

An Intelligent Data Mining Approach Using Neuro-Rough Hybridization to Discover Hidden Knowledge from Information Systems

REZA SABZEVARI AND GH. A. MONTAZER*

Department of Mechatronics Engineering

Islamic Azad University of Qazvin

Qazvin, Iran

Student Member of Young Researchers' Club (YRC)

**Department of Information Engineering*

Tarbiat Modares University

P.O. Box: 14115-179, Tehran, Iran

E-mail: montazer@modares.ac.ir

In this paper we discuss on the necessity of applying data mining operators on information systems containing a set of variables which describe the characteristics and behaviors of a specific system and could be exploited in approximating system's functionality. For the problem of function approximation, we developed a new approach combining two intelligent methods. At first we used an algorithm based on the notions of rough set theory as a preprocessor to our information system. Afterward an artificial neural network is employed as a function approximator to obtain values for decision attributes of information system while values of condition ones are passed to the network. This method has been applied to a real problem of approximating values for two hydraulic-geotechnical control variables of rubble mound breakwaters, and the results have been discussed.

Keywords: data mining, information systems, modeling, function approximation, rough sets theory, artificial neural networks

1. INTRODUCTION

The amount of data stored in databases continues to grow fast. Intuitively, this large amount of stored data contains valuable hidden knowledge, which could be used to improve the performance of decision-making process [1]. There is a suspicion that there might be nuggets of useful information hiding in the masses of unanalyzed or under-analyzed data, and therefore semi-automatic methods for locating interesting information from data would be useful [2]. This fact leads to feel the necessity of intelligent data analysis, a field called Knowledge Discovery which is always compounded with the term of Data Mining [3]. Knowledge Discovery (KD) aims to extract high-level knowledge or create a high-level description from real-world data sets [2]. Data mining is the core of Knowledge Discovery. The process of knowledge discovery employs several preprocessing methods, to facilitate data mining algorithms, and also post-processing methods to refine and improve the discovered knowledge [4]. Data mining is a particular

Received August 25, 2006; revised January 15, 2007; accepted February 14, 2007.

Communicated by Suh-Yin Lee.

* Corresponding author: Gh. A. Montazer.

step in this process involving the application of specific algorithms for extracting patterns (models) from data. Additional steps in the KD process are such as data preparation, data selection, data cleaning, incorporation of appropriate prior knowledge and proper interpretation of the results of mining, ensure that useful knowledge is derived from the data [5].

Soft computing methodologies, involving fuzzy sets, neural networks, genetic algorithms and rough sets are most widely applied in the data mining phase of the overall KD process. Fuzzy sets provide a natural framework for the process to deal with uncertainty [6]. Neural networks [7] and rough sets [8] are widely used for classification and rule generation. Genetic algorithms are involved in various optimization and search processes, like query optimization [9] and template selection [10]. Other approaches like Case Based Reasoning [11] and Decision Trees [12] are also widely used to solve data mining problems. The power of data mining including its problem solving capabilities, performance and utilization depends on developing generic and also problem specific algorithms employing methods from different fields of science.

In some cases, an information system may contain some relations between attributes which disturb the process of extracting rules for approximating values of decision attributes. In these cases, discovering such relations and omitting them from the information system will help the process of decision making.

In the following part of this paper, section 2, we present a brief description for basic concepts including *data mining*, *rough sets theory* and *artificial neural networks*. Section 3 describes the situation which subjected problem occurs and discusses on the reasons cause these problems. Also some experimental witnesses show that in the case of occurrence of such problems we can not train a neural network to work as a function approximator. The next part, section 4, is dedicated to a data mining algorithm based on the notions of rough sets theory to perform a preprocessing task on database. And finally in section 5 we have presented and discussed on some architectures for the neural network used to approximate the functionality of information system with reduced set of condition attributes.

2. BASIC CONCEPTS

2.1 Data Mining

Knowledge discovery is defined as the process of identifying valid, novel, potentially useful and ultimately understandable patterns in data [13]. *Data* is a set of facts F , and a *pattern* is an expression E in a *language* L describing the facts in a subset F_E of F [14]. A parameter measures the validity of discovered patterns, which is a function C mapping expressions in L to a partially or totally ordered measure space M_C and called as certainty measure. For an expression E in L about a subset $F_E \subset F$ can be assigned a certainty measure $c = C(E, F)$. Novelty of patterns can be measured by a function $N(E, F)$ with respect to changes in data or knowledge. Patterns should potentially lead to some useful actions, which this term is measured by a utilization function $u = U(E, F)$ mapping expressions E in L to a partially or totally ordered measure space M_U . As the goal of knowledge discovery is to make patterns understandable to humans, this feature is measured by a function $s = S(E, F)$ mapping expressions in L to a partially or totally ordered

measure space M_S [15]. *Interestingness* of a pattern is a combination of validity, novelty, usefulness, and understandability, and can be expressed as $i = I(E, F, C, N, U, S)$ which maps expressions in L to a measure space M_I [14]. A pattern $E \in L$ is called knowledge if for some user-specified threshold $i \in M_I$ where $I(E, F, C, N, U, S) > i$, one can select some other thresholds like $c \in M_C$, $s \in M_S$, and $u \in M_U$, and call a pattern E knowledge, if and only if $C(E, F) > c$ and $S(E, F) > s$, and $U(E, F) > u$.

In simple words, knowledge discovery refers to the overall process of turning low-level data into high-level knowledge. As mentioned before, an important step in this process is data mining. Data mining is an interdisciplinary field with a general goal of predicting outcomes and uncovering relationships in data [2]. Data mining involves fitting models to or determining patterns from observed data. The fitted models play the role of inferred knowledge. Deciding whether the model reflects useful knowledge or not, is a part of the overall knowledge discovery process for which subjective human judgment is usually required.

The more common model functions in current data mining tasks could be specified as Classification [16], Regression [17], Clustering [18], Rule Generation [19], Discovering Association Rules [20], Summarization [21], Dependency Modeling [22] and Sequence Analysis [23].

2.2 Rough Sets Theory

Rough Sets Theory (RST) was introduced by Pawlak in 1982 [24]. It has attracted the attention of many researchers and practitioners all over the world, who contributed to its development and application during the last decade [25-28]. This theory describes dependencies between attributes, to evaluate significance of attributes, and to deal with inconsistent data.

The Rough Sets philosophy is founded on the assumption that with every objects of the universe of discourse we associate some information (*i.e.* data knowledge). Objects characterized by the same information are indiscernible in view of the available information about them. The indiscernibility relation generated in this way is the mathematical basis for the Rough Sets Theory [25].

By an information system S , $S = \{U, Q, V, f\}$, where $U = \{x_1, x_2, \dots, x_n\}$ is a finite set of objects and $Q = \{q_1, q_2, \dots, q_m\}$ is a finite set of attributes which this set is further classified into two disjoint subsets: condition attributes C and decision attributes D where $Q = C \cup D$. We define V_p as the domain of values for attribute p where:

$$V = \cup \{V_p \mid p \in Q\} \quad (1)$$

and $f: U \times Q \rightarrow V$ is a total function such that $f(x_i, q) \in V_q$ for every $q \in Q$ and $x_i \in U$.

Let $E \subseteq Q$ and $x, y \in U$. We say that x, y are indiscernible by the set of attributes E in S , if $f(x, q) = f(y, q)$ for every $q \in E$. Thus every $E \subseteq Q$ generates a binary relation denoted by I_E . Obviously, I_E is an equivalence relation for any E . Equivalence classes of the relation I_E are called *E-elementary* sets in S and $I_E(x)$ denotes the *E-elementary* set containing the object $x \in U$, hence [26]:

$$I_E(x) = \{y \in U \mid \forall q \in E, f(x, q) = f(y, q)\}. \quad (2)$$

Now let $E \subseteq Q$ and $Y \subseteq U$. The E -Lower approximation of Y , denoted by $\underline{E}Y$, and the E -Upper approximation of Y , denoted by $\overline{E}Y$ are defined respectively as:

- $\overline{E}Y = \cup \{x \in U \mid I_E(x) \cap Y \neq \Phi\}$,
- $\underline{E}Y = \cup \{x \in U \mid I_E(x) \subseteq Y\}$,

and the E -boundary of set Y is defined as $BND(Y) = \overline{E}Y - \underline{E}Y$.

Set $\underline{E}Y$ is the set of all elements of U , which can be certainly classified as elementary of Y , employing the set of attribute E . Set $\overline{E}Y$ is the set of elements of U , which can be possibly classified as elements of Y using the set of attribute E . The set $BND(Y)$ is the set of elements, which cannot be certainly classified as elements of Y , using the set of attribute E . Rough approximations have been shown in Fig. 1.

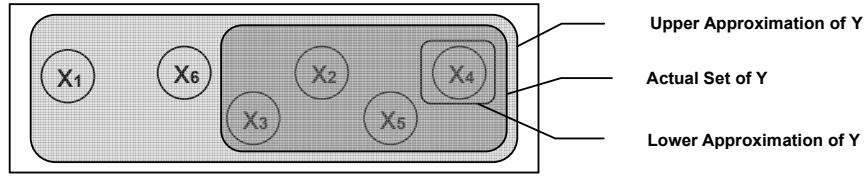


Fig. 1. Representation of lower and upper approximations.

Table 1. An information system.

U	Q				
	C				D
	c_1	c_2	c_3	c_4	d
x_1	A	Yes	10	-5	Low
x_2	B	Yes	10	-5	High
x_3	A	Yes	10	7	High
x_4	A	Yes	50	7	High
x_5	B	No	10	-5	High
x_6	A	No	10	-5	Low

Consider the example of an information table which has been shown in Table 1. In this knowledge system, we have:

- The set of objects $U = \{x_1, x_2, x_3, x_4, x_5, x_6\}$,
- The set of condition attributes $C = \{c_1, c_2, c_3, c_4\}$,
- The set of decision attributes $D = \{d\}$,
- Particular sets of attribute values given as: $V_{c_1} = \{A, B\}$, $V_{c_2} = \{\text{Yes}, \text{No}\}$, $V_{c_3} = \{10, 50\}$, $V_{c_4} = \{-5, 7\}$ and $V_d = \{\text{Low}, \text{High}\}$.
- The set of attribute values is given as: $= \{V_{c_1}, V_{c_2}, V_{c_3}, V_{c_4}, V_d\}$.

Let $Y = \{x \in U \mid d(x) = \text{High}\} = \{x_2, x_3, x_4, x_5\}$ and $E = \{c_2, c_3\}$, then:

- $I_E = \{\{x_1, x_2, x_3\}, \{x_4\}, \{x_5, x_6\}\}$,

- $I_E(x_1) = \{x_1, x_2, x_3\}$,
- $I_E(x_5) = \{x_5, x_6\}$,
- $\overline{E}Y = \{x_1, x_2, x_3, x_4, x_5, x_6\}$,
- $\underline{E}Y = \{x_4\}$,
- $BND(Y) = \{x_1, x_2, x_3, x_5, x_6\}$.

2.3 Artificial Neural Networks

Artificial Neural Network (ANN) is a loosely modeled system based on the human brain, which mimics biological information processing mechanisms. It is an inherently multiprocessor-friendly architecture which has ability to account for any functional dependency. The network discovers (learns or models) the nature of the dependency without needing to be prompted. A neural network typically consists of many simple processing units, also called as electronic processing elements, which are wired together in a complex communication network, without any processing unit following a logical sequence of rules. Behavior of a trained ANN depends on the weights, which are also referred to as strengths of the connections between the processing elements [29].

Common applications of neural networks can be grouped in the categories such: Clustering [30], Classification [31], Pattern Recognition [32], Function Approximation [33], Prediction [34] and Dynamical Systems [35].

A neural network may be composed of several layers of identical processing elements. The layers of a multilayer network play different roles. A layer includes a combination of the weights, the multiplication and summing operations, the bias b , and the transfer function f . If a particular layer contains R units, the outputs of that layer can be thought of as an R -dimensional vector, $p = [p_1, p_2, \dots, p_R]^T$, where the T superscript means *transpose*. If the R -dimensional output vector P provides the input values to each unit in an S -dimensional layer, each unit in the S -dimensional layer will have R weights associated with the connections from the previous layer. Thus, there are S weight vectors associated with this layer; each weight vector is a R -dimensional one which corresponds to each of the S units. The weight vector of the i th unit can be written as $W_i = (w_{i1}, w_{i2}, \dots, w_{iR})^T$ [36].

The net input to the i th unit can be written in terms of the inner product of the input vector and the weight vector. For vectors of equal dimensions, the inner product is defined as the sum of the products of the corresponding components of the two vectors. If the neuron has a bias b which is summed with the weighted inputs to form the net input n , this sum, n , is the argument of the transfer function f . So, we have

$$n_i = \sum_{j=1}^R p_j w_{ij} + b_i \quad (3)$$

and

$$a_i = f(W_i P + b_i) \quad (4)$$

where n is the number of connections to the i th unit [36]. As it is apparent in equations above, each unit generates output while passing input vector through a transfer function.

3. PROBLEM STATEMENT

Unstructured data gathering will cause inclusion of dummy data in database, which identifying these dummy data is a very complicated and expensive task. This problem almost happens in data acquisition process and appears as invalid instances in information system or redundant condition attributes. Invalid instances are generated when some measurement errors occur by the reason of some uncontrollable factors in measurement environment, environmental difficulties in measurement process, instrumental errors or human mistakes. Having no idea about affecting parameters in a phenomenon or improper composition in the set of condition attributes may cause redundancy in the set of condition attributes. This situation will happen when we could not have a real good perception about the target system. This problem arises when we deal with large scale or complicated systems where analyzing such systems is a vital and expensive task [37].

Our story began when we dealt with an information system containing the hydraulic-geotechnical control parameters of rubble mound breakwaters. In this problem we want to approximate the value of two factors considering 13 variables appeared as condition attributes in the information system. To solve this problem we tried to model the system using a neural network as a function approximator. All we have from the system is only a database with 1,440 samples which some of those have been shown in Table 2. These samples represented as rows of an information system consist of 15 real values;

Table 2. Some of hydraulic database information.

No	Condition Attributes													Decision Attributes	
	H_s	T_s	ρ_s	ρ_w	P	N	$\text{Cot}(\alpha)$	S	d_p	d_{\min}	Φ_{BRE}	Φ_b	ρ_{sb}	D_{n50}	SF
1	3	8	2500	1025	0.4	5000	2	2	8	5	34	23	18	1.3744	1.234
2	3	8	2500	1025	0.4	5000	2	2	8	5	34	23	20	1.3744	1.291
3	3	8	2500	1025	0.4	5000	2	2	8	5	34	26	18	1.3744	1.331
4	3	8	2500	1025	0.4	5000	2	2	8	5	34	26	20	1.3744	1.378
5	3	8	2500	1025	0.4	5000	2	2	8	5	40	23	18	1.3744	1.331
6	3	8	2500	1025	0.4	5000	2	2	8	5	40	23	20	1.3744	1.404
7	3	8	2500	1025	0.4	5000	2	2	8	5	40	26	18	1.3744	1.462
8	3	8	2500	1025	0.4	5000	2	2	8	5	40	26	20	1.3744	1.528
9	3	8	2500	1025	0.4	5000	2	5	8	5	34	23	18	1.1442	1.232
10	3	8	2500	1025	0.4	5000	2	5	8	5	34	23	20	1.1442	1.291
11	3	8	2500	1025	0.4	5000	2	5	8	5	34	26	18	1.1442	1.337
12	3	8	2500	1025	0.4	5000	2	5	8	5	34	26	20	1.1442	1.383
13	3	8	2500	1025	0.4	5000	2	5	8	5	40	23	18	1.1442	1.335
14	3	8	2500	1025	0.4	5000	2	5	8	5	40	23	20	1.1442	1.409
15	3	8	2500	1025	0.4	5000	2	5	8	5	40	26	18	1.1442	1.463
16	3	8	2500	1025	0.4	5000	2	5	8	5	40	26	20	1.1442	1.533
17	3	8	2500	1025	0.4	5000	3	2	8	5	34	23	18	1.1222	1.444
18	3	8	2500	1025	0.4	5000	3	2	8	5	34	23	20	1.1222	1.525
19	3	8	2500	1025	0.4	5000	3	2	8	5	34	26	18	1.1222	1.592
20	3	8	2500	1025	0.4	5000	3	2	8	5	34	26	20	1.1222	1.669
21	3	8	2500	1025	0.4	5000	3	2	8	5	40	23	18	1.1222	1.54
22	3	8	2500	1025	0.4	5000	3	2	8	5	40	23	20	1.1222	1.635
23	3	8	2500	1025	0.4	5000	3	2	8	5	40	26	18	1.1222	1.698
24	3	8	2500	1025	0.4	5000	3	2	8	5	40	26	20	1.1222	1.802
25	3	8	2500	1025	0.4	5000	3	5	8	5	34	23	18	0.9343	1.447
26	3	8	2500	1025	0.4	5000	3	5	8	5	34	23	20	0.9343	1.517
27	3	8	2500	1025	0.4	5000	3	5	8	5	34	26	18	0.9343	1.579
28	3	8	2500	1025	0.4	5000	3	5	8	5	34	26	20	0.9343	1.656
29	3	8	2500	1025	0.4	5000	3	5	8	5	40	23	18	0.9343	1.572
30	3	8	2500	1025	0.4	5000	3	5	8	5	40	23	20	0.9343	1.647

which 13 of them belongs to initial variables of breakwater, used as network inputs, and the other 2 ones are decision parameters in the process of breakwater's design process affected by the initial variables, used as network targets.

To model the behavior of this information system we picked up a feed-forward back-propagation network [38], as we know these networks are proper for function approximation tasks [39]. Table 3 represents the architecture of networks and their training parameters which also have been supported with results obtained by training these networks. As it is well known, using linear and sigmoid transfer functions any desired function could be approximated [40], so we used combinations of these two transfer functions for different layers of networks. We randomly picked 960 samples, about 66.6% of database entries, and used them for training the networks. And the other left 480 samples, about 33.3% of database entries, are used to test our trained networks.

As each input vector is applied to the network, the network outputs are compared with the target ones. Simply speaking, the error is calculated as the difference between the target output and the corresponding output came from network. We want to minimize the average sum of these errors. For this reason we used MSE (Mean Squared Errors) as the performance function to measure networks' performance, according to the mean of squared errors, while training. Through the train process weights and biases of the network are iteratively adjusted to minimize the network performance function. The MSE function operates as below:

$$MSE = \frac{1}{N} \sum_{i=1}^N e_i^2 = \frac{1}{N} \sum_{i=1}^N (t_i - a_i)^2 \quad (5)$$

where N is the number of instances whose error is desired to be calculated, t is the target for corresponding input and a is network response to that input.

As it is obvious in Table 3, using the TRAINGD (Gradient Descent with momentum and adaptive learning rate back-propagation) as the network train function will result late convergence for network performance. This fact is apparent as the values for train epochs are larger in comparison with the cases when TRAINRP (Resilient Back-Propagation) is used as train function.

Examining all these architectures, which have been brought in Table 3, we did not reach to the goal of approximating the function of information system. After a few number of epochs all networks' outputs stayed on a fixed number and the train process stops. This fact shows that there is a hidden relation among input parameters which do not allow the network to be trained. Fig. 2 shows a sample of this fact.

The last column of Table 3 shows the performance reached by training networks. In this column we can see that the mean squared errors of network outputs did not goes below 0.133559, which this performance dose not satisfy the accuracy needed by the system. In all architectures stated in Table 3 the network returns a fixed number to arbitrary inputs, which means that these networks have not trained successfully. This fact is apparent in sample graphs depicted in Fig. 2 while the outputs for all cases lies on a vertical line, whereas they should be fitted ideally on $x = y$ line when we plot the outputs in the space of "Network Outputs – Desired Outputs". In up coming sections, we show that how preprocessing operations can solve this problem and give accurate results as appeared in Table 4.

Table 3. Neural network architectures using the original inputs.

Network	No. of Inputs	Input Transfer Function	No. of Outputs	Output Transfer Function	No. of Hidden Layers	Hidden Layers' Transfer Functions	No. of Hidden Layers' Neurons	Train Function	Performance Function	Train Epochs	Network Performance
Network1	13	TANSIG	2	PURELIN	1	TANSIG	18	TRAINGDX	MSE	10000	0.143467
Network2	13	TANSIG	2	PURELIN	1	TANSIG	18	TRAINRP	MSE	49	0.138563
Network3	13	PURELIN	2	PURELIN	1	TANSIG	26	TRAINRP	MSE	45	0.138563
Network4	13	PURELIN	2	PURELIN	1	TANSIG	26	TRAINGDX	MSE	179	0.138563
Network5	13	PURELIN	2	PURELIN	2	TANSIG	27	TRAINGDX	MSE	6114	0.133559
Network6	13	PURELIN	2	PURELIN	2	TANSIG	26	TRAINRP	MSE	53	0.138563
Network7	13	PURELIN	2	PURELIN	3	TANSIG	26	TRAINRP	MSE	36	0.138563
Network8	13	PURELIN	2	PURELIN	2	TANSIG	27	TRAINRP	MSE	39	0.138563
Network9	13	PURELIN	2	PURELIN	2	TANSIG	39	TRAINRP	MSE	49	0.138563
Network10	13	PURELIN	2	PURELIN	3	TANSIG	39	TRAINRP	MSE	36	0.138563
Network11	13	PURELIN	2	PURELIN	3	TANSIG	39	TRAINRP	MSE	43	0.138563

Table 4. Neural network architectures using reduced inputs.

Network	No. of Inputs	Input Transfer Function	No. of Outputs	Output Transfer Function	No. of Hidden Layers	Hidden Layers' Transfer Functions	No. of Hidden Layers' Neurons	Train Function	Performance Function	Train Epochs	Network Performance	LCC for the 1 st Output	LCC for the 2 nd Output
Network12	9	TANSIG	2	PURELIN	1	TANSIG	18	TRAINRP	MSE	10000	0.00172518	0.99305	0.88121
Network13	9	TANSIG	2	PURELIN	1	LOGSIG	18	TRAINRP	MSE	10000	0.00129189	0.99679	0.8937
Network14	9	PURELIN	2	PURELIN	1	TANSIG	26	TRAINRP	MSE	10000	0.00128981	0.99738	0.90469
Network15	9	PURELIN	2	PURELIN	1	TANSIG	26	TRAINGDX	MSE	10000	0.0269706	0.88239	0.35461
Network16	9	PURELIN	2	PURELIN	2	TANSIG	9	TRAINRP	MSE	10000	0.00158017	0.99597	0.89095
Network17	9	PURELIN	2	PURELIN	2	TANSIG	9	TRAINRP	MSE	10000	0.000935462	0.99832	0.92201
Network18	9	PURELIN	2	PURELIN	2	TANSIG	9	TRAINRP	MSE	10000	0.00136495	0.99694	0.90218
Network19	9	PURELIN	2	PURELIN	2	TANSIG	27	TRAINRP	MSE	10000	0.000496624	0.99874	0.94661

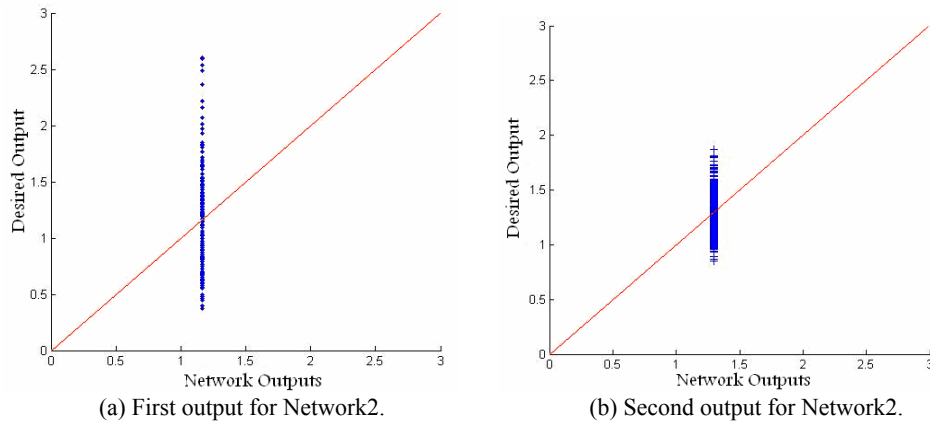


Fig. 2. Sample network results' using all condition attributes as network inputs.

In the process of training back-propagation networks some failures may occur that prevent network to be trained. These failures generally arise from two sources: *network paralysis* [41] and *local minima* [42]. In the first case, as the network trains, the weights may be adjusted to very large values. The total input of a hidden unit or output unit can therefore reach very high values (either positive or negative level), and because of the sigmoid activation function the unit will have an activation very close to zero or very close to one. So, the weight adjustments will be close to zero, and the training process can come to a virtual standstill. If a network stops learning trapped in a local minimum before reaching an acceptable solution, a change in the number of hidden nodes [29] or in the learning parameters will often fix the problem; or we can simply start over with a different set of initial weights [36]. Also probabilistic methods can help to avoid this trap, but they tend to be slow.

4. SOLUTION ARCHITECTURE

To solve the problem mentioned in the previous section our approach combines two intelligent methods to deal with information systems having some hidden relations in their condition attributes, which these relations will disturb the process of identifying and modeling the system behavior. As described in previous section, feed-forward back-propagation neural networks with different configurations in their architectures could not be trained to deal with our information system. This shows that the system contains some hidden relation in its condition attributes or some invalid instances. So, the only way exists is to eliminate such relations and the possible invalid instances. To do so and to avoid such problems we used the rough set theory to keep away from redundancies and also to reduce the input parameters and provide the affecting ones for our desired application. In the next step we try to train a feed-forward back-propagation neural network to approximate the system's functionality, employing the new set of input parameters resulted from the rough set based algorithm.

The architecture of our approach is described in Fig. 3. In this architecture, input parameters will pass through a rough analysis system which will act as a data mining

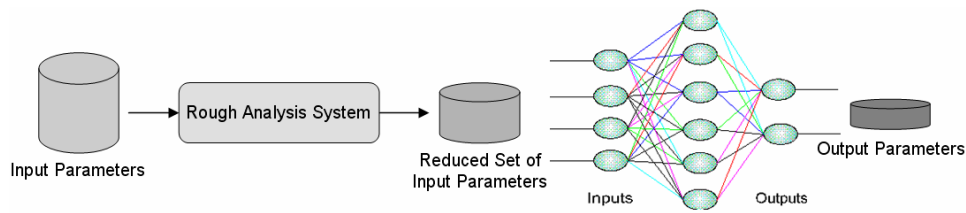


Fig. 3. The solution architecture used in our approach.

core for our system. Outputs of this system are appeared as a new database with some reductions in rows and columns. This means that redundancies in both attributes and entities of information system are discovered and omitted from the database. This block-set also recognizes condition attributes strongly affecting each decision one. After this process, new set of condition attributes will be passed through an artificial neural network and the corresponding decision attributes will be appeared on network outputs.

5. DATA MINING ALGORITHM BASED ON ROUGH SETS THEORY TO DISCOVER INTERNAL RELATIONS

Our knowledge system, including the empirical relation among condition attributes and decision ones, is partially presented in Table 2. These data have been extracted from experimental measurements using MSTAB and BREAKWATER software [8]. Basic problems blinking in this database are the vagueness of decision variables and also the uncertain relation between object-attribute values and their corresponding results in decision columns. It is obvious that the larger size of database, the more difficulties in decision processes [28].

Many algorithms have been developed to reduce the conditions and have been used in many problems [8, 43]. In this paper we have presented a modified procedure, collecting useful parts of previous approaches. In our approach, employing the rough sets theory we used the algorithm described below to reduce the size of foregoing information system and also discovering the hidden knowledge lies on its entries to ease the process of function approximation. Basic steps in data analysis will be described in the following as steps 1 to 4. In these steps for the case of simplicity we will describe our algorithm on a simplified information system, such came in Table 1, and then will apply it on our real information system, rubble mound breakwater control parameters, at the end of each step [27].

Step 1: The first step of this algorithm is to eliminate unnecessary input variables from the table. This task can be accomplished eliminating each attribute and verifying if lower approximation of the resulting table is equal to lower approximation of the original one. In Table 1 we have $\underline{CU} = \{x_1, x_2, x_3, x_4, x_5, x_6\}$. If we remove c_1 then $P_1 = \{c_2, c_3, c_4\}$ And $P_1U = \{x_3, x_4\}$. Therefore because of $\underline{CU} \neq P_1U$, we cannot wipe out c_1 . If we examine removing c_2 and c_3 , we can show that $P_2U = \underline{CU}$ and $P_3U = \underline{CU}$. Therefore, we can eliminate c_2 and c_3 . But examine eliminating c_4 , we will have $P_4U \neq \underline{CU}$, so c_4 could not be deleted. Hence, using this step the Table 1 can be reduced to Table 5.

If we want to apply this step on our real information system, Table 2, we have to check the possibility of eliminating every condition attributes under the condition described above. Doing so we find that for $D = \{D_{n50}\}$, 7 attributes ($\rho_w, P, N, d_p, d_{\min}, \Phi_{BRE}, \Phi_b$) and for $D = \{SF\}$, 4 attributes (ρ_w, P, N, d_p) could be omitted.

Table 5. Reduced information system by step 1.

U	\underline{Q}		
	C		D
	c_1	c_4	d
x_1	A	- 5	Low
x_2	B	- 5	High
x_3	A	7	High
x_4	A	7	High
x_5	B	- 5	High

Table 6. Reduced information system by step 2.

U	\underline{Q}		
	C		D
	c_1	c_4	d
x_1	A	- 5	Low
x_2	B	- 5	High
x_3	A	7	High

Step 2: The second step is to remove repeated objects in Table 5. Using this step the resulting table is shown in Table 6. Applying this step on results achieved by the previous step for breakwater information system, we find out that we can remove 237 objects for $D = \{D_{n50}\}$, and 634 objects for $D = \{SF\}$.

Step 3: The third step is to remove unnecessary values of attributes for each decision rule. This is known as finding the *core* values. This task can be accomplished eliminating each condition attribute value and verifying whether the table is consistent. A table is consistent if for every combination of condition attributes presented in Table 6, a unique value for the decision attribute could be achieved. In Table 6, If we eliminate the value of $(x_1, c_1) = A$, then the table become inconsistent. Therefore, we cannot eliminate this value. But we can eliminate the values of $(x_3, c_1) = A$ and $(x_2, c_4) = - 5$. Applying this step, the resulting table has been shown in Table 7. As the third step, on the results gained applying previous step to our information system, we should eliminate values of such condition attributes that will not affect on table consistency.

Step 4: The next step is to eliminate objects that are repeated in the resulting table which in this example, Table 7, there is no object that has been repeated.

Table 7. Reduced information system by step 3.

U	\underline{Q}		
	C		D
	c_1	c_4	d
x_1	A	- 5	Low
x_2	B	-	High
x_3	-	7	High

Table 8. Some of reduced hydraulic database for $D = \{SF\}$.

No.	H_s	T_s	ρ_s	$\text{Cot}(\alpha)$	S	d_{\min}	Φ_{BRE}	Φ_b	ρ_{sb}	SF
1	5	8	-	3	-	5	34	23	18	1.285
2	5	8	-	3	-	5	34	23	20	1.347
3	5	8	-	3	-	5	34	26	18	1.404
4	5	8	-	3	-	5	34	26	20	1.455
5	5	8	-	3	-	5	40	23	18	1.396
6	5	8	-	3	-	5	40	23	20	1.461
7	5	8	-	3	-	5	40	26	18	1.521
8	5	8	-	3	-	5	40	26	20	1.595
9	5	8	-	3	5	5	34	23	18	1.464
10	5	8	-	3	5	5	34	23	20	1.525
11	5	8	-	3	5	5	34	26	18	1.591
12	5	8	-	3	5	5	34	26	20	1.653
13	5	8	-	3	5	5	40	23	18	1.595
14	5	8	-	3	5	5	40	23	20	1.667
15	5	8	-	3	5	5	40	26	18	1.736

Table 9. Some of reduced hydraulic database for $D = \{D_{n50}\}$.

No.	H_s	T_s	ρ_s	$\text{Cot}(\alpha)$	S	ρ_{sb}	D_{n50}
1	7	11	-	2	2	18	3.0425
2	7	11	-	2	2	20	3.0425
3	7	11	-	2	2	18	3.0425
4	7	11	-	2	2	20	3.0425
5	7	11	-	2	2	18	3.0425
6	7	11	-	2	2	20	3.0425
7	7	11	-	2	2	18	3.0425
8	7	11	-	2	2	20	3.0425
9	7	11	-	2	5	18	2.5331
10	7	11	-	2	5	20	2.5331
11	7	11	-	2	5	18	2.5331
12	7	11	-	2	5	20	2.5331
13	7	11	-	2	5	18	2.5331
14	7	11	-	2	5	20	2.5331
15	7	11	-	2	5	18	2.5331

Following the steps described above, two tables are resulted with totally 15 condition attributes and 3654 values for these attributes; (see Tables 8 and 9). While the original information table, Table 2, has totally 26 condition attributes for two decision ones which these condition attributes have 38880 different values. Comparing the optimized information system with the original one, it is obvious that the information system has been condensed at a very high rate, which will diminish the complexities in system modeling and function approximation tasks.

6. APPROXIMATION THE FUNCTIONALITY PRESENTED BY INFORMATION SYSTEM

After applying the rough set based algorithm as a preprocessing procedure on the information system, we will have a set of parameters free from any vagueness, redundancy and also any internal relation. Using the new set of input parameters we follow the procedure described in previous sections, section 4, designing a neural network in order to model our system and approximate its functionality. Architectures and train parameters for the examined neural networks are presented in Table 4. Also results on these investigations can be found in this table.

It is obvious, from Table 4, that using the new set of attributes, at the first try, will lead us to successfully trained feed forward back-propagation neural network which could approximate decision values from the condition ones. To make a balance on network accuracy and architecture simplicity, we continued examining different configurations of

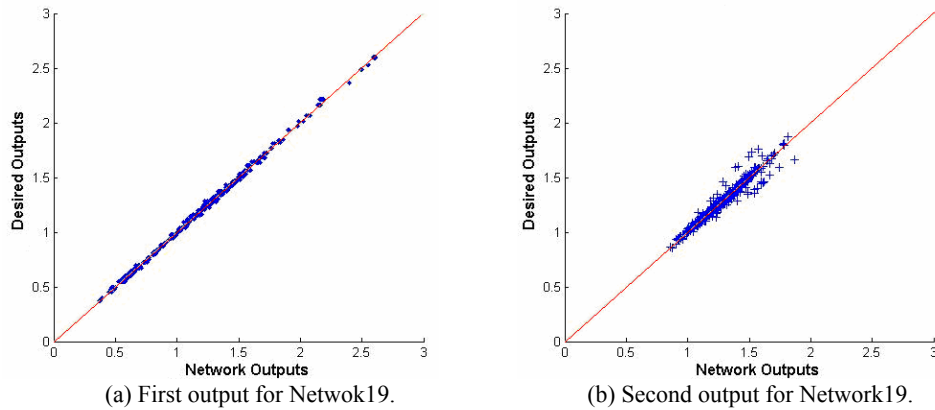


Fig. 4. Network results' using reduced set of condition attributes.

nodes and layers. As we know, the architecture simplicity is as important as the response accuracy, because it will help to improve the network and also will simplify the implementation.

According to Table 4, we can easily get that our system should be a second order function. Because we have reached to very small errors, presented by Linear Correlation Coefficients (LCC) of outputs, using a network architecture with a couple of hidden layers, Network19. Fig. 4 shows the network outputs in comparison with desired ones for Network19. Consider that fitting all outputs on the line $y = x$, in Fig. 4, is the ideal situation.

7. CONCLUSION

System modeling and function approximation are from vital problems in engineering. Solving these problems needs a real good system identification, which is not possible in most cases. By the way, system modeling and function approximation are inevitable tasks when we deal with engineering problems. Modern intelligent approaches, such as neural networks, provide some tools conquering such these problems. In some cases these intelligent methods could not give the proper solution to our problems, of course when we use them as solitude approaches.

This paper has presented an approach to deal with such systems. We showed that due to unknown relations in the condition attributes of some information systems, employing a neural network could not be helpful for approximating functionalities presented by such information systems. In some cases it's easy to extract features and relations hided on data in an information system, but in many cases it could be impossible when we have no idea about what we are looking for. As a solution to such cases an algorithm based on the notions of rough sets theory is introduced and employed to discover hidden relations in information tables. Exploiting this algorithm the size of database have been reduced, which saves lots of computational costs for engineers and computers.

REFERENCES

1. S. Mitra, S. K. Pal, and P. Mitra, "Data mining in soft computing framework: a survey," *IEEE Transactions on Neural Networks*, Vol. 13, 2002, pp. 3-14.
2. M. Kantardzic, *Data Mining: Concepts, Models, Methods and Algorithms*, John Wiley & Sons, Inc., New York, USA, 2002.
3. P. Baldi and S. Brunak, *Bioinformatics: the Machine Learning Approach*, MIT Press Cambridge, MA, USA, 2001.
4. S. Mitra and T. Acharya, *Data Mining: Multimedia, Soft Computing, and Bioinformatics*, John Wiley & Sons, Inc., NY, USA, 2003.
5. U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "The KDD process for extracting useful knowledge from volumes of data," *Communications of the ACM*, Vol. 39, 1996, pp. 27-34.
6. Y. Y. Liu and X. Q. Wu, "Evaluation for data fusion system based on uncertainty," *Journal of Data Acquisition & Processing*, Vol. 20, 2005, pp. 150155.
7. A. Ataei, "Design of rubble mound breakwaters using artificial neural networks," Vol. M. Sc. Tarbiat Modares University, Tehran, Iran, 2002.
8. P. Yang, "Data mining diagnosis system based on rough set theory for boilers in thermal power plants," *Frontiers of Mechanical Engineering in China*, Vol. 1, 2006, pp. 162-167.
9. F. Pentaris and Y. Ioannidis, "Query optimization in distributed networks of autonomous database systems," *ACM Transactions on Database Systems*, Vol. 31, 2006, pp. 537-583.
10. A. Lumini and L. Nanni, "A clustering method for automatic biometric template selection," *Pattern Recognition*, Vol. 39, 2006, pp. 495-497.
11. M. M. Richter and A. Aamodt, "Case-based reasoning foundations," *The Knowledge Engineering Review*, Vol. 20, 2006, pp. 203-207.
12. C. Scott and R. D. Nowak, "Minimax-optimal classification with dyadic decision trees," *IEEE Transactions on Information Theory*, Vol. 52, 2006, pp. 1335-1353.
13. U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, *Advances in Knowledge Discovery and Data Mining*, American Association for Artificial Intelligence, CA, USA, 1996.
14. H. Mannila, "Data mining: machine learning, statistics, and databases," in *Proceedings of the 8th International Conference on Scientific and Statistical Database Management*, 1996, pp. 2-9.
15. A. J. Knobbe, "Multi-relational data mining," The Dutch Graduate School for Information and Knowledge Systems, Utrecht University, Nederland, 2004.
16. A. Buja and Y. S. Lee, "Data mining criteria for tree-based regression and classification," in *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2001, pp. 27-36.
17. R. A. Berk, "Data mining within a regression framework," Thesis, Department of Statistics, University of California, Los Angeles, 2005.
18. Y. Jiang, Z. Y. Zhang, P. L. Qiu, and D. F. Zhou, "Clustering algorithms used in data mining," *Electronic Information Technology*, Vol. 27, 2005, pp. 655-662.
19. E. Kretschmann, W. Fleischmann, and R. Apweiler, "Automatic rule generation for protein annotation with the C4. 5 data mining algorithm applied on SWISS-PROT,"

- Bioinformatics*, Vol. 17, 2001, pp. 920-926.
20. X. D. He and W. G. Liu, "Comparison of association rules mining methods in data mining," *Computer Engineering and Design*, Vol. 26, 2005, pp. 1265-1268.
 21. T. Mielikainen, "Summarization techniques for pattern collections in data mining," Ph.D. Thesis, Department of Computer Science, University of Helsinki, Finland, 2005.
 22. S. Kaski, J. Nikkila, J. Sinkkonen, L. Lahti, J. E. A. Knuuttila, and C. Roos, "Associative clustering for exploring dependencies between functional genomics data sets," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 2, 2005, pp. 203-216.
 23. T. Niu and Z. Hu, "Dynamic visual data mining: biological sequence analysis and annotation using SeqVISTA," *International Journal of Bioinformatics Research and Applications*, Vol. 1, 2005, pp. 18-30.
 24. Z. Pawlak, "Rough sets," *International Journal of Computer and Information Science*, Vol. 11, 1982, pp. 341-356.
 25. Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Norwell, MA, USA, 1992.
 26. Gh. A. Montazer and M. R. Tayefeh, "A new approach for statistical data analysis using rough sets techniques," in *Proceedings of the 6th International Statistics Conference*, 2002, pp. 146-150.
 27. Gh. A. Montazer, R. Sabzevari, H. G. P. Khatir, and F. N. Saleh, "Expert clustering of hydraulic-geotechnical control parameters of rubble mound breakwaters using rough set theory," in *Proceedings of the 5th WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics*, 2006, pp. 133-138.
 28. Gh. A. Montazer and R. Sabzevari, "Intelligent parameter reduction using rough set theory and sensitivity analysis," *WSEAS Transactions on Systems*, Vol. 6, 2007, pp. 623-630.
 29. B. Krose and P. van der Smagt, "An introduction to neural networks," Department of Mathematics and Computer Science, University of Amsterdam, 1993.
 30. A. Horzyk, "Unsupervised clustering using self-optimizing neural networks," in *Proceedings of the 5th International Conference on Intelligent Systems Design and Applications*, 2005, pp. 118-123.
 31. M. Weygandt, R. Stark, C. Blecker, B. Walter, and D. Vaitl, "Realtime fMRI pattern classification using artificial neural networks," *Clinical Neurophysiology*, Vol. 118, Issue 4, 2007, pp. 114.
 32. E. Sanchez, "Fuzzy logic and neural networks in artificial intelligence and pattern recognition," in *Proceedings of SPIE Conference on Stochastic and Neural Methods in Signal Processing, Image Processing, and Computer Vision*, 1991, pp. 474-483.
 33. M. M. Fischer, "Neural networks: a general framework for non-linear function approximation," *Transactions in GIS*, Vol. 10, 2006, pp. 521-533.
 34. S. J. Hsieh and K. Sharma, "Thermography and neural networks for SRAM voltage stress prediction," in *Proceedings of International Conference on SPIE*, Vol. 6205, 2006, pp. 62050Q.
 35. H. Leitgeb, "Interpreted dynamical systems and qualitative laws: from neural networks to evolutionary systems," *Synthesis*, Vol. 146, 2005, pp. 189-202.

36. J. A. Freeman and D. M. Skapura, *Neural Networks: Algorithms, Applications, and Programming Techniques*, Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1991.
37. T. Y. Lin and N. Cercone, *Rough Sets and Data Mining: Analysis of Imprecise Data*, Kluwer Academic Publishers, USA, 1996.
38. O. Kaynak, X. Yu, and M. Onder Efe, "A general backpropagation algorithm for feedforward neural networks learning," *IEEE Transactions on Neural Networks*, Vol. 13, 2002, pp. 251-254.
39. N. B. Karayiannis, A. Mukherjee, J. R. Glover, J. D. Frost, J. R. A. Hrachovy, and E. M. Mizrahi, "An evaluation of quantum neural networks in the detection of epileptic seizures in the neonatal electroencephalogram," *Soft Computing – A Fusion of Foundations, Methodologies and Applications*, Vol. 10, 2006, pp. 382-396.
40. I. A. Basheer and M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application," *Microbiological Methods*, Vol. 43, 2000, pp. 3-31.
41. A. Ismail and A. P. Engelbrecht, "Global optimization algorithms for training product unit neural networks," in *Proceedings of IEEE-INNS-ENNS International Joint Conference on Neural Networks*, Vol. 1, 2000, pp. 132-137.
42. M. P. Perrone and L. N. Cooper, *When Networks Disagree: Ensemble Methods for Hybrid Neural Networks*, Chapman and Hall, London, 1993.
43. X. Yin, Z. Zhou, N. Li, and S. Chen, "An approach for data filtering based on rough set theory," in *Proceedings of the 2nd International Conference on Advances in Web-Age Information Management*, LNCS 2118, 2001, pp. 367-374.



Reza Sabzevari received his B.Sc. degree in Computer-Hardware Engineering from Islamic Azad University of Qazvin, Iran, in 2003 and his M.S. degree in Mechatronics Engineering from the same university in 2007. His research interests center around machine vision, machine learning, artificial intelligence, information engineering and data mining. He is a member of Young Researchers' Club.



Gholam Ali Montazer received his B.Sc. degree in Electrical Engineering from Kh. N. Toosi University of Technology, Tehran, Iran, in 1991, the M.Sc. degree in Electrical Engineering from Tarbiat Modares University, Tehran, Iran, in 1994, and the Ph.D. degree in Electrical Engineering from the same university, in 1998. He is an Assistant Professor of the Department of Information Engineering in Tarbiat Modares University, Tehran, Iran. His areas of research include information engineering, knowledge discovery, intelligent methods, system modeling, e-learning and image mining.