

---

# Discovering Stock Price Prediction Rules Using Rough Sets

Hameed Al-Qaheri<sup>1</sup>, Aboul Ella Hassanien<sup>1,2</sup> and Ajith Abraham<sup>3</sup>

<sup>1</sup> Department Of Quantitative Methods and Information Systems,  
College of Business Administration, Kuwait University, Kuwait

<sup>2</sup> Information Technology Department, FCI, Cairo University  
5 Ahamed Zewal Street, Orman, Giza, Egypt  
Email:alqaheri@cba.edu.kw, a.hassanien@fci-cu.edu.eg, abo@cba.edu.kw

<sup>3</sup> Norwegian University of Science and Technology  
O.S. Bragstads plass 2E, NO-7491 Trondheim, Norway  
Email:ajith.abraham@ieee.org, abraham.ajith@acm.org

**Abstract:** The use of computational intelligence systems such as neural networks, fuzzy set, genetic algorithms, etc. for stock market predictions has been widely established. This paper presents a generic stock pricing prediction model based on rough set approach. To increase the efficiency of the prediction process, rough sets with Boolean reasoning discretization algorithm is used to discretize the data. Rough set reduction technique is applied to find all the reducts of the data, which contains the minimal subset of attributes that are associated with a class label for prediction. Finally, rough sets dependency rules are generated directly from all generated reducts. Rough confusion matrix is used to evaluate the performance of the predicted reducts and classes. Using a data set consisting of daily movements of a stock traded in Kuwait Stock Exchange, a preliminary assessment indicates that rough sets is shown to be applicable and is an effective tool to achieve this goal. For comparison, the results obtained using rough set approach were compared to that of neural networks algorithm and it was shown that Rough set approach have a higher overall accuracy rate and generates more compact and fewer rules than neural networks.

## 1 Introduction

The complexity and difficulty of predicting stock prices, with a reasonable level of precision, on one hand, and the emergence of data mining techniques [12] and computational intelligence techniques such as neural networks, fuzzy set, evolutionary algorithms, rough set theory, etc., as alternative techniques to the conventional statistical regression and Bayesian models with better

performance, on the other hand, have paved the road for the increased usage of these techniques in various areas of economics and finance [10, 9, 4]. Including in these area is the utilization of genetic algorithms and genetic programming [36], for portfolio optimization [1], stocks selection using neural network [2] and predicting the S&P 100 index using rough sets [3] and stocks and futures traders have come to rely upon various types of intelligent systems to make trading decisions [23, 25, 26, 27, 28, 29, 30]. Several intelligent systems have in recent years been developed for modeling expertise, decision support and complicated automation tasks [32, 33, 34, 35].

In recent years and since its inception, rough set theory has gained momentum and has been widely used as a viable intelligent data mining and knowledge discovery techniques in many applications including financial and investment areas. For example, building trading systems using the rough sets model was studied by several researchers. Ziarko et al. [5], Golan and Edwards [6, 7] applied the rough sets model to discover strong trading rules, which reflect highly repetitive patterns in data, from historical database of the Toronto stock exchange. A detailed review of applications of rough sets in financial domain can be found in [8].

The focus of this paper is to present a generic stock price prediction model using rough set theory. The model is expected to extract knowledge in the form of rules from daily stock movements that would guide investors whether to buy, sell or hold a stock. To increase the efficiency of the prediction process, rough sets with Boolean reasoning discretization algorithm is used to discretize the data. Rough set reduction technique is, then, applied to find all reducts of the data which contains the minimal subset of attributes that are associated with a class used label for prediction. Finally, rough sets dependency rules are generated directly from all generated reducts. Rough confusion matrix is used to evaluate the performance of the predicted reducts and classes. Using a data set consisting of daily movements of a stock traded in Kuwait Stock Exchange, a preliminary assessment is reported. And for comparison purposes, the results of Rough Sets are compared with that of the neural networks.

This paper is organized as follows. Section 2 gives a brief introduction to rough sets. Section 3 discusses the proposed rough set prediction model in detail. Experimentation is covered in Section 4 including data preparation and its characteristic, analysis, results and discussion of the results and finally, conclusions are provided in Section 5.

## 2 Rough sets: Foundations

Rough set theory, a new intelligent mathematical tool proposed by Pawlak [17, 18, 19], is based on the concept of approximation spaces and models of sets and concepts. The data in rough sets theory is collected in a table called a decision table. Rows of the decision table correspond to objects, and columns correspond to features. In the data set, we also assume that a set

of examples with a class label to indicate the class to which each example belongs are given. We call the class label a decision feature, the rest of the features are conditional. Let  $\mathcal{O}, \mathcal{F}$  denote a set of sample objects and a set of functions representing object features, respectively. Assume that  $B \subseteq \mathcal{F}, x \in \mathcal{O}$ . Further, let  $[x]_B$  denote:

$$[x]_B = \{y : x \sim_B y\}.$$

Rough set theory defines three regions based on the equivalent classes induced by the feature values: lower approximation  $\underline{B}X$ , upper approximation  $\overline{B}X$  and boundary  $BND_B(X)$ . A lower approximation of a set  $X$  contains all equivalence classes  $[x]_B$  that are subsets of  $X$ , and upper approximation  $\overline{B}X$  contains all equivalence classes  $[x]_B$  that have objects in common with  $X$ , while the boundary  $BND_B(X)$  is the set  $\overline{B}X \setminus \underline{B}X$ , i.e., the set of all objects in  $\overline{B}X$  that are not contained in  $\underline{B}X$ . So, we can define a rough set as any set with a non-empty boundary.

The indiscernibility relation  $\sim_B$  (or by  $Ind_B$ ) is a fundamental principle of rough set theory. Informally,  $\sim_B$  is a set of all objects that have matching descriptions. Based on the selection of  $B$ ,  $\sim_B$  is an equivalence relation partitions a set of objects  $\mathcal{O}$  into equivalence classes. The set of all classes in a partition is denoted by  $\mathcal{O} / \sim_B$  (also by  $\mathcal{O} / Ind_B$ ). The set  $\mathcal{O} / Ind_B$  is called the quotient set. Affinities between objects of interest in the set  $X \subseteq \mathcal{O}$  and classes in a partition can be discovered by identifying those classes that have objects in common with  $X$ . Approximation of the set  $X$  begins by determining which elementary sets  $[x]_B \in \mathcal{O} / \sim_B$  are subsets of  $X$ . For a detailed review of the basic material, reader may consult sources such as [17, 18, 19].

### 3 Rough Sets Prediction Model (RSPM)

Figure (1) illustrates the overall steps in the proposed Rough Set Prediction Model(RSPM) using a UML Activity Diagram where a square or rectangular represents a data object, a rounded rectangular represents an activity, solid and dashed directed lines indicate control flow and data object flow respectively. Functionally, RSPM can be partitioned into three distinct phases:

- *Pre-processing phase(Activities in Dark Gray)*. This phase includes tasks such as extra variables addition and computation, decision classes assignments, data cleansing, completeness, correctness, attribute creation, attribute selection and discretization.
- *Analysis and Rule Generating Phase(Activities in Light Gray)*. This phase includes the generation of preliminary knowledge, such as computation of object reducts from data, derivation of rules from reducts, rule evaluation and prediction processes.

- *Classification and Prediction phase(Activities in Lighter Gray)*. This phase utilize the rules generated from the previous phase to predict the stock price movement

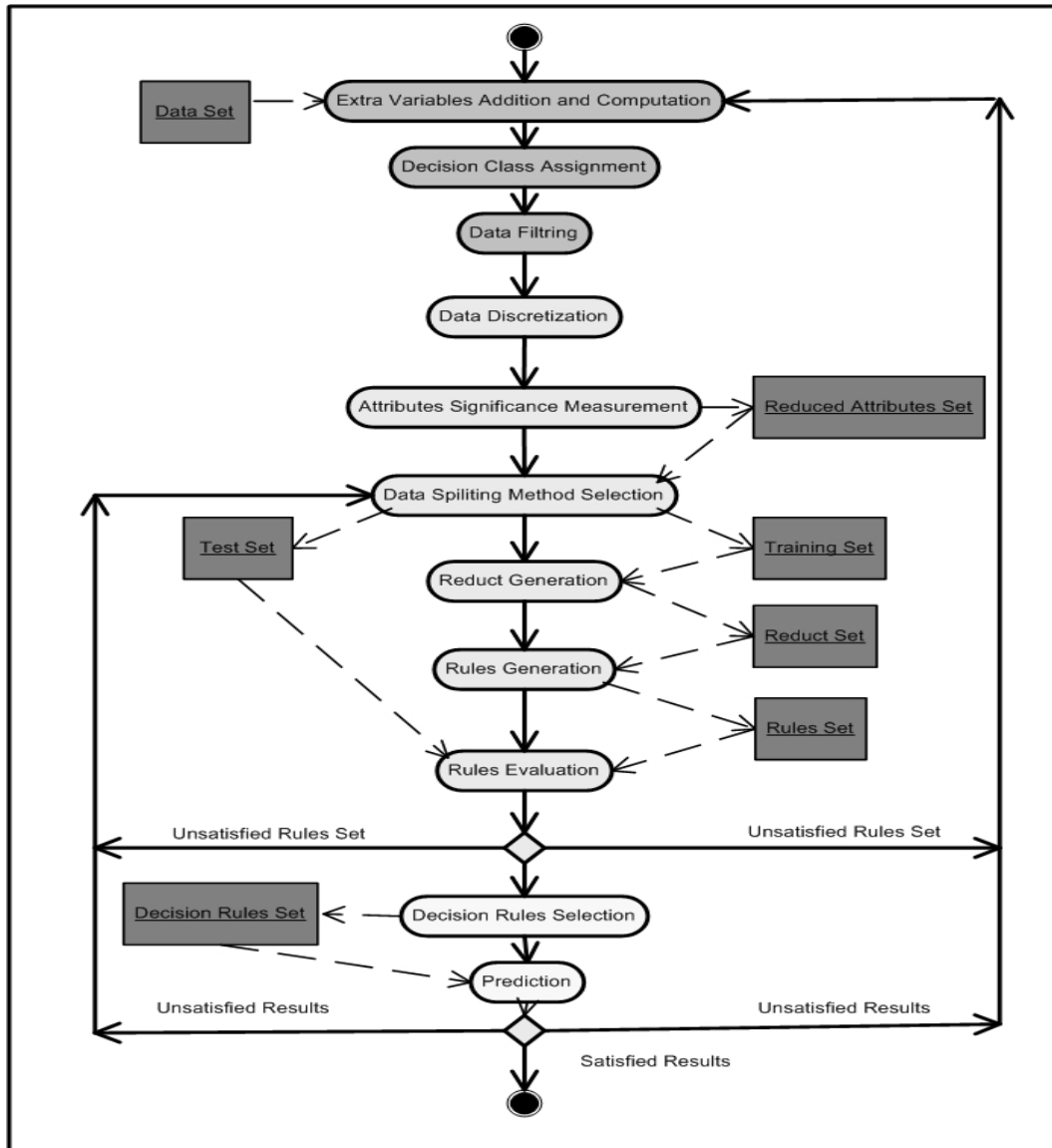


Fig. 1 Overall steps in the Rough Set Prediction Model(RSPM)

### 3.1 Pre-processing phase

In this phase, the decision table required for rough set analysis is created. In doing so, a number of data preparation tasks such as data conversion, data cleansing, data completion checks, conditional attribute creation, decision attribute generation, discretization of attributes are performed. Data splitting is also performed which created two randomly generated subsets, one subset for analysis containing %75 of the objects in the data set and one validation containing the remainder %25 of the objects. It must be emphasized that data conversion performed on the initial data must generate a form in which specific rough set tools can be applied.

#### Data completion and discretization of continuous-valued attributes

Often, real world data contain missing values. Since rough set classification involves mining for rules from the data, objects with missing values in the data set may have undesirable effects on the rules that are constructed. The aim of the data completion procedure is to remove all objects that have one or more missing values. Incomplete data or information systems exist broadly in practical data analysis, and approaches to complete the incomplete information system through various completion methods in the preprocessing stage are normal in data mining and knowledge discovery. However, these methods may result in distorting the original data and knowledge, and can even render the original data to be un-minable. To overcome these shortcomings inherent in the traditional methods, we used the decomposition approach for incomplete information system ( i.e. decision table )proposed in [15].

Attributes in concept classification and prediction, may have varying importance in the problem domain being considered. Their importance can be pre-assumed using auxiliary knowledge about the problem and expressed by properly chosen weights. However, when using rough set approach for concept classification, rough set avoids any additional information aside from what is included in the information table itself. Basically, rough set approach tries to determine from the available data in the decision table whether all the attributes are of the same strength and, if not, how they differ in respect of the classifier power. Therefor, some strategies for discretization of real value attributes have to be used when we need to apply learning strategies for data classification with real value attributes (e.g. equal width and equal frequency intervals). It has been shown that the quality of learning algorithm depends on this strategy [13]. Discretization uses data transformation procedure which involves finding cuts in the data sets that divide the data into intervals. Values lying within an interval are then mapped to the same value. Performing this process leads to reducing the size of the attributes value set and ensures that the rules that are mined are not too specific. For the discretization of continuous-valued attributes, we adopt, in this paper, rough sets with

boolean reasoning (RSBR) algorithm proposed by Zhong et al. [15]. The main advantage of RSBR is that it combines discretization of real valued attributes and classification. The main steps of the RSBR discretization algorithm are outlined below ( refer to Algorithm-1 ).

---

**Algorithm 1** RSBR Discretization Algorithm

---

Input: Information system table ( $S$ ) with real valued attributes  $A_{ij}$  and  $n$  is the number of intervals for each attribute.

Output: Information table ( $ST$ ) with discretized real valued attribute

- 1: **for**  $A_{ij} \in S$  **do**
- 2: Define a set of boolean variables as follows:

$$B = \left\{ \sum_{i=1}^n C_{ai}, \sum_{i=1}^n C_{bi}, \sum_{i=1}^n C_{ci}, \dots, \sum_{i=1}^n C_{ni} \right\} \quad (1)$$

- 3: **end for**  
Where  $\sum_{i=1}^n C_{ai}$  correspond to a set of intervals defined on the variables of attributes  $a$
- 4: Create a new information table  $S_{new}$  by using the set of intervals  $C_{ai}$
- 5: Find the minimal subset of  $C_{ai}$  that discerns all the objects in the decision class  $D$  using the following formula:

$$\mathcal{R}^u = \wedge \{ \Phi(i, j) : d(x_i) \neq d(x_j) \} \quad (2)$$

Where  $\Phi(i, j)$  is the number of minimal cuts that must be used to discern two different instances  $x_i$  and  $x_j$  in the information table.

---

### 3.2 Analysis and Rule Generating Phase

As we mentioned before, Analysis and Rule Generating Phase includes generating preliminary knowledge, such as computation of object reducts from data, derivation of rules from reducts, and prediction processes. These stages lead towards the final goal of generating rules from information system or decision table.

#### Relevant attribute extraction and reduction

One of the important aspects in the analysis of decision tables is the extraction and elimination of redundant attributes and also the identification of the most important attributes from the data set. Redundant attributes are attributes that could be eliminated without affecting the degree of dependency between the remaining attributes and the decision. The degree of dependency is a measure used to convey the ability to discern objects from each other.

The minimum subset of attributes preserving the dependency degree is called reduct. The computation of the core and reducts from a decision table is, in a way, selecting relevant attributes [11, 21].

In decision tables, there often exist conditional attributes that do not provide (almost) any additional information about the objects. These attributes need to be removed in order to reduce the complexity and cost of decision process [11, 14, 20, 21]. A decision table may have more than one reduct. And any of these reducts could be used to replace the original table. However, finding all the reducts from a decision table is NP-complete but fortunately, in applications, it is usually not necessary to find all of them – one or a few of them are sufficient. Selecting the best reduct is important. The selection depends on the optimality criterion associated with the attributes. If a cost function could be assigned to attributes, then the selection can be based on the combined minimum cost criteria. But in the absence of such cost function, the only source of information to select the reduct from is the contents of the table. In this paper, we adopt the criteria that the best reducts are the those with minimal number of attributes and – if there are more such reducts – with the least number of combinations of values of its attributes cf. [11, 16].

In general, rough set theory provides useful techniques to reduce irrelevant and redundant attributes from a large database with a lot of attributes. The dependency degree (or approximation quality, classification quality) and the information entropy are two most common attribute reduction measures in rough set theory. In this paper, we use the dependency degree measure to compute the significant features and measuring the effect of removing a feature from the feature sets. [24].

### Computation of the reducts

A reduced table can be seen as a rule set where each rule corresponds to one object of the table. The rule set can be generalized further by applying rough set value reduction method. The main idea behind this method is to drop those redundant condition values of rules and to unite those rules in the same class. Unlike most value reduction methods, which neglect the difference among the classification capabilities of condition attributes, we first remove values of those attributes that have less discrimination factors. Thus more redundant values can be reduced from decision table and more concise rules can be generated. The main steps of the Rule Generation and classification algorithm are outlined in Algorithm-2:

---

**Algorithm 2** Reduct Generation algorithm

---

Input: information table ( $ST$ ) with discretized real valued attribute.Output: reduct sets  $R_{final} = \{r_1 \cup r_2 \cup \dots \cup r_n\}$ 

```

1: for each condition attributes  $c \in C$  do
2:   Compute the correlation factor between  $c$  and the decisions attributes  $D$ 
3:   if the correlation factor  $> 0$  then
4:     Set  $c$  as relevant attributes.
5:   end if
6: end for
7: Divide the set of relevant attribute into a different variable sets.
8: for each variable sets do
9:   Compute the dependency degree and compute the classification quality
10:  Let the set with high classification accuracy and high dependency as an initial
    reduct set.
11: end for
12: for each attribute in the reduct set do
13:   Calculate the degree of dependencies between the decisions attribute and that
    attribute.
14:   Merge the attributes produced in previous step with the rest of conditional
    attributes
15:   Calculate the discrimination factors for each combination to find the highest
    discrimination factors
16:   Add the highest discrimination factors combination to the final reduct set.
17: end for
18: repeat
    Statements 12
19: until all attributes in initial reduct set is processed

```

---

**Rule generation from a reduced table**

The generated reducts are used to generate decision rules. The decision rule, at its left side, is a combination of values of attributes such that the set of (almost) all objects matching this combination have the decision value given at the rule's right side. The rule derived from reducts can be used to classify the data. The set of rules is referred to as a classifier and can be used to classify new and unseen data. The main steps of the Rule Generation and classification algorithm are outlined as Algorithm-3:

The quality of rules is related to the corresponding reduct(s). We are especially interested in generating rules which cover largest parts of the universe  $U$ . Covering  $U$  with more general rules implies smaller size rule set.

**3.3 Classification and Prediction Phase**

Classification and prediction is the last phase of our proposed approach. We present a classification and prediction scheme based on the methods and tech-



**Algorithm 3** Rule GenerationInput: reduct sets  $R_{final} = \{r_1 \cup r_2 \cup \dots \cup r_n\}$ 

Output: Set of rules

---

```

1: for each reduct  $r$  do
2:   for each correspondence object  $x$  do
3:     Contract the decision rule  $(c_1 = v_1 \wedge c_2 = v_2 \wedge \dots \wedge c_n = v_n) \longrightarrow d = u$ 
4:     Scan the reduct  $r$  over an object  $x$ 
5:     Construct  $(c_i, 1 \leq i \leq n)$ 
6:     for every  $c \in C$  do
7:       Assign the value  $v$  to the correspondence attribute  $a$ 
8:     end for
9:     Construct a decision attribute  $d$ 
10:    Assign the value  $u$  to the correspondence decision attribute  $d$ 
11:  end for
12: end for

```

---

niques described in the previous sections. Figure (2) illustrates the classification scheme for a construction of particular classification and prediction algorithm. To transform a reduct into a rule, one only has to bind the condition feature values of the object class from which the reduct originated to the corresponding features of the reduct. Then, to complete the rule, a decision part comprising the resulting part of the rule is added. This is done in the same way as for the condition features. To classify objects, which has never been seen before, rules generated from a training set will be used. These rules represent the actual classifier. This classifier is used to predict to which classes new objects are attached. The nearest matching rule is determined as the one whose condition part differs from the feature vector of re-object by the minimum number of features. When there is more than one matching rule, we use a voting mechanism to choose the decision value. Every matched rule contributes votes to its decision value, which are equal to the  $t$  times number of objects matched by the rule. The votes are added and the decision with the largest number of votes is chosen as the correct class. Quality measures associated with decision rules can be used to eliminate some of the decision rules.

The global strength defined in [11] for rule negotiation is a rational number in  $[0, 1]$  representing the importance of the sets of decision rules relative to the considered tested object. Let us assume that  $T = (U, A \cup \{d\})$  is a given decision table,  $u_t$  is a test object,  $Rul(X_j)$  is the set of all calculated basic decision rules for  $T$ , classifying objects to the decision class  $X_j(v_d^j = v_d)$ ,  $MRul(X_j, u_t) \subseteq Rul(X_j)$  is the set of all decision rules from  $Rul(X_j)$  matching tested object  $u_t$ . The global strength of decision rule set  $MRul(X_j, u_t)$  is defined by the following form [11]:

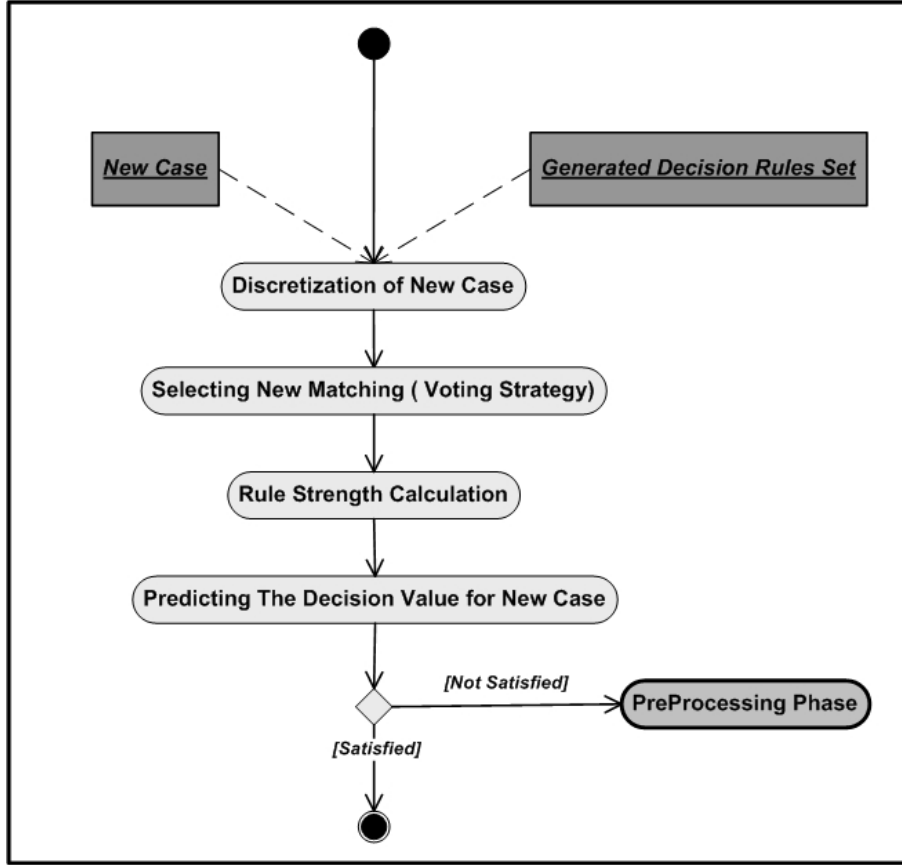


Fig. 2: Rough Set Classification and Prediction Scheme

$$MRul(X_j, u_t) = \frac{|\bigcup_{r \in MRul(X_j, u_t)} |Pred(r)|_A \cap |d = v_d^j|_A|}{||d = v_d^j|_A|}.$$

Measure of strengths of rules defined above is applied in constructing classification algorithm. To classify a new case, rules are first selected matching the new object. The strength of the selected rule sets is calculated for any decision class, and then the decision class with maximal strength is selected, with the new object being classified to this class.

## 4 Experimentation

### 4.1 Data Set and its Characteristics

To test and verify the prediction capability of the proposed RSPM, the daily stock movement of a banking stock traded in Kuwait Stock Exchange and spanning over a period of 7 years ( 2000-2006), were captured. Figure (3) depicts a sample of the stock's daily movements.

Sector	stk	Ticker	Date	Last	High	Low	Vol	Trade	Value
Banking	102	GBK	01/02/2000	410	410	400	150000	9	60800
Banking	102	GBK	01/03/2000	405	405	405	140000	4	56700
Banking	102	GBK	01/04/2000	405	405	405	1010000	31	409050
Banking	102	GBK	01/05/2000	405	405	405	370000	7	149850
Banking	102	GBK	01/11/2000	400	400	400	130000	5	52000
Banking	102	GBK	01/12/2000	400	400	400	10410000	14	4164000
Banking	102	GBK	15/01/2000	400	400	400	990000	16	396000
Banking	102	GBK	16/01/2000	400	400	400	1450000	27	580000
Banking	102	GBK	17/01/2000	400	400	400	1740000	30	696000
Banking	102	GBK	18/01/2000	400	400	400	1550000	22	620000

Fig. 3: A Sample of the Stock Daily Movement

Table 1 shows the attributes used in the creation of the rough set decision table, where *MA*: Moving average of price, *UP*: Upward price change, *Dw*: Downward price change;  $P_i$ : closing price. The first five attributes in the Table, i.e. Last( or Closing Price), High, Low, Trade, and Value) were extracted from the stock daily movement. The other important attributes in the table were compiled from the literature [22] along with the formula for their computation. The decision attributed, *D*, in the Table 1, which indicates the future direction of the the data set, is constructed using the following formula:

$$Dec_{att} = \frac{\sum_{i=1}^{i=n} ((n+1) - i) \cdot sign[close(i) - close(0)]}{\sum_{i=1}^n i} \quad (3)$$

where close (0) is today's closing price and close (i) is the ith closing price in the future. Equation (1) specifies a range -1 to +1 for  $Dec_{att}$ . A value of +1 indicate that every day up to  $n$  days in the future, the market closed higher than today. Similarly, -1 indicates that every day up to  $n$  days in the future, the market closed lower than today.

**Table 1.** Stock Price Movement Decision Table

Attribute	Attribute description
Last	closing price
High	High price
Low	Low price
Trade	
Value	
$Lag_i, i = 1..6$	An event occurring at time $t + k$ ( $k > 0$ ) is said to lag behind event occurring at time $t$ ,
$Aver_5$	moving average of 5 days for close price
Momentum	$P_i - P_{i-4}$
Disparity in 5 days	$\frac{P_i}{MA_5} * 100$
Price Osculator	$OSCP = 100 - \frac{100}{1 + \frac{\sum_{i=0}^{n-1} UP_{i-1}/n}{\sum_{i=0}^{n-1} DW_{i-1}/n}}$
RSI (relative strength index)	$= 100 - \frac{100}{\sum_{i=0}^{n-1} UP_i/n}$
ROC	rate of change $\frac{P_i - P_{i-n}}{P_i} * 100$
D	Decision attribute

Figure (4) present a snapshot of the 21 index for the period covering from Jan. 1st 2000 to Jan. 31th 2000, and the fluctuation of the  $Dec_{att}$ . And Figure (5) shows part of the calculated daily stock movement time series data set according the attributes described in Table 1.

## 4.2 Analysis, Results and Discussion

For many data mining tasks, it is useful to learn about the general characteristics of the given data set and to identify the outliers - samples that are not consistent with the general behavior of the data model. Outlier detection is important because it may affect the classifier accuracy. As such we performed several descriptive statistical analysis, such as measures of central tendency and data dispersion. And in our statistical analysis we used the mean and the median to detect the outliers in our data set. Table (2) represent the statistical analysis and essential distribution of attributes, respectively.

We reach the minimal number of reducts that contains a combination of attributes which has the same discrimination factor. The final generated reduct sets which are used to generate the list of rules for the classification are:

**{high, low, last, momentum, disparity in 5 days, Roc}**

A natural use of a set of rules is to measure how well the ensemble of rules is able to classify new and unseen objects. To measure the performance of the rules is to assess how well the rules do in classifying new cases. So we apply the rules produced from the training set data to the test set data.

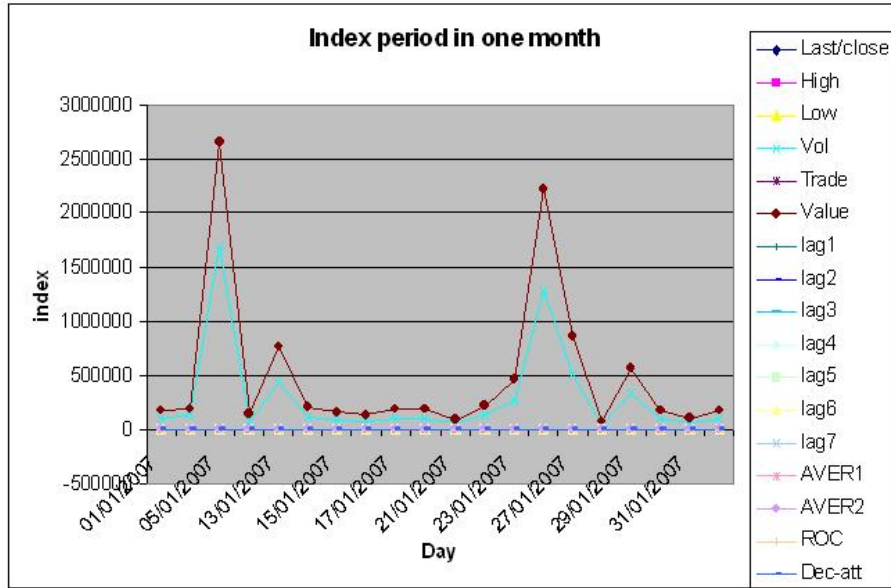


Fig. 4 Snapshot of 21 index for the period covering Jan. 2000

Date	Last/close	High	Low	Vol	Trade	Value	lag1	lag2	lag3	lag4	lag5	lag6	lag7	AVER1	AVER2	ROC	Dec.att
01/01/2007	1720	1720	1700	105000	10	178700	1720	1720	1720	1720	1720	1720	1720	1720	1721.67	0	0
04/01/2007	1500	1500	1480	130000	12	194000	1480	1480	1520	1520	1520	1520	1520	1500	1511.67	1.351351	1
05/01/2007	1580	1620	1560	1685000	76	2657300	1580	1620	1580	1600	1600	1620	1592	1601.67	0	0	0
07/01/2007	1740	1780	1740	32500	8	143950	1780	1740	1740	1740	1700	1740	1700	1748	1720	-2.24719	-1
13/01/2007	1800	1820	1760	435000	24	773000	1800	1840	1860	1840	1780	1720	1828	1770	0	0	0
14/01/2007	1800	1800	1780	117500	9	209200	1800	1800	1840	1860	1840	1780	1720	1820	1776.67	0	0
15/01/2007	1780	1780	1760	95000	25	168050	1800	1800	1800	1840	1860	1840	1780	1804	1781.67	-1.11111	-1
16/01/2007	1780	1800	1760	72500	20	128750	1780	1800	1800	1800	1840	1860	1840	1792	1786.67	0	0
17/01/2007	1760	1760	1740	110000	11	191600	1780	1780	1800	1800	1800	1840	1860	1784	1790	-1.1236	-1
20/01/2007	1760	1760	1740	110000	11	191600	1760	1780	1780	1800	1800	1840	1840	1776	1793.33	0	0
21/01/2007	1760	1760	1700	55000	6	94900	1760	1760	1780	1780	1800	1800	1800	1768	1796.67	0	0
22/01/2007	1740	1740	1700	132500	24	226450	1760	1760	1760	1780	1780	1800	1800	1760	1793.33	-1.13636	-1
23/01/2007	1740	1740	1700	270000	24	464050	1740	1760	1760	1760	1780	1780	1800	1752	1785	0	0
24/01/2007	1760	1780	1720	1280000	91	2216600	1740	1740	1760	1760	1760	1780	1780	1752	1776.67	1.149425	1
27/01/2007	1740	1760	1720	502500	42	869100	1760	1740	1740	1760	1760	1760	1780	1748	1768.33	-1.13636	-1
28/01/2007	1740	1740	1720	40000	3	69200	1740	1760	1740	1740	1760	1760	1760	1744	1763.33	0	0

Fig. 5: Samples of the banking sector data - after post processing

The following present the rules in a more readable format:

**R1: IF** Closing Price(Last) = (403 OR 408) AND  
 High = (403 OR 408) AND  
 Low = (3 OR 8) AND  
 momentum = (403 OR 408) AND  
 disparityin5dayes = (100.48700 OR 100.60700) AND  
 ROC = (-0.50505 OR 0.51021)  
**THEN** Decision Class is 0.0

**Table 2.** Statistical results of the attributes

Attribute	Mean	Std. Dv	Median	Correlation with decision class
Last-Close	497.8	145.17	490.0	0.255
High	498.9	145.6	490	0.2500
Low	493.7	143.5	485.0	0.24
Vol	626189.3	1314775.6	240000	0.097
Trade	13.3	15.12	8.0	0.185
Value	322489.3	674862.3	118900.0	0.1065
Lag1	522.25	94.5	490.0	-0.0422
Lag2	493.8	0.4828	490.0	0.0055
Lag3	496.4	148.5	490.0	0.092
Aver5	501.5	103.6	488.0	0.075
Momentum	2.44	163.1	0.0	0.266
Disparity in 5 days	99.0	25.2	100.3	0.28
Price Osculator	.0002	0.095	0.006	0.156
RSI	49.8	1.4.36	49.8	-0.035
ROC	-4.7	21.5	0.0	-0.365

Table (3) shows a partial set of the generated rules. These obtained rules are used to build the prediction system.

Several runs were conducted using different setting with strength rule threshold. Rule importance and rule strength measures are used to obtain a sense of the quality of the extracted rules. These measures are chosen according to the number of times a rule appears in all reducts, number of generated reducts, and the support the strength of a rule. The rule importance and Rule Strength are given by the following forms:

**Rule Importance** Rule Importance measures ( $Importance_{rule}$ ) is used to assess the quality of the generated rule and it is defined as follows:

$$Importance_{rule} = \frac{\tau_r}{\rho_r}, \quad (4)$$

where  $\tau_r$  is the number of times a rule appears in all reducts and  $\rho_r$  is the number of reduct sets.

**Rule Strength** the strength of a rule,  $Strength_{rule}$ , states how well the rule covers or represent the data set and can be calculated as follows:

$$Strength_{rule} = \frac{Support_{rule}}{|U|}, \quad (5)$$

where  $|U|$  denotes the number of all objects in the training data or objects in the universe in general. The strength of a rule states how well the rule covers or represents the data set.

Table (4) shows the number of generated rules using rough sets and for the sake of comparison we have also generated rules using neural network.

**Table 3.** A Partial Set of the generated rules

Rule number	rule form
R1	Last/close=(403 or 408) AND High=(403 RO 408) AND Low=(403 or 408) AND momentum=(3 OR 8) AND disparityin5dayes=(100.48700 or 100.60700) AND ROC=(-0.50505 or 0.51021) $\implies d = 0$
R2	Last/close=(398 or 403) AND High=(398 or 403) AND Low=(393 or 398) AND momentum=(-2 or 3) AND disparityin5dayes=(125.19600 or 125.43000) AND ROC=(-0.50505 or 0.51021) $\implies d = 0$
R3	Last/close=(403 or 408) AND High( 403 or 408) AND Low=(398 or 403) AND momentum(3 or 8) AND disparityin5dayes=(100.93900 or 101.01500) AND ROC=(0.51021) $\implies d = 1.0$
R4	Last/close=(378 or 385) AND High( 378 or 385 ) AND Low=(378 or 385) AND momentum=(-25 or -17) AND disparityin5dayes=(97.70110) AND ROC=(-0.50505) $\implies d = -1.0$
R5	Last/close=(183 or 370) AND High=(368, 373) AND Low=(183, 368) AND momentum=(-37, -32) AND disparityin5dayes=(113.76700 or 120.81700) AND ROC=(-0.50505) $\implies d = 1.0$
R6	Last/close=(403, 408) AND High=(403 or 408) AND Low=(398 or 403) AND momentum=(-2 or 3) AND disparityin5dayes=(100.24500 or 100.27300) AND ROC=(0.51021) $\implies d = 1.0$

Table (4) indicates that the number of rules generated using neural networks is much larger than the rough sets.

**Table 4.** Number of generated rules

Method	Generated rule numbers
Neural networks	630
Rough sets	371

Measuring the performance of the rules generated from the training data set in terms of their ability to classify new and unseen objects is also impor-

tant. Our measuring criteria were Rule Strength and Rule Importance [31] and to check the performance of our method, we calculated the confusion matrix between the predicted classes and the actual classes as shown in Table (5). The confusion matrix is a table summarizing the number of true positives, true negatives, false positives, and false negatives when using classifiers to classify the different test objects.

**Table 5.** Model Prediction Performance (Confusion Matrix)

Actual	Predict Class1	Predict Class2	Predict Class3	Accuracy
Class1 (-1)	39	1	0	0.975 %
Class2 (0)	0	76	0	1.0 %
Class3 (+1)	0	2	34	0.94%
	1.0	.962	1.0	0.9802 %

## 5 Conclusions and Future Research

The paper presented a generic stock price prediction model using rough set theory. The model was able to extract knowledge in the form of rules from daily stock movements. These rules then could be used to guide investors whether to buy, sell or hold a stock. To increase the efficiency of the prediction process, rough sets with Boolean reasoning discretization algorithm is used to discretize the data. Rough set reduction technique is, then, applied to find all reducts of the data which contains the minimal subset of attributes that are associated with a class used label for prediction. Finally, rough sets dependency rules are generated directly from all generated reducts. Rough confusion matrix is used to evaluate the performance of the predicted reducts and classes.

Using a data set consisting of daily movements of a stock traded in Kuwait Stock Exchange, a preliminary assessment showed that performance of the rough set based stock price prediction model, given the limited scoped of the data set, was highly accurate and as such this investigation could lead to further research using a much larger data set consisting of the entire Kuwait Stock Exchange, which would in turn prove the model's generalizability that the model is accurate and sufficiently robust and reliable as a forecasting and prediction model. For comparison purposes, the results obtained using rough sets were compared to those generated by neural networks algorithms. It was shown, using the same constrained data set, that rough set approach have a higher overall accuracy rates and generate more compact and fewer rules than neural networks. A future research, based on this finding, could be to implement a hybrid approach using Rough Sets as reducts generator



and neural networks for knowledge discovery and rule generator utilizing the Rough set reducts.

## References

1. R.J. Bauer Jr., (1994) Genetic Algorithms and Investment Strategies, Wiley, New York, 1994.
2. G. Mani, K.K. Quah, S. Mahfoud, D. Barr (1995) An analysis of neural-network forecasts from a large-scale, real-world stock selection system. Proceedings of the IEEE/IAFE 1995 Conference on Computational Intelligence for Financial Engineering (CIFER95), IEEE, New Jersey, 1995, pp. 72-78.
3. Skalko C., (1996) Rough sets help time the OEX. *Journal of Computational Intelligence in Finance* 4 (6) (1996) 2027.
4. Andreas Graflund (2002) Bayesian Inference Approach to Testing Mean Reversion in the Swedish Stock Market. Available at SSRN: <http://ssrn.com/abstract=250089>
5. W. Ziarko, R. Golan and D. Edwards (1993) An application of datalogic/R knowledge discovery tool to identify strong predictive rules in stock market data. In Proceedings of AAAI Workshop on Knowledge Discovery in Databases, Washington, DC, pp. 891-901.
6. Golan, R.H. and Ziarko, W. (1995) A methodology for stock market analysis utilizing rough set theory. Proceedings of the IEEE/IAFE Computational Intelligence for Financial Engineering, 9-11 Apr, pp. 32-40.
7. Golan, R., (1995) Stock market analysis utilizing rough set theory. Ph.D. Thesis, Department of Computer Science, University of Regina, Canada.
8. Francis E. H. Tay, and Lixiang Shen (2002) Economic and financial prediction using rough sets model. *European Journal of Operational Research* Volume 141, Issue 3, 16 September 2002, pp. 641-659
9. Jovina Roman and Akhtar Jameel (1996) Backpropagation and Recurrent Neural Networks in Financial Analysis of Multiple Stock Market Returns. Proceedings of the 29th Hawaii International Conference on System Sciences (HICSS) Volume 2: Decision Support and Knowledge-Based Systems, pp. 454-463.
10. Shuxiang Xu and Ming Zhang (2005) Data mining - an adaptive neural network model for financial analysis. *Third International Conference on Information Technology and Applications, ICITA2005*, vol. 1(4-7), pp. 336-340.
11. Bazan, J., Nguyen, H.S., Nguyen, S.H., Synak, P., Wróblewski, J.: Rough Set Algorithms in Classification Problem. In: Polkowski, L., Tsumoto, S., Lin, T.Y. (eds.), *Rough Set Methods and Applications*. Physica Verlag (2000) pp. 49-88.
12. Cios K., and Pedrycz W., and Swiniarski R., (1998) *Data Mining Methods for Knowledge Discovery*. Kluwer Academic Publishers, 1998.
13. Dougherty J., Kohavi R., and Sahami M., (1995) Supervised and Unsupervised Discretization of Continuous Features. Proc. of the XII International conference on Machine Learning, San Francisco, pp. 294-301.
14. Gediga G. Duentsch I. (2000) Statistical Techniques for Rough Set Data Analysis. In: *Rough Sets: New Developments*. Physica Verlag, pp. 545-565. 907-911.
15. Qizhong, Zhang (2007) An Approach to Rough Set Decomposition of Incomplete Information Systems. 2nd IEEE Conference on Industrial Electronics and Applications, ICIEA2007, 23-25 May 2007, pp. 2455-2460.

16. Ning S. Ziarko W. Hamilton J. and Cercone N. (1995) Using Rough Sets as Tools for Knowledge Discovery. In: First International Conference on Knowledge Discovery and Data Mining KDD'95, U.M. Fayyad, R. Uthurusamy (eds.), Montreal, Que., Canada, AAAI, pp.263-268.
17. Pawlak Z., (1982) Rough Sets. Int. J. Computer and Information SCI., Vol. 11 pp. 341-356.
18. Pawlak Z. (1991) Rough Sets- Theoretical aspect of Reasoning about Data. Kluwer Academic Publishers, 1991.
19. Pawlak Z., Grzymala-Busse J., Slowinski R., Ziarko, W., ”(1995) Rough sets. Communications of the ACM, Vol. 38, No. 11, pp. 89-95.
20. Polkowski L. (2003) Rough Sets: Mathematical Foundations. Physica-Verlag.
21. Starzyk J.A. Dale N. and Sturtz K. (2000) A mathematical foundation for improved reduct generation in information systems. In Knowledge and Information Systems Journal, Springer, pp.131-147.
22. Lixiang Shen and Han Tong Loh (2004) Applying rough sets to market timing decisions. Decision support Systems journal, vol. 37, pp. 583-597.
23. A. Abraham, B. Nath and P.K. Mahanti In: V.N. Alexandrov et al., Editors, Hybrid Intelligent Systems for Stock Market Analysis, Computational Science, Springer-Verlag, Germany, USA (2001), pp. 337345
24. A.E. Hassanien, M.E. Abdelhafez, H.S. Own, “Rough Sets Data Analysis in Knowledge Discovery: A Case of Kuwaiti Diabetic Children Patients”, *Advances in Fuzzy System*, 2008, *in press*.
25. A. Abraham, N.S. Philip, B. Nath, P. Saratchandran, Performance analysis of connectionist paradigms for modeling chaotic behavior of stock indices, Second International Workshop on Intelligent Systems Design and Applications, Computational Intelligence and Applications, Dynamic Publishers Inc., USA, 2002, pp. 181186.
26. A. Abraham, N.S. Philip and P. Saratchandran, Modeling chaotic behavior of stock indices using intelligent paradigms, Int. J. Neural Parallel Sci. Comput. 11 (2003) (12), pp. 143160.
27. W.S. Chan and W.N. Liu, Diagnosing shocks in stock markets of southeast Asia, Australia, and New Zealand, Math. Comput. Simulation 59 (2002) (13), pp. 223232.
28. E.S. Francis Tay and L.J. Cao, Modified support vector machines in financial time series forecasting, Neurocomputing 48 (2002) (14), pp. 847861.
29. K.J. Kim and I. Han, Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index, Expert Syst. Appl. 19 (2000) (2), pp. 125132.
30. W. Leigh, N. Modani, S. Purvis and T. Roberts, Stock market trading rule discovery using technical charting heuristics, Expert Syst. Appl. 23 (2002) (2), pp. 155159.
31. Li J. and Cercone N. (2005) A Rough Set Based Model to Rank the Importance of Association Rules. In: RSFDGrC'2005 (2), LNAI, Springer, pp.109-118.
32. W. Leigh, R. Purvis and J.M. Ragusa, Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural network and genetic algorithm: a case study in romantic decision support, Decision Support Syst. 32 (2002) (4), pp. 361377.
33. Maqsood, M.R. Khan and A. Araham, Neural network ensemble method for weather forecasting, Neural Comput. Appl. 13 (2004) (2), pp. 112122

34. R. Tsaih, Y. Hsu and C.C. Lai, Forecasting S&P 500 stock index futures with a hybrid AI system, *Decision Support Syst.* 23 (1998) (2), pp. 161-174.
35. J. Van den Berg, U. Kaymak and W.-M. Van den Bergh, Financial markets analysis by using a probabilistic fuzzy modelling approach, *Int. J. Approximate Reasoning* 35 (2004) (3), pp. 291-305
36. C. Grosan and A. Abraham, Stock Market Modeling Using Genetic Programming Ensembles, *Genetic Systems Programming*, Nadia Nedjah et al. (Eds.), *Studies in Computational Intelligence*, Springer Verlag Germany, ISBN: 3-540-29849-5, pp. 131-146, 2006.