mine and non-mine data

B. Initial Categorizing System

The database taken which consists of inconsistent and redundant data, has been categorized. The database is now divided into classes namely "very small", "small", "average", "high" and "very high". Here the proper considerations have been made which checks that the classes would not be affected by the extreme values of the attributes.

C. Rough Sets System

Rough Set theory helps to derive simple rules which can be used to classify mines. It is assumed that data is presented in form of information tables, column of which represent attributes (Blob Size, Blob Aspect, Gray Scale) and decision variable (mine or non mine), and each row is a tuple showing values of each object for these attributes. Let A be a set of attributes chosen to classify the tuple. The basic intuition of the approach lies in the fact that all tuple having same values for all attributes in A are likely to have same value of decision attributes. The set of all tuple are divided into indiscernible sets such that elements of each set have same value at each attribute in A [10].

The crisp rules will exist to classify crisp sets while no such rule exists for rough set. Table 1 shows a small part of real data extracted from images provided by JRC (Joint Research Centre) landmine signature database. The numerical values of attributes have been categorized into very low, low, average, high and very high.

Table I: An abstract view of the categorized landmine database

	Blob Aspect	Blob Size	Gray Scale	Type
O_1	very low	very low	very low	Mine
O_2	very low	very low	very low	Non-mine
O_3	very low	very low	very low	Non-mine
O_4	very low	very low	very low	Non-mine
O_5	Low	very low	very low	Non-mine
O_6	Low	very low	very low	Non-mine
O_7	Low	very low	very low	Non-mine
O_8	very low	very high	average	Mine
O_9	very low	very high	average	Mine
O_{10}	very low	very high	average	Mine

Considering A={Blob Aspect, Blob Size, Gray Scale} the indiscernible sets are:

i. $\{O_1, O_2, O_3, O_4\}$
ii. $\{O_5, O_6, O_7\}$
iii. $\{O_8, O_9, O_{10}\}$

Now the required set is $D = \{O^1, O^8, O_9, O_{10}\}$

Since it is impossible to express D in terms of union of I_1 and I_2 so D is a rough set. An example of a crisp set is $C = \{O_5, O_6, O_7, O_8, O_9, O_{10}\}$. Also, it is apparent that no such rule exists for testing membership in D.

Different Approximations

Since it is not possible to formulate simple rules to test membership in D, upper and lower bounds of D are calculated. In our case $T = \{I_1, I_2, I_3\}$. Upper bound of rough set D can be defined as Upper(D) = U (X*i*) such that Xi Є T and Xi ∩ D ≠φ Similarly, lower bound of rough set D is Low(D) = U(Xi) such that Xi Є T and Xi ⊆ D. In given case Upper(D) = I_1 U $I_3 = \{O_1, O_2, O_3, O_4, O_8, O_9, O_{10}\}$ and Low(D) = $I_3 = \{O_8, O_9, O_{10}\}$.

Now, given the values of an object O for all attributes in A, to predict whether O is a mine or not one can proceed as follows:

If (O is not an element of Upper(D))
O is certainly not a member of D
else if (O Є Low(D)) O is certainly a member of D
else O may or may not be a member of D.

Boundary Set

One is unsure about membership O only if it belongs to upper bound but not to lower bound. This region is called boundary region of D. The decision sub-system has been prepared for this only. Thus, Boundary (D) = Upper(D) – Low(D).

If the object lands up in boundary region then we say O may or may not belong to D. Let I{O} denote the equivalence class to which the object O belongs. Then the degree of membership ∊ is defined as

$\alpha(O)$ in D = [I{O}∩D] / I{O}.

It follows that if O ∈ Low (D), I{O} ⊆ D, hence I{O}∩ D = I{O}, so $\alpha(O)=1$. Similarly, if O ∈ Upper(D), I{O}∩D = φ and so $\alpha(O) = 0$. For boundary region $\alpha(O)$ lies in (0,1). Thus alpha denotes the certainty with which O ∈ D [17].

Equivalence Class

Equivalence classes has been made based on the decision attributes values, the whole database is divided into various classes. The equivalence classes formed by table I is shown in figure 4.

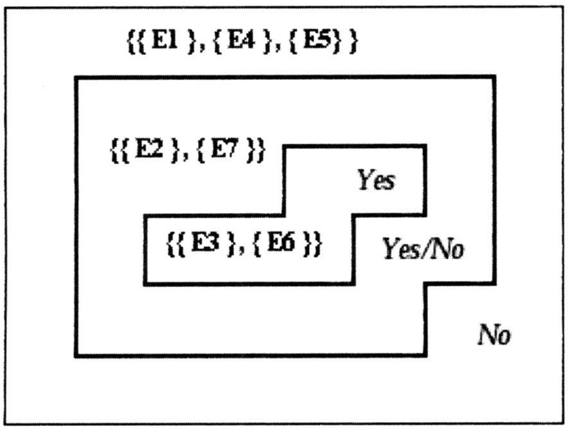

Fig.4. Equivalence classes obtained using rough theory

Discernible Attributes and Decision Matrix

A decision matrix has been created for the indiscernibility of the various tuple in the database. The Decision matrix shows how the values in a tuple is different from all the tuple in the database, which helps in the formulation of the rules. The decision matrix has been shown in fig. 5.

Rule Extraction

The rules has been extracted form the decision matrix. The rules that are given by the lower bounds only are taken into consideration.

To analyze the importance of an attribute partial dependency [10] M→N is calculated, where M and N are sets of attributes. The attribute set N is said to depend on M with degree α, where

$$\beta = \frac{|Pos_M(N)|}{|U|}$$

where

$$Pos_M(N) = \bigcup_{X \in (U/N)} \text{Low}(X)$$

and U is the universal set, and |S| denotes cardinality of set S. $Pos_M(N)$ is called positive region of partitions with respect to M, and it represents the set of all such elements which can be uniquely classified to elementary sets formed by I(N). Thus β is ratio of number of elements which can be classified (by means of M) to elementary sets formed by N, to the total number of elements. A high value of β between M and N indicate that N does not carry much information and the attribute set M will classify nearly as good as the attribute set M U N.

equvclas	x12	x13	x14	x15	x16	x17	x18	x19	x20	x21	x22	x23	x24
1	b	b	bc	abc	ab	abc	ab	abc	ab	ac	ab	ab	abc
2	bc	bc	bc	abc	abc	abc	abc	abc	abc	ac	abc	abc	abc
3	bc	bc	bc	abc	abc	abc	abc	abc	abc	abc	abc	abc	abc
4	abc	abc	abc	abc	abc	abc	abc	abc	abc	abc	abc	abc	abc
5	abc	abc	abc	ab	abc	abc	abc	abc	abc	abc	abc	abc	abc
6	ab	ab	abc	bc	b	bc	b	abc	ab	ac	ab	ab	abc
7	ab	ab	abc	bc	b	bc	b	abc	ab	abc	ab	ab	abc
8	ab	ab	abc	abc	ab	abc	ab	abc	ab	abc	ab	ab	abc
9	abc	abc	abc	abc	abc	abc	abc	abc	abc	abc	abc	abc	abc
10	ab	ab	abc	abc	ab	abc	ab	bc	b	bc	b	b	bc
11	abc	abc	ab	abc	abc	ab	abc	bc	bc	bc	bc	bc	b
12		b	bc	ac	ab	abc	ab	ac	ab	abc	ab	a	ac
13			c	abc	a	ac	ab	abc	a	abc	ab	ab	abc
14				abc	ac	a	abc	abc	ac	abc	abc	abc	ab
15					bc	bc	bc	ac	abc	abc	abc	ac	ac
16						c	b	abc	a	abc	ab	ab	abc
17							bc	abc	ac	abc	abc	abc	ab
18								abc	ab	abc	a	ab	abc
19									bc	b	bc	c	c
20										bc	b	b	bc
21											bc	bc	bc
22												b	bc
23													c
23													c

Fig.5. The Decision Matrix containing discernible attributes

D. Decision Sub-System

The Decision Subsystem gets the input from the "Rough Logic System". In this sub-system a decision has been taken whether to directly classify the tuple which is operated upon or send it into the next system of neural networks. In this sub-system, a boundary set has been calculated i.e., Boundary set = (Upper Bound – Lower Bound). If the classification attributes are in the rules derived from the lower bounds of the rough set classifier then the tuple is classified into mines with 100% probability, else that tuple is send into the neural networks classifier with the normalized value. And finally it gave the result with certain probability and the result above a cut off can be classified as mines, but that value has to be quite high so that the tuple can be classified as mines. The stress has been given that the concerned data can be safely classified as mine or non-mine, but if any ambiguity is there then it's wiser to classify it as mine with certain probability, because if a mine is classified as non-mine then it can cause a lot of destruction

E. Neural Networks System
Normalize the database

If the output of decision sub-system shows that the data lies under the boundary region then it's forward to the neural network sub-system. Before sending the data which is in discrete form now, has to be traced back for its original value and then that has to be normalized accordingly to the entries present in the database. Now this can be sent into the neural network system for further classification.

Training the neural network

The Back propagation model of ANN has been used as the final classification system for the ambiguous tuple, i.e., which came under the boundary region of Rough sets classifier. The Back Propagation algorithm modifies the network weights to minimize the mean squared error between the desired and the actual outputs of the networks.

The sigmoid function is used as the activation function i.e.,

$$y_j = \frac{1}{1+e^{-x_j}}$$

where y_i is the activity level of the j^{th} unit in the previous layer. The network computes the error E of the activities of all output units as:

$$E = \frac{1}{2}\sum_i (y_i - d_i)^2$$

where y_j is the activity level of the j^{th} unit and d_j is the desired output of the j^{th} unit.

The back-propagation algorithm consists of four steps:

1. Compute the error derivative, EA i.e., the difference between the actual and the desired activity.

$$EA_j = \frac{\partial E}{\partial y_j} = y_j - d_j$$

2. Compute the error change as the total input received by an output unit is changed. EI is EA multiplied by the rate at which the output of a unit changes as its total input is changed.

$$EI_j = \frac{\partial E}{\partial x_j} = \frac{\partial E}{\partial y_j} \times \frac{dy_j}{dx_j} = EA_j y_j (1 - y_j)$$

3. Compute the error change as a weight on the connection into an output unit is changed. EW is EI multiplied by the activity level of the unit from which the connection emanates.

$$EW_{ij} = \frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial x_j} \times \frac{\partial x_j}{\partial W_{ij}} = EI_j y_i$$

4. Compute the error change as the activity of a unit in the previous layer is changed. When the activity of a unit in the previous layer changes, it affects the activities of all the output units to which it is connected. So to compute the overall effect on the error, these separate effects on output units have to be added together. It is the EI multiplied by the weight on the connection to that output unit.

$$EA_i = \frac{\partial E}{\partial y_i} = \sum_j \frac{\partial E}{\partial x_j} \times \frac{\partial x_j}{\partial y_i} = \sum_j EI_j W_{ij}$$

By using steps 2 and 4, the EAs of one layer of units is converted into EAs for the previous layer. This procedure can be repeated to get the EAs for as many previous layers as desired. Once the EA of an unit is known, we can use steps 2 and 3 to compute the EWs on its incoming connections. This back propagation model has been implemented in the paper.

After getting the minimum error the weights have been saved. The tuple which is inputted in this system has passed through the network and now classified as mine or non-mine with a probability. And for safety purposes it has been considered that the mines can never be classified as non-mines but the reverse can be true because if a mine can be classified as non-mine it can cause a lot of destruction.

In this way the tuple which came under the boundary region can be put into the neural network system and can be classified accordingly.

V. Result And Conclusion

The dataset consists of total 142 tuple and out of these 100 are used to make the hybrid classifier. After making the classifier, the remaining 42 tuple are used to test the classifier. Out of 142 tuple, the 100 tuple are chosen randomly and the remaining 42 are used as test in different ways, the following tables shows the results of different classifiers.

I: The worst case classification

Classifier	Test tuple	No. of classifications	No. of mis-classifications	% Accuracy
Rough Sets	42	33	09	78.57%
ANN	42	35	07	83.33%
Hybrid Classifier	42	37	05	88.09%

II. The best case classification

Classifier	Test tuple	No. of classifications	No. of mis-classifications	% Accuracy
Rough Sets	42	39	03	92.85%
ANN	42	38	04	90.47%
Hybrid Classifier	42	41	01	97.23%

The mutual dependencies of attributes among themselves is negligible except the {BlobAspect, BloSize}→ GrayScale dependency. As a consequence, the degree of {BlobAspect, BlobSize}→Type dependency is nearly same as that of {BlobAspect, BlobSize, GrayScale }→ Type dependency thus identifying GrayScale as least important feature. This result is quite expected, as the debris considered contains non metals like piece of wood, as well as metals like copper.

The result of the hybrid classifier has an improvement over both the rough set classifier and the neural network classifier and has better result in both the worst and the best case scenario. This is due to fact that hybrid classifier uses the best feature of the rough sets i.e., the lower bound rules and send rest to the neural network classifier for further classification.

The rough sets is used to find the 'first rule set', an abstract view of the rules obtained can be seen in fig. 5. From this the final rule set is obtained. The size of final rule set is 27.

A	B	C	D	Cardinality	Class	Decision Class
0	2	0	0	6	1	1
0	2	1	0	6	2	1
0	4	1	0	3	3	1
1	4	1	0	8	4	1
1	4	3	0	7	5	1
2	2	0	0	4	6	1
2	4	0	0	3	7	1
3	4	0	0	2	8	1
3	4	1	0	1	9	1
4	4	0	0	6	10	1
4	4	2	0	1	11	1
0	0	0	1	1	12	2
0	1	0	1	8	13	2
0	1	2	1	6	14	2
2	0	3	1	1	15	2
2	1	0	1	1	16	2
2	1	2	1	3	17	2
2	3	0	1	7	18	2
4	0	4	1	4	19	2
4	1	0	1	5	20	2
4	2	4	1	4	21	2
4	3	0	1	7	22	2
4	0	0	0	2	23	2
4	0	0	1	1	23	2
4	0	2	0	1	24	2

Fig.6. An abstract view of rough logic rule set of the classifier.

The efficiency can further be increased by improving the neural network classifier which is used to classify the ambiguous data. The genetic algorithm can be used for better and optimized results of the neural network classifier.

The designed classifier can be used on the parallel computing architecture. The different task can be done parallelly to speed up the classification. The architecture can be broken into different granules and can be given to different processors for their respective processing. A Specialized processor can be used to optimize rough set calculations.

REFERENCES

[1] Lech Polkowski; Rough Sets Mathematical Foundations. ISBN - 3-7608-1510-1.

[2] Mehrotra, Kishan; Mohan, Chilukuri K.; Ranka, Sanjay; Elements of Artifical Neural Networks; The MIT Press. ISBN - 0-262-13328-8.

[3] W. Pedrycz, G. Vukovich, Feature analysis through information granulation, Pattern Recognition 35 (4) (2002) 825–834.

[4] M. Sebban, R. Nock, A hybrid filter/wrapper approach of feature selection using information theory, Pattern Recognition 35 (4) (2002) 835–846.

[5] N. Kwak, C.H. Choi, Input feature selection for classification problems, IEEE Trans. Neural Networks 13 (1) (2002) 143–159.

[6] B. Flury, H. Riedwyl, Multivariate Statistics: A Practical Approach, Prentice-Hall, Englewood Cli/s, NJ, 1988.

[7] Pawlak, Z.Rough Sets. International Computer Science, 1982, 11.

[8] Pawlak, Z. Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic, 1991.

[9] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning About Data, Kluwer Academic Publishing, Dordrecht, 1991.

[10] Zdislaw Pawlak. Why Rough Sets? Proc. Of The 5th IEEE International Conf. On Fuzzy Systems, Vol. 2,Pp 738-743,1996.

[11] Zdzislaw Pawlak, "Granularity Of Knowledge, Indiscernibility And Rough Sets", Proc. Of IEEE International Conf. On Fuzzy Systems: FUZZ-IEEE 98, Vol. 1,Pp. 106-110,1998

[12] Daijin Kim And Sung-Yang Bang: A Handwritten Numerical Character Classififcation Using Tolerant Rough Set, IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 22,No. 9,September 2000.

[13] M. Szczuka, "Refining Classifiers with Neural Networks," Int'l J. Computer and Information Sciences, vol. 16, pp. 39-56, 2001.

[14] L.A. Zadeh. Fuzzy sets. Inforamtion and control, 8,338- 353.