Malaysian Journal of Computer Science, Vol. 12 No. 1, June 1999, pp. 1-8

HYBRID EXPERT SYSTEM OF ROUGH SET AND NEURAL NETWORK

Moawia Elfaki Yahia and Ramlan Mahmod
Department of Computer Science
Faculty of Science and Environmental Studies
Universiti Putra Malaysia
43400 UPM Serdang, Selangor, Malaysia
emails: moawia@fsas.upm.edu.my
ramlan@fsas.upm.edu.my

ABSTRACT

The combination of neural network and expert system can accelerate the process of inference, and then rapidly produce associated facts and consequences. However, neural network still has some problems such as providing explanation facilities, managing the architecture of network and accelerating the training time. Thus to address these issues we develop a new method for pre-processing based on rough set and merge it with neural network and expert system. The resulting system is a hybrid expert system model called a Hybrid Rough Neural Expert System (HRNES).

Keywords*: *Hybrid systems, Expert systems, Rough sets, Neural networks, Data Pre-processing

1.0 INTRODUCTION

Neural network can serve as a knowledge base of expert systems that do classification tasks. Neural network depends on training data to program the systems, so that it can support the knowledge acquisition of expert systems where historical data are available. In situation where rules cannot be directly determined or when it may take too long to elicit them, neural network can be useful for fast identification of implicit knowledge by automatically analysing cases of historical data. A trained neural network then rapidly processes information to produce associated facts and consequences. Therefore, knowledge base inside the expert system can be replaced by associative memory that encodes knowledge implicitly as connection weights.

Despite the advances of these systems, debugging the knowledge base of expert system is still a big problem. Also the architecture of neural network and accelerating the training time are important issues. Moreover, most neural network systems lack explanation facilities. A neural network does not provide any comprehensive explanations as to how the input attributes are used to produce the output predictions. With current technologies, training times can be excessive and tedious; thus, the need for frequent retraining may make a particular application impractical. The best way to represent input data, and choice of architecture and number of nodes and layers, are still subject to trial and error. Furthermore, there is no way to determine the relative importance of input data in neural network. That is needed to filter out the usable information from the large amount of input noise. Of course, once the network is in actual use, the process of gathering unnecessary data is costly and simply confuses the issue of explaining how the network reaches final predictions.

Rough set theory, introduced by Pawlak in 1982 [10, 11], is a new mathematical tool to deal with vagueness and uncertainty. It has proved its soundness and usefulness in many real life applications. Rough set theory offers effective methods that are applicable in many branches of AI. The idea of rough set consists in approximation of a set by a pair of sets called the lower and upper approximations of the set. The definition of the approximations follows from an indiscernibility relation between elements of the sets, called objects. Objects are described by attributes of a qualitative or quantitative nature. Rough set approach to expert system appears in rule induction of expert system by providing two sets of rules; certain rules and possible rules. Also the rough set can be a useful tool for pre-processing data for neural networks by applying its concept of attribute reduction to reduce the network's input vector, and hence to scale down the size of the whole architecture of the network.

2.0 LITERATURE SURVEY

In recent years, models for developing appropriate hybrid systems using artificial intelligence (AI) technologies have appeared. One reason of this approach is to build more powerful systems that can reduce drawbacks of implementing a single AI technologies alone. The development of integrated technologies of neural network (NN) and expert systems (ES) has shown some advancement. The complementary features of neural networks and expert systems allow the combination of these two technologies to make more powerful systems than can be built with either of the two [9]. This integration leads to the emergence of a new systems called Neural Expert Systems (NES) or Connectionist Expert Systems (CES), firstly introduced by Gallant in

1988 [2]. Neural expert systems are expert systems that have neural network for their knowledge bases.

Many researches have been done to improve this integration. Beri and Trotta [1] propose a way to improve the connectionist expert system by introducing the dependency graph among inferential nodes. The dependency graph enhances the determination of significant input variables for evaluating an output goal. Two drawbacks of this method are: inefficient backward chaining mechanism and lack of explanation in inferential process. Glorfeld [3] presents a methodology to simplify network models by using a backward selection process to eliminate input variables that are not contributing to the network ability. Although the developed models based on the important measure are good, there is no guarantee that the selected models are actually the optimal models. The heuristic basis of the variable selection procedure means that this methodology cannot be guaranteed to automatically work in all problem domains and with all data sets. Hayashi [4] proposed a novel expert system using fuzzy teaching input by generalization of the information derived from training data and embodiment of knowledge in the form of a fuzzy neural network. A method was proposed to extract automatically fuzzy if-then rules from the trained neural network generated by training data with fuzzy teaching input.

Recently, rough set theory approach to the field of neural network seems to be attractive for researchers. An empirical study concerns the use of the rough set approach to reduction of data for neural network classifying objects described by quantitative and qualitative attributes [6]. The promising results show that the rough set approach is a useful tool for pre-processing of data for neural networks. Another research focuses on joining two forms of learning, the technique of neural networks and the technique of rough sets [16]. This approach serves as a promising example that a better understanding of the ability of each technique can lead to novel and useful ways of combining them. In this paper, we contribute to previous works of approaching the rough set to the field of neural network, and offer a novel method for pre-processing of data for the neural network inside the knowledge base of neural expert system [13].

3.0 ROUGH SETS THEORY AND KNOWLEDGE ANALYSIS

The observation that one cannot distinguish objects on the basis of given information about them is the starting point of the rough set theory. In other words, imperfect information causes indiscernibility of objects. The indiscernibility relation induces an approximation space made of equivalence classes of indiscernible objects. A rough set is a pair of a lower and an upper approximation of a set in terms of these equivalence classes [10]. Thus, a rough set can provide a tool for knowledge analysis that helps in discovering relationships between objects and decisions, so as to get a minimum representation of information system in terms of decision rules.

The complete presentation of this theory can be found in Pawlak's book [10]. Applications of the rough set theory, some new theoretical developments and comparisons with related approaches have been recently published [12]. Here, we use some basic concepts, drawn from earlier papers [8, 14].

3.1 Information System and Decision Table

An *information system* is a finite table, the rows of which are labelled by *objects*, whereas columns are labelled by *attributes* and entries of the table are *attribute-values*. By an information system we mean a 4-tuple $S = (U, A, V, f)$, where U is a finite set of objects, A is a finite set of attributes, $V = \cup_{a \in A} V_a$ and V_a is a domain of attribute a, and $f : U \times A \rightarrow V$ is a total function such that $f(x,a) \in V_a$ for every $a \in A$, $x \in U$, called information function. If we distinguish *condition* and *decision attributes* in an information system, we get a *decision table.* The set of condition attributes is then denoted by C and the set of decision attributes by D, s.t. $A = C \cup D$. Then the decision table T can be written as $T = (U,C,D,V,f)$, or shortly, $T = (C,D)$.

3.2 Indiscernibility Relation and Approximation Space

Let $S = (U, A, V, f)$ be an information system and let $P \subseteq A$ and $x, y \in U$, we say that x and y are indiscernible by the set of attributes P in S iff $f(x,a) = f(y,a)$ for every $a \in P$. Thus every $P \subseteq A$ generates a binary relation on U which will be called an *indiscernibility relation,* denoted by IND(P). Obviously, IND(P) is an equivalence relation for any P. Equivalence classes of IND(P) are called P-elementary sets in S. The family of all equivalence classes of relation IND(P) on U is denoted by $U \mid \mathrm{IND}(P)$ or, in short, $U \mid P$.

Let $Y \subseteq U$. The P-lower approximation of Y, denoted by $\underline{P}Y$, and P-upper approximation of Y, denoted by $\overline{P}Y$, are defined as

$$\underline{P}Y = \cup \{ X \in U \mid P: X \subseteq Y\} \text{ and}$$
$$\overline{P}Y = \cup \{ X \in U \mid P: X \cap Y \neq \Phi \}$$

Set $\underline{P}Y$ is the set of all objects from U which can be certainly classified as elements of Y, employing the set of attributes P. Set $\overline{P}Y$ is the set of objects from U which can be possibly classified as elements of Y, using the set of attributes P. The set $\mathrm{Bn}_p(Y)$ is the set of objects which cannot be certainly classified to Y using the set of attributes P only. With every set $Y \subseteq U$, we can associate an *accuracy of approximation* of set Y by P in S defined as

$$\alpha_P(Y) = \text{card}(\underline{P}Y) / \text{card}(\overline{P}Y)$$

We will also need an approximation of a partition of U. Let S be an information system, $P \subseteq A$, and let $\mathrm{Y} = \{Y_1, Y_2, \ldots, Y_n\}$ be a partition of U. By the P-lower approximation of Y in S we mean the sets $\underline{P}\,\mathrm{Y} = \{\underline{P}Y_1, \underline{P}Y_2, \ldots, \underline{P}Y_n\}$. The coefficient

$$\gamma_P(\mathrm{Y}) = \sum_{i=1}^{n} \text{card}(\underline{P}Y_i) / \text{card}(U)$$

is called the *quality of approximation of partition* Y by set of attributes P or, in short, *quality of classification.* It expresses the ratio of all P-correctly classified objects to all objects in the system.

3.3 Reduction and Dependency of Attributes

An important issue is that of attributes reduction in such a way that the reduced set of attributes provides the same quality of classification as the original set of attributes. In the information system S, the minimal subset $R \subseteq P \subseteq A$ such that $\gamma_P(\mathrm{Y}) = \gamma_R(\mathrm{Y})$ is called Y-reduct of P and denoted $\text{RED}_{\mathrm{Y}}(P)$. Let us notice that an information system may have more than one Y-reduct. Intersection of all Y-reducts is called the Y-core of P denoted by $\text{CORE}_{\mathrm{Y}}(P)$, i.e. $\text{CORE}_{\mathrm{Y}}(P) = \cap\, \text{RED}_{\mathrm{Y}}(P)$.

Discovering dependencies among attributes is of primary importance in the rough set approach to knowledge analysis. We will say that set of attributes $Q \subseteq A$ *depends* on set of attributes $P \subseteq A$, denoted as $P \to Q$, if each equivalence class of the equivalence relation generated by P is included in some equivalence class generated by Q, i.e. $P \to Q$ if $\text{IND}(P) \subseteq \text{IND}(Q)$. Obviously, Q depends on P if values of attributes in Q are uniquely determined by values of attributes in P, thus there is a functional dependency between values of Q and P. Furthermore, the relative significance of an individual attribute $a \in P$ with respect to dependency between P and Q is represented by *significance factor* SGF(a,P,Q), given by

$$\text{SGF}(a,P,Q) = \gamma_P(U \mid Q) - \gamma_{P-\{a\}}(U \mid Q)$$

The significance factor reflects the degree of increase of dependency level between P and Q as a result of addition of attribute a to P.

4.0 HYBRID ROUGH NEURAL EXPERT SYSTEM

There are several reasons for developing expert system models that have neural network as their knowledge bases. By using the learning algorithms from previous parts, expert system can be generated from training examples. This would be especially helpful where there is a large amount of noisy data. One reason is the possibility of neural network to make inferences from partial information. This is an important aspect of real-world applications, and connectionist models can be especially good at handling noisy and partial information. The trained neural network inside the knowledge base of the expert system can accelerate the process of inference, and then rapidly produce associated facts and consequences. However, neural network still faces some limitations. This confronts us with some issues such as providing explanation facilities, managing the architecture of network and accelerating the training time. A new method based on rough set analysis is developed and merged with neural expert system. This yields a new hybrid expert system called a Hybrid Rough Neural Expert System (**HRNES**).

4.1 Structure of Hybrid Expert System

The rough neural expert system model is an extension of the expert system model introduced by Gallent [2]. The structure of the new system is given in Fig. 1. The major parts of the system are:

- ***The Knowledge Acquisition*** is an information-director module consisting of two main parts:
    - ***Pre-processing Rough Engine*** is an engine for pre-processing of data for the neural network using the concepts of rough set theory. The process here consists of four phases; data acquisition, decision table formalization, attribute reduction, and binarization (Fig. 2).
    - ***Learning Program*** is a procedure accepting reduced training examples, implementing learning phase by using specific learning algorithm, and then producing a final network to knowledge base in the form of weight matrix.
- ***The Knowledge Base*** is a problem-specific module containing information that controls inferencing. It consists of two main parts:

- ***Neural Network*** is a final network produced by the learning program in the form of weight matrix.

- ***Variable Names & Questions*** are a set of pairs of information used by the inference engine. Each pair consists of a name of each cell in the network corresponding to variables of interest and a question for each input variable to elicit the value of that variable from the user.

- ***Neural Network Inference Engine*** is an expert system inference engine that uses a connectionist network knowledge base. It does the task of inferencing based on partial information.

- ***The User Interface*** is a component that links the inference engine to the external environment using standard programming techniques.

4.2 Pre-Processing Rough Engine

The rough set approach to neural network can appear by providing a tool for pre-processing for neural network. In this paper a new method for pre-processing data for neural network based on rough set has been developed and merged with neural expert system. The process consists of acquisition of data, formalization of decision tables, reduction of attributes, and binarization of input data [7, 15]. First decision table has to be developed from training example cases. By applying the rough approximation concepts to decision table, reduction in attributes can be discovered, and automatically we will achieve reduction in network's input vector, and hence we can manage the whole architecture of neural network.

4.2.1 Acquisition of Data

The first step is to receive a set of training example cases in the form of a matrix in which rows are represented by cases and columns are represented by variables. Each case consists of values of input variables and output variables, and may include values of intermediate variables such as diseases in the case of medical diagnosis. In this case, each training example can be a patient's case history which consists of symptoms, diseases and treatments.

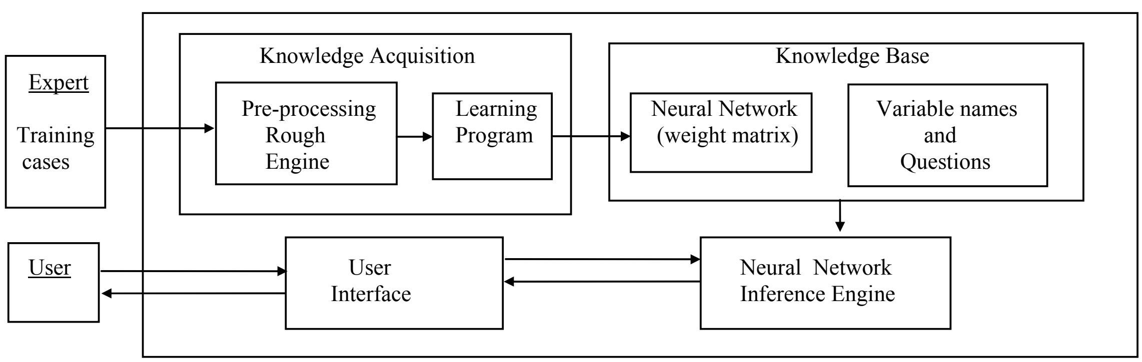

Fig. 1: Structure of Rough Neural Expert System

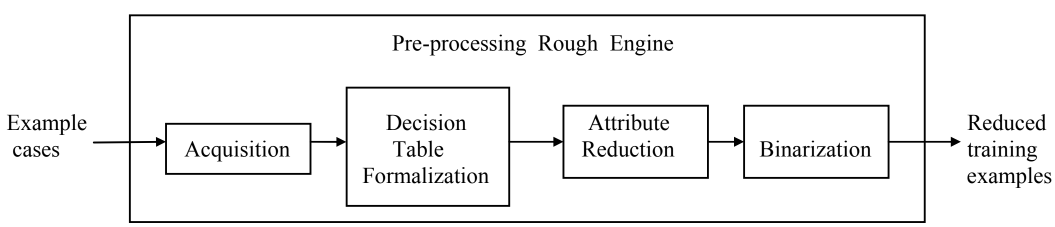

Fig. 2: Phases of Pre-processing Rough Engine

4.2.2 Formalization of Decision Tables

The raw data for neural network can be formulated in a decision table in which input variables are labelled by condition attributes (*C*), output variables (goals) are labelled by decision attributes (*D*), and cases are labelled by objects (*U*). Each interval attribute *p* will be handled by translating its values into ordered nominal terms prescribed by domain (*Vp*). Each nominal attribute *q* should be splitted, according to its cardinality, into binary sub-attributes of domain (Vq_i) (*i*=1 to n). The resulting structure should be a decision table $T = (U,C,D,V)$.

4.2.3 Reduction of Attributes

The objective is to find a minimal subset of related attributes that preserves the classification of the original attributes of decision table. The rough set theory provides the tool to deal with this issue. The concept of reduct is a tool that can be used to achieve that goal. There are many reducts that can be discovered at the analysis of the decision table. We are mostly interested in the best reduct. The general problem of finding all reducts is NP-hard [17], but in most cases it is usually not necessary to find all reducts. Here we follow the criteria that the best reduct is the one with minimum number of attributes and that if there are two or more reducts with the same number of attributes, then the reduct with the least number of combination of its attributes is selected [5]. So we offer an algorithm based on computing of core from decision table.

ALGORITHM :-

1. Accept a decision table $T = (U,C,D,V)$.
2. Compute the quality of classification of decision attributes on the set of condition attributes $\gamma_C = \gamma_C (U|D)$ {if $\gamma_C = 1$ then *D* is fully dependent on *C*}.
3. Compute the core of condition attributes based on the classification of decision attributes $\text{CORE}_D(C) = \cap \text{RED}_D(C)$.
4. $BR = \text{CORE}_D(C)$ {initialize the best reduct by the core}.
5. If $\gamma_{BR}(U|D) = \gamma_C$, then stop { the core represents the best reduct}.
6. $C = C - BR$.
7. Repeat: For every attribute $c \in C$, compute significance factor of attribute *c* when is added to *BR* $\text{SGF}(c,BR,D) = \gamma_{BR\cup\{c\}}(U|D) - \gamma_{BR}(U|D)$.
8. Select attribute $c \in C$ of maximum $\text{SGF}(c,BR,D)$.
9. If there are several attributes c_i (*i*=1 to *m*) with the maximum value of $\text{SGF}(c,BR,D)$, select the attribute c_j which has the least number of combination value with *BR, i.e.* with minimum of card $(U| BR+\{c_i\})$.
10. $BR = BR \cup \{c_j\}$ {add selected attribute to the set of best reduct}.
11. $C = C - \cup\{c_i\}$ (*i*=1 to *m*) {subtract all attributes of maximum significance factor from the set of condition attributes}.
12. If $\gamma_{BR}(U|D) = \gamma_C$, then stop, else go to step 7.
13. The best reduct is the set *BR*.

4.2.4 Binarization of Input Data

The method of presenting the input data to neural network may affect the performance of learning, and hence an undesired result may be produced. Searching for the best method is important when we are dealing with a complex or sophisticated network. The classification of variables helps us to present the input data to neural network in a suitable way.

5.0 ILLUSTRATION EXAMPLE

To illustrate the use of pre-processing rough engine, let us consider an example of patient records. Eight example cases given by a medical doctor create a training set presented in Table 1. Each patient case consists of two nominal input variables 'History' and 'Pain', two interval input variables 'Fever' and 'Weight loss', and one output variable 'Decision' which determines whether the patient is healthy or not.

To formulate a decision table from this set, the two interval variables will be scaled to form two ordered nominal attributes, and one of the two nominal variables 'History' must be split into three binary attributes. The final decision table is presented in Table 2.

To obtain a minimal set i.e. the best reduct, of condition attributes from the decision table, we apply the given algorithm as follows:

1. The quality of classification is $\gamma_C = 1$. It means that using all condition attributes we can perfectly approximate the decision.
2. The core is the set $\{ c_2 \}$ with $\gamma_{\text{CORE}} = 0.75 < \gamma_C$, hence it is not a reduct. Thus, we put $BR = \{c_2\}$ and continue with the remaining attributes of *C*.
3. Computed significance factors of attributes of *C* are presented in Table 3.
4. There are three attributes ($c_{1\text{-}1}$, $c_{1\text{-}3}$ and c_4) with maximum value of $\text{SGF}(c,BR,D)$, hence to select one of them we need to know the number of combination of their values with *BR:*

	$c_{1\text{-}1}$	$c_{1\text{-}3}$	c_4
No. of combination Values with BR	6	5	8

5. Attribute $c_{1\text{-}3}$ has the minimum number of combination value with *BR*, and when it is added to

BR it yields $\gamma_{BR} = 1$, hence the best reduct is the set $BR = \{c_{1\text{-}3}, c_2\}$.

Finally, we select a suitable way to present the reduced training examples to neural network. The proposed form of final input to neural network is given in Table 4.

Table 1: A Training Set of Patient Records

Patient	History	Fever	Pain	Weight loss	Decision
1	exist	40	Yes	2.3	ill
2	exist	37	Yes	5.9	healthy
3	none	36	Yes	4.7	healthy
4	relative	42	Yes	7.4	ill
5	relative	39	Yes	6.6	healthy
6	relative	41	Yes	2.3	ill
7	exist	42	No	3.7	ill
8	none	37	No	1.6	healthy

Table 2: The Decision Table from Training Set

Pat#	Condition Attributes						Decision
	$C_{1\text{-}1}$ **'exist'**	$c_{1\text{-}2}$ **'none'**	$c_{1\text{-}3}$ **'relative'**	c_2	c_3	c_4	
1	Yes	no	no	High	yes	Low	ill
2	Yes	no	no	Normal	yes	High	healthy
3	No	yes	no	Normal	yes	Medium	healthy
4	No	no	yes	Very high	yes	High	ill
5	No	no	yes	High	yes	High	healthy
6	No	no	yes	Very high	yes	Low	ill
7	Yes	no	no	very high	no	Medium	ill
8	No	yes	no	normal	no	Low	healthy

Table 3: Significance Factors of the Set of Condition Attributes

Attribute *c*	$\gamma_{BR\cup\{c\}}$	**SGF(*c,BR,D*)**
$c_{1\text{-}1}$	1	0.25
$c_{1\text{-}2}$	0.75	0.00
$c_{1\text{-}3}$	1	0.25
c_3	0.75	0.00
c_4	1	0.25

Table 4: The Final Input to Neural Network Based on Minimal Set

Inp#	Input neurons		Output neurons	
	History (u_1)	**Fever (u_2)**	**Ill patient (o_1)**	**Healthy patient (o_2)**
1	0	0.0	1	0
2	0	-0.5	0	1
3	0	-0.5	0	1
4	1	+0.5	1	0
5	1	0.0	0	1
6	1	+0.5	1	0
7	0	+0.5	1	0
8	0	-0.5	0	1

6.0 CONCLUSION

Clearly, the construction of knowledge base for expert systems is not a straightforward task. Neural network, after working on training examples, can bring us to the desired goal. In our presentation, we have attempted to demonstrate a methodology for direct acquisition of knowledge base from neural network, based on rough sets theory.

Rough set gives a minimal representation, so that a reduction of input to the network can be achieved which results in accelerating the training time. Rough sets theory presents a promising approach to knowledge-acquisition problem for expert systems in general and specially for neural expert systems. These methods are suitable for classifying pattern from abundant and noisy data.

REFERENCES

[1] M. Beri and G. Trotta, "An Approach to Develop Classification Expert Systems with Neural Network", in N. Gercone and F. Gardin (Eds), *Computational Intelligence III,* Elsevier science, Netherlands, 1991.

[2] S. I. Gallant, "Connectionist Expert Systems", C*ommunication of the ACM,* Vol. 31, No. 2, 1988, pp. 152-169.

[3] L. Glerfeld, "Methodology of Simplification and Interpretation of Backpropagation Based Neural Network Models", *Expert Systems with Applications,* Vol. 10, No. 1, 1996, pp. 37-54.

[4] Y. Hayashi, "A Neural Expert System Using Fuzzy Teaching Input", In *Proceedings of 92 IEEE International Conference on Fuzzy System FUZZ-IEEE,* Piscataway, NJ, 1992, pp. 485-491.

[5] X. Hu and N. Cercone, "Learning in Relational Database: A Rough Set Approach", *Computational Intelligence,* Vol. 11, No. 2, 1995, pp. 323-338.

[6] J. Jelonek, et al. "Rough Set Reduction of Attributes and Their Domains for Neural Networks". *Computational Intelligence,* Vol. 11 No. 2, 1995, pp. 339-347.

[7] R. Mahmod and M. E. Yahia, "Fusion Method of Rough Sets Theory in Neural Expert Systems", *Information Sciences - International Journal,* (submitted).

[8] R. Mahmod and M. E. Yahia, *Rough Sets Decision Making for Uncertain Data*, Department of Computer Science, Universiti Putra Malaysia, Technical report no. TR018, 1997.

[9] L Medsker and J. Liebowitz, *Design and Development of Expert Systems and Neural Network, M*acmillan Publishing Company, New York, 1994.

[10] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data,* Kluwer, Netherlands, 1991.

[11] Z. Pawlak, et al. "Rough Sets", *Communication of the ACM,* Vol. 38, No. 11, 1995, pp. 89-95.

[12] R. Slowiniski, (ed) *Intelligent Decision Support. Handbook of Applications and Advances of Rough Sets Theory,* Kluwer, Dordrecht, Netherlands, 1992.

[13] M. E. Yahia and R. Mahmod, "Fusion Method of Rough Set in Neural Expert System". *Fifth International Workshop of Rough Sets and Soft Computing (RSSC'97) in Proceedings of Joint Conference of Information Sciences (JCIS'97)*, Vol. 3, NC, USA, 1997, pp. 263-264.

[14] M. E. Yahia and R. Mahmod, *The Emergence of Rough Sets Theory*, Department of Computer Science, Universiti Putra Malaysia, TR001-1997.

[15] M. E. Yahia and R. Mahmod, "Rough Set Based Neural Expert Systems", *Proceedings of Information Technology Colloquium (INTEC'97),* Department of Computer Science, Universiti Putra Malaysia, 1997, pp. A6-A10.

[16] R. Yasdi, "Combining Rough Sets Learning and Neural Network Learning Method to Deal with Uncertain and Imprecise Information", *Neurocomputing,* Vol. 7, No. 1, 1995, pp. 61-64.

[17] W. Zirako, "The Discovery, Analysis, and Representation of Data Dependencies in Databases", In G. Piatetsky-Shapiro, and W. Frawley, (eds), *Knowledge Discovery in Databases,* AAAI/MIT Press, Cambridge, MA 1990, pp. 31-54.

BIOGRAPHY

Ramlan Mahmod is a senior lecturer at Computer Science Department, University Putra Malaysia. He received his B.Sc from Western Michigan University, USA and his M.Sc from Central Michigan University, USA. He obtained his Ph.D in Artificial Intelligence and Neural Networks from Bradford University, U.K. His research interest is in the field of artificial intelligence, neural networks, data mining and security in computing.

Moawia E. Yahia is a lecturer at Faculty of Mathematical Sciences, University of Khartoum, Sudan. He received his B.Sc (honor) and M.Sc in computer science from the same Faculty. Currently, he is enrolling as a Ph.D student at Department of Computer Science, Universiti Putra Malaysia. His research interest is in the field of artificial intelligence including expert systems, neural networks, knowledge discovery, machine learning and handling of uncertainty by rough set.