ResearchGate

Developing a rule change trading system for the futures market using rough set analysis

Article *in* Expert Systems with Applications · October 2016

DOI: 10.1016/j.eswa.2016.04.031

2 authors:

Youngmin Kim
Missouri University of Science and Technology

David Enke
Missouri University of Science and Technology

Expert Systems With Applications 59 (2016) 165–173

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Developing a rule change trading system for the futures market using rough set analysis

Youngmin Kim[a], David Enke[b,*]

[a] Department of Engineering Management and Systems Engineering, Missouri University of Science and Technology, 205 Engineering Management, 600 W. 14th Street, Rolla, MO 65409-0370, USA

[b] Department of Engineering Management and Systems Engineering, Missouri University of Science and Technology, 221 Engineering Management, 600 W. 14th Street, Rolla, MO 65409-0370, USA

ARTICLE INFO

Article history:
Received 9 March 2016
Revised 24 April 2016
Accepted 25 April 2016
Available online 26 April 2016

Keywords:
Rough set
Genetic algorithm
Rule change trading system
Futures market

ABSTRACT

Many technical indicators have been selected as input variables in order to develop an automated trading system that determines buying and selling trading decision using optimal trading rules within the futures market. However, optimal technical trading rules alone may not be sufficient for real-world application given the endlessly changing futures market. In this study, a rule change trading system (*RCTS*) that consists of numerous trading rules generated using rough set analysis is developed in order to cover diverse market conditions. To change the trading rules, a rule change mechanism based on previous trading results is proposed. Simultaneously, a genetic algorithm is employed with the objective function of maximizing the payoff ratio to determine the thresholds of market timing for both buying and selling in the futures market. An empirical study of the proposed system was conducted in the Korea Composite Stock Price Index 200 (KOSPI 200) futures market. The proposed trading system yields profitable results as compared to both the buy-and-hold strategy, and a system not utilizing a genetic algorithm for maximizing the payoff ratio.

1. Introduction

It is well-known that the stock market is a dynamic and complex system with noisy, non-stationary data (Kim & Han, 2000; Peters, 1994), making the prediction of future stock prices one of the more challenging problems (Baa & Yang, 2008; Bao, 2008; Yao & Herbert 2009). For more than a century a large variety of technical indicators have been developed and have been broadly used by investors as a decision support tool when analyzing the stock market (Menkhoff, 2010; Murphy, 1999). With the development of various computational intelligence approaches, in particular machine learning and data mining techniques, many researchers have begun to develop automated trading systems to predict future stock price movements (Bagheri, Peyhani, & Akbari, 2014; Booth, Gerding, & McGroarty, 2014; Patel, Shah, Thakkar, & Kotecha, 2015; Rather, Agarwal, & Sastry, 2015), as well as determine the market timing of buying and selling stocks based on technical indicators (Barak, Dahooie, & Tichý, 2015; Bogullu, Enke, & Dagli, 2002; Cervelló-Royo, Guijarro, & Michniuk, 2015; Chavarnakul & Enke, 2008, 2009; Chen & Chen, 2016; Mabu et al., 2013; Thawornwong et al., 2001). Algorithms such as fuzzy logic, neural networks, evolutionary algorithms, support vector machines, and rough set analysis, among others, have been utilized given their ability to handle uncertainty and imprecise stock market data (Vanstone & Tan, 2003, 2005). Among these approaches, rough set theory, which was introduced by Pawlak (1982), has increased in popularly for stock market analysis, market timing, and trading rule generation. Rough set theory provides a powerful mathematical tool that allows one to acquire information from vague and uncertainty data that is not typically possible utilizing traditional set theory (Pawlak, 1982, 1997; Slowinski et al., 1997; Tay & Sehn, 2002).

However, one of the important problems when using rough set theory (or pre-processing in data mining) involves the discretization of continuous data, given that rough set analysis can only handle discrete data. Nguyen and Skowron (1995) proved that the complexity of the discretization problem is an NP-hard problem in data pre-processing. Many discretization methods have been used to help overcome this problem, such as equal width, equal frequency, clustering based, entropy based, and Chi-square based discretization methods. Moreover, attribute reduction is another significant process of rough set analysis, allowing one to discover a reduct (i.e., identify a subset of attributes) with a minimal number of attributes among all attributes. This is also known to be an NP-hard problem (Pawlak et al., 1995). Many reduction methods,

* Corresponding author.
E-mail addresses: kimym@mst.edu (Y. Kim), enke@mst.edu (D. Enke).

http://dx.doi.org/10.1016/j.eswa.2016.04.031
0957-4174/

such as discretization methods, have been developed for rough set analysis. For example, reducts can now be generated using techniques such as manual reducers, genetic algorithms, the Johnson algorithm, and the dynamic reduct method (Øhrn, 1999). Therefore, when researchers or traders develop a trading system based on technical indicators using rough set analysis, two processes of discretization and reducts are required in order to generate trading rules (or decision rules) in the domain of the financial markets.

Numerous researchers have applied rough set analysis to generate trading rules for the stock market. For example, Shen and Loh (2004) developed a trading system for generating market-timing signals using rough sets analysis. They used a modified Chi-squared algorithm for discretization and an entropy measure for finding reducts. Yao and Herbert (2009) demonstrated a rough set analysis for the discovery of decision rules from the New Zealand stock exchanges. They proposed a unique ranking system for decision rules based on the strength of the rule that employed an equal frequency method for discretization, along with a genetic algorithm for searching reducts to generate trading rules. Lee et al. (2010, 2012) developed a framework called the expanded real-time rule-based trading system (*eRRTS*), which expanded the set of templates of a real-time rule-based trading system (*RRTS*) through the use of rough set modeling and a dynamic time warping (DTW) algorithm. They applied an equal frequency method and manual reducers for generating trading rules. However, a stable trading system is not always possible when applying only optimal trading rules or a single trading rule since there is no guarantee that the extracted trading rules are always useful (Pardo, 2008; Pring 1991; Wang, Yu, & Cheung, 2014). In general, these studies are limited in that the generated trading rules are difficult to apply in the future due to endless changes in the market.

To overcome the aforementioned limitations, this study focuses on extracting a number of trading rules in order to cover diverse market situations and then change the trading rules according to trading performance. To do this, rough set analysis is adopted for generating trading rules that consist of a combination of an equal frequency discretization and a manual reduction method. The resulting *Rule Change Trading System* (*RCTS*) is developed for the futures market. *RCTS* extracts and stores numerous trading rules within a trading rule repository, with the subsequent buying and selling signals determined by a unique rule change mechanism. Additionally, the approach adopts a genetic algorithm in order to facilitate the rule change mechanism.

This study is organized as follows. Section 2 briefly reviews the rough set analysis and the genetic algorithm. Section 3 describes the construction procedure for the *RCTS*. Section 4 presents an empirical study that was performed to validate the performance of the *RCTS*. Finally, concluding remarks and areas for future work are presented in Section 5.

2. Research background

2.1. Rough set theory

Since rough set theory was first introduced by Pawlak (1982), it has been popularly applied to generate trading rules in the stock market. Rough set theory provides a mathematical approach to handle vagueness and imprecise information in data (Tay & Sehn, 2002), and includes three important concepts: the indiscernibility relation, set approximation, and reduction of attributes (Cheng et al., 2010).

Let $S = (U, A \cup \{d\})$ be a decision table (S), where U is the universe of discourse (a non-empty finite set of objects), $d (d \notin A)$ is the decision attribute, and the elements of A are called conditional attributes. As an example, a very simple decision table is shown in Table 1. The set of objects U consists of five objects, four conditional attributes, and one decision attribute with two values (i.e., Up or Down).

Table 1
A decision table.

U	Conditional attributes (A)				Decision attribute (D)
	a_1	a_2	a_3	a_4	d
x_1	1	0	2	1	Up
x_2	0	0	1	1	Up
x_3	2	0	2	1	Up
x_4	0	0	2	2	Down
x_5	1	0	2	1	Down

However, as illustrated in Table 1, it is not possible to discern between the x_1 and x_5 objects since the values of conditional attributes are the same, while the decision attribute (value) is different. This relationship is called the *indiscernibility relation* (*IND*) in rough set theory, which, if $B \subseteq A$, is defined as follows:

$$IND(B) = \{(x, x') \in U \times U | \forall a \in B, a(x) = a(x')\}.$$

As given in Table 1, if we take into consideration the set $B = \{a_1, a_2, a_3, a_4\}$, the B-indiscernibility relation defines $IND(B) = \{\{x_1, x_5\}, \{x_2\}, \{x_3\}, \{x_4\}\}$. Likewise, if it takes two attributes $B = \{a_1, a_2\}$, the B-indiscernibility relation defines $IND(B) = \{\{x_1, x_5\}, \{x_2, x_4\}, \{x_3\}\}$.

To approximate a set $X (X \subseteq U)$ on the basis of information in the set of attributes B, the B-lower ($\underline{B}X$) and B-upper ($\overline{B}X$) approximations of $X$are used. The objects in $\underline{B}X$ can definitely be classified as elements of X, while the objects in $\overline{B}X$ can only be classified as possible elements of X. The difference between $\overline{B}X$ and $\underline{B}X$ is called the B-boundary region of $X (BN_B(X))$. The set of X is called the rough set if the set of $BN_B(X)$ is non-empty, otherwise, the set is crisp. Definitions of lower approximations, upper approximations, and boundary regions of X are as follows:

$$\underline{B}X = \{x \mid [x]_B \subseteq X\}$$

$$\overline{B}X = \{x \mid [x]_B \cap X \neq \phi\}$$

$$BN_B(X) = \overline{B}X - \underline{B}X$$

where $[x]_B$ is the equivalence classes of the *B-indiscernibility* relation. Let $X = \{x : D(x) = Up\}$, as given in Table 1. The set X is approximated by the set of conditional attributes $B = \{a_1, a_2, a_3, a_4\}$. Thus, the following approximations are obtained: $\underline{B}X = \{x_2, x_3\}$, $\overline{B}X = \{x_1, x_2, x_3, x_5\}$, and $BN_B(X) = \{x_1, x_5\}$.

The minimal subset of attributes is called a "reduct", which provides the same quality of approximation (discernibility) as the whole set of attributes. In other words, the other attributes, except the elements in a reduct, can be removed without losing any information. The B-reduct of X is denoted by $RED(B)$. The intersection of all reducts is called the B-core of X, or $CORE(B)$, which is a set of the most significant attributes. As given in Table 1, reducts are defined as $RED(B) = \{a_1, a_4\}$ or $RED(B) = \{a_1, a_3\}$, whereas core is $CORE(B) = \{a_1\}$ (i.e., $X = \{x : D(x) = Up\}$).

After performing reducts, the decision rules can be generated by determining the decision attributes value, based on the condition attributes values. The rules are presented in an "*IF condition(s) THEN decision(s)*" format. Each decision rule is assessed based on measurements of support, accuracy, and coverage. The accuracy is calculated by dividing the support of the decision attribute by the support of the conditional attributes. The coverage is calculated by dividing the support of the conditional attributes by the total number of objects.

For stock market analysis, objects in the data set could contain the same range of technical indicators, such as oscillating indicators having a wavelike pattern (Yao & Herbert, 2009). Thus, one

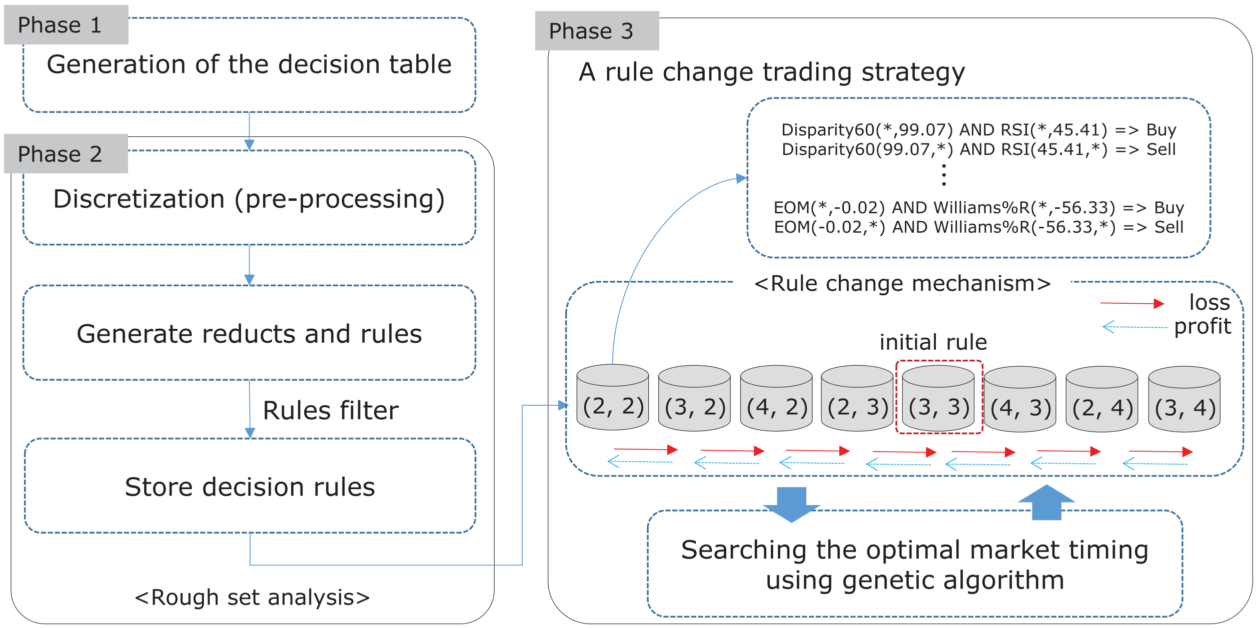

Fig. 1. The construction procedure of the proposed *RCTS*.

could generate decision rules by using rough sets based on the indiscernibility relation.

2.2. Genetic algorithm

A genetic algorithm (GA), which is based on a natural selection process that mimics biological evolution, was introduced by Holland (1975) and expanded by Goldberg (1989). GAs are well-known approaches to provide an optimal or near-optimal solution for optimization problems (Kim & Han, 2000; Oh et al., 2006). In particular, a GA is suitable for parameter optimization problems with an objective function, subject to some constraints (Shin & Han 1999). GAs and evolutionary systems have been applied in many fields for several years and have successfully proved to be a powerful tool for financial applications, including optimal trading rules, stock selection, and portfolio management, among others (Enke & Mehdiyev, 2013). The GA is implemented by following a specific evolution-based cycle. Each chromosome within the randomly initialized population is evaluated using a fitness function. By performing operations such as reproduction, crossover, and mutation on the chromosomes whose fitness have just been measured, a new population is created. The fitness of all chromosomes within the population are then evaluated. The aforementioned process is iterated until a best/acceptable chromosome is found (Wong & Tan, 1994).

In this study the GA is employed to optimize the thresholds of buying and selling signals in the rule change mechanism. To find a feasible solution, each chromosome, which is designed to denote the thresholds of buying or selling signals, is optimized utilizing the reproduction, crossover, and mutation operators.

3. A rule change trading system

The proposed trading system includes the three phases of the *Rule Change Trading System* (*RCTS*). The first phase generates input data that consist of condition attributes (i.e., technical indicators) and a decision attribute (i.e., Up or Down) to establish the decision table (or information system) for the rough set analysis. The second phase extracts and stores the decision rules by using rough set analysis within the trading rule repository. In the final phase, the rule change trading strategy is conducted to generate the optimal trading signals by the rule change mechanism within the GA. Fig. 1 presents the overall structure of the *RCTS*.

3.1. Phase 1: Generation of the decision table

The decision table scheme begins by generating the decision table, including conditional attributes $A = \{a_1, a_2, \cdots, a_l\}$ and a decision attribute $D = \{d\}$. The values of the conditional attributes are calculated using each formula of l technical indicators using real-time data consisting of open, high, low, and close price data, including trading volume during the time interval $(t-1, t]$. The value of the decision attribute is defined by the direction of the futures price, which is categorized as Down (-1) or Up (1). Down (-1) means that the next day's price is lower than the today's price, whereas Up (1) means that the next day's price is higher than today's price. Therefore, the decision table can be denoted by $S = (U, A \cup \{d\})$. For the analysis, various technical indicators $(l = 30)$ that are frequently used by experts and researcher were utilized (see the Appendix). Oscillating indicators that move up and down within a price range were considered to provide an indiscernibility relation for rough set theory.

3.2. Phase 2: Generate and store trading rule sets using rough set

Data pre-processing is first conducted by means of *discretization* as rough set analysis can handle discrete attributes. Generally, discretization refers to the process of converting or partitioning continuous attributes a_j $(j = 1, 2, \cdots, l)$ into discrete attributes through a set of cut points $P_j = \{P_{j1}, P_{j2}, \cdots, P_{jq}\}$, where P_{j1} and P_{jq} are the minimum and the maximum values of a_j, respectively. The intervals for a_j can be defined as $I = \{[P_{j1}, P_{j2}], (P_{j2}, P_{j3}], \cdots, (P_{jq-1}, P_{jq}]\}$ through the cut points (Rahman & Islam, 2016). The equal frequency discretization method is used in this study. This method first determines P_{j1} and P_{jq} of a_j, and then sorts all values in ascending order. The sorted values are assigned into k-intervals $(k = q - 1)$, so each interval contains the same number of objects (approximately N/k data, where N is the total number of objects) (Kotsiantis & Kanellopoulos, 2006).

Second, this step produces all possible reducts through the discretized data in the previous step. The generation of the reducts is an important step in rough set analysis since the reducts can result in obtaining a set of minimal attributes that discern a maximum number of objects through the decision table. The core information of the discretized data is used to generate the reducts, which are necessary for generating a specific rule. For this study, the manual reduces method, which enables an analyst to manually specify an attribute subset, is used to generate all the possible reduct combinations of technical indicators. For example, 3 (=cardinality of reducts) combinations of 10 attributes can generate 120 ($\leq \frac{10\times9\times8}{3\times2} = 120$) reducts. However, some reducts are filtered out because there is insufficient support of the data for generating decision rules.

Based on the reducts generated in Step 2, the decision rules are expressed in 'IF-THEN' form, which combines the conditional attributes with the decision attribute. A decision rule (r) can be expressed as

$$r: if\,(a_1, p^k_{a_1(i)}) \wedge (a_2, p^k_{a_2(i)}) \wedge \cdots \wedge (a_l, p^k_{a_l(i)})\ then\ (d)$$

where a_1, a_2, $\cdots$, a_l are conditional attributes (i.e., technical indicators) and $p^k_{a_1(i)}, p^k_{a_2(i)}, \cdots, p^k_{a_l(i)}$ denote the range of each corresponding technical indicator (a_1, a_2, $\cdots$, a_l) with ith interval ($i = 1, 2, \cdots, k$) among k intervals. For this study, insufficient decision rules are filtered out by measurements such as support (minimum 20% of the total number of each training dataset), coverage (>20%), and accuracy (>50%).

Thus, a set of generated decision rules $R(k, l)$ can be expressed as

$$R(k, l) = \{r_1, r_2, \cdots, r_n\}$$

where k is the number of intervals (k-intervals) for discretization and l is the cardinality of reducts. For example, $R(2, 2)$ is a set of decision rules generated by 2 intervals and 2 for the cardinality of reducts. For this study, decision rules are extracted by all combinations of k (=2, 3, 4) and l (=2, 3, 4). However, a set of decision rules $R(4, 4)$ is intentionally excluded from the trading rules repository because $R(4, 4)$ is so elaborate that almost all decision rules filter out. As a result, all extracted decision rules (8 total sets of decision rules) are stored in the trading rules repository.

3.3. Phase 3: A rule change trading strategy

3.3.1. A rule change mechanism

The stored decision rules in the trading rules repository from phase 2 need to be converted to trade the futures in the market. A rule change mechanism is proposed that measures the ratio of a set of decision rules to generate buying or selling signals. The ratio of a set of decision rules $R(k, l)$ at time t is calculated as follows:

$$RR_t(k, l) = \frac{S_t(k, l)}{N(k, l)}$$

where $N(k, l)$ is the total number of extracted decision rules $R(k, l)$ during the training data, and $S_t(k, l)$ is the sum of the value of satisfied decision rules $R(k, l)$ at time t. Note that RR_t close to 1 implies that many decision rules suggest a buying (or long position) signal. On the other hand, if RR_t is close to −1, this implies that many decision rules are suggesting a selling (or short position) signal. Therefore, a rule change trading strategy is implemented such that RR_t initiates a buying (selling) signal if RR_t is more (less) than a threshold, $\alpha(\beta)$.

The rule change mechanism actives if profit from the previous trade is positive (+), such that $RR_t(k, l)$ changes into $RR_t(k-1, l)$ or $RR_t(k, l-1)$ during the next trading. To the contrary, if profit from the previous trade is negative (-), $RR_t(k, l)$ changes into $RR_t(k+1, l)$ or $RR_t(k, l+1)$ for the next trade. Note that $RR_t(3, 3)$ is determined

Table 2
Training period and testing period.

Window number	Training period	Testing period
1	Sept. 14, 2007–Mar. 13, 2008	Mar. 14, 2008–Jun. 12, 2008
2	Dec. 14, 2007–Jun. 12, 2008	Jun. 13, 2008–Sept. 11, 2008
3	Mar. 14, 2008–Sept. 11, 2008	Sept. 12, 2008–Dec. 11, 2008
4	Jun. 13, 2008–Dec. 11, 2008	Dec. 12, 2008–Mar. 12, 2009
5	Sept. 12, 2008–Mar. 12, 2009	Mar. 13, 2009–Jun. 11, 2009
6	Dec. 12, 2008–Jun. 11, 2009	Jun. 12, 2009–Sept. 10, 2009
7	Mar. 13, 2009–Sept. 10, 2009	Sept. 11, 2009–Dec. 10, 2009
8	Jun. 12, 2009–Dec. 10, 2009	Dec. 11, 2009–Mar. 11, 2010
9	Sept. 11, 2009–Mar. 11, 2010	Mar. 12, 2010–Jun. 10, 2010
10	Dec. 11, 2009–Jun. 10, 2010	Jun. 11, 2010–Sept. 9, 2010
11	Mar. 12, 2010–Sept. 9, 2010	Sept. 10, 2010–Dec. 9, 2010
12	Jun. 11, 2010–Dec. 9, 2010	Dec. 10, 2010–Mar. 10, 2011
13	Sept. 10, 2010–Mar. 10, 2011	Mar. 11, 2011–Jun. 9, 2011
14	Dec. 10, 2010–Jun. 9, 2011	Jun. 10, 2011–Sept. 8, 2011
15	Mar. 11, 2011–Sept. 8, 2011	Sept. 9, 2011–Dec. 8, 2011
16	Jun. 10, 2011–Dec. 8, 2011	Dec. 9, 2011–Mar. 8, 2012
17	Sept. 9, 2011–Mar. 8, 2012	Mar. 9, 2012–Jun. 14, 2012
18	Dec. 9, 2011–Jun. 14, 2012	Jun. 15, 2012–Sept. 13, 2012
19	Mar. 9, 2012–Sept. 13, 2012	Sept. 14, 2012–Dec. 13, 2012
20	Jun. 15, 2012–Dec. 13, 2012	Dec. 14, 2012–Mar. 14, 2013
21	Sept. 14, 2012–Mar. 14, 2013	Mar. 15, 2013–Jun. 13, 2013
22	Dec. 14, 2012–Jun. 13, 2013	Jun. 14, 2013–Sept. 12, 2013
23	Mar. 15, 2013–Sept. 12, 2013	Sept. 13, 2013–Dec. 12, 2013
24	Jun. 14, 2013–Dec. 12, 2013	Dec. 13, 2013–Mar. 13, 2014

Table 3
Rule change strategies for *RCTS.*

Strategy	Type of trading rules
Fix	$R(3, 3)$
Discretization	$R(2, 3)$,$R(3, 3)$,$R(4, 3)$
Reducts	$R(3, 2)$,$R(3, 3)$,$R(3, 4)$
Combine	$R(2, 2)$,$R(3, 2)$,$R(4, 2)$,$R(2, 3)$,$R(3, 3)$,$R(4, 3)$,$R(2, 4)$,$R(3, 4)$
Combine with GA	$R(2, 2)$,$R(3, 2)$,$R(4, 2)$,$R(2, 3)$,$R(3, 3)$,$R(4, 3)$,$R(2, 4)$,$R(3, 4)$

as the initial trading rules by trial-and-error. A rule change trading strategy is described as follows:

The initial trading rule RR_t ($k = 3, l = 3$)
IF *RR_t is more than α*
  If the position is none $(t-1)$ then Long position
  Else If the position is short position $(t-1)$ then exit position and rule change
  Else Hold
Else IF *RR_t is less than β*
  If the position is none $(t-1)$ then Short position
  Else If the position is long position $(t-1)$ then exit position and rule change
  Else Hold
ELSE *No position*

3.3.2. Searching the optimal trading signals using genetic algorithms

Instead of using the Sharpe Ratio or the profit, a payoff ratio (also known as Profit/Loss ratio) is used in the training period as the fitness. The payoff ratio is defined as the system's average profit per trade divided by the average loss per trade (Wang et al., 2014). For the defined fitness function, the objective function of *RCTS* is denoted as *PR* and is calculated as follows:

$$Maximize \quad PR = \frac{\frac{\sum W}{N_W}}{\frac{\sum L}{N_L}} = \frac{\overline{W}}{\overline{L}}$$

where PR is the payoff ratio, ΣW (ΣL) is the total profit (loss) of winning (losing) trades, and N_W (N_L) is the number of winning

Table 4
Trading results of *RCTS-Fix* in the testing periods.

No. Window	Profit (pt)	No. trades	Win%	Profit factor	Payoff ratio	Buy and hold
1	−16.25	50	58.00	0.73	0.53	14.85
2	34.91	61	72.13	1.92	0.74	−36.35
3	37.10	61	73.77	1.29	0.46	−33.80
4	24.25	52	73.08	1.43	0.53	−5.05
5	4.35	69	69.57	1.08	0.47	35.35
6	7.45	115	51.30	1.13	1.07	32.20
7	1.05	1	100.00	–	–	1.00
8	1.90	54	51.85	1.05	0.98	1.45
9	1.50	1	100.00	–	–	−0.90
10	−0.35	74	64.86	0.99	0.54	14.95
11	−31.55	43	46.51	0.38	0.44	30.45
12	−1.35	47	76.60	0.96	0.29	−1.45
13	0.15	50	62.00	1.00	0.61	12.55
14	50.00	37	54.05	2.28	1.94	−35.00
15	−32.50	16	31.25	0.39	0.85	12.70
16	−28.15	53	35.85	0.59	1.05	12.15
17	25.80	109	54.13	1.43	1.21	−15.55
18	−25.95	29	41.38	0.57	0.81	9.30
19	7.00	64	59.38	1.19	0.81	8.30
20	−1.80	40	45.00	0.94	1.15	−1.85
21	−13.60	47	48.94	0.73	0.76	−17.50
22	−17.25	7	28.57	0.25	0.64	16.85
23	14.30	54	68.52	1.41	0.65	−3.40
24	3.35	1	100.00	–	–	−6.65
Average	1.85	47	61.11	0.91	0.66	1.86

Table 5
Trading results of *RCTS-Discretization* in the testing periods.

No. Window	Profit (pt)	No. trades	Win%	Profit factor	Payoff ratio
1	−4.20	57.00	64.91	0.92	0.50
2	3.66	58.00	65.52	1.08	0.57
3	−1.55	55.00	72.73	0.99	0.37
4	28.40	56.00	75.00	1.54	0.51
5	9.25	58.00	72.41	1.15	0.44
6	24.25	41.00	56.10	2.68	2.10
7	1.05	1.00	100.00	–	–
8	5.90	47.00	63.83	1.23	0.70
9	1.50	1.00	100.00	–	–
10	3.45	69.00	60.87	1.07	0.69
11	−31.80	46.00	39.13	0.39	0.60
12	2.10	49.00	69.39	1.06	0.47
13	−2.50	40.00	72.50	0.95	0.36
14	51.25	50.00	66.00	1.92	0.99
15	−10.60	10.00	30.00	0.67	1.57
16	−20.05	31.00	35.48	0.60	1.10
17	−16.65	27.00	40.74	0.50	0.73
18	−42.45	52.00	61.54	0.44	0.28
19	0.70	49.00	67.35	1.02	0.50
20	18.25	64.00	59.38	1.61	1.10
21	20.00	2.00	100.00	–	–
22	−18.60	2.00	50.00	0.20	0.20
23	11.00	37.00	56.76	1.42	1.08
24	3.35	1.00	100.00	–	–
Average	1.49	37	65.82	0.89	0.60

(losing) trades. Therefore, the buying α $(-1 < \alpha < 1)$ and selling β $(-1 < \beta < 1)$ threshold are optimized to maximize *PR* within the training data (i.e., $\alpha > \beta$). In addition, the number of trades $(N_W + N_L > 30)$ and the winning rate $(0.35 < \frac{N_W}{N_W+N_L} < 0.65)$ are restricted to be constraints.

When using GAs for the thresholds of buying and selling, the GA parameters are important because they affect the scope of the search space during the evolutionary process (Oh et al., 2006). Through preliminary experiments, the population size, crossover rate, and mutation rate were selected as 1000, 0.5, and 0.06, respectively. The number of generations was fixed at 500 and the GA stops when there is no improvement of 1% over the last 5000 trials.

4. Experimental results and analysis

4.1. Data collection and experimental setup

To develop and evaluate the *RCTS*, the 30-min Korea Stock Price Index 200 (KOSPI 200) futures contract data were utilized. Data was collected from the CHECKExpert terminal of the Korea Securities Computing Corporation (KOSCOM). To verify the *RCTS* robustness, a sliding window method was employed. The overall experimental period was from September 14, 2007 to March 13, 2014. A sliding window method based on the maturity date of the KOSPI 200 future contracts was designed to divide the training and testing periods, as shown in Table 2. The *RCTS* was trained

Table 6
Trading results of *RCTS-Reduct* in the testing periods.

No. Window	Profit (pt)	No. trades	Win%	Profit factor	Payoff ratio
1	−7.25	58	56.90	0.87	0.66
2	40.02	63	71.43	2.04	0.82
3	44.20	62	74.19	1.34	0.47
4	17.45	49	71.43	1.28	0.51
5	0.80	71	67.61	1.01	0.49
6	16.60	105	52.38	1.31	1.19
7	1.05	1	100.00	–	–
8	3.05	45	66.67	1.11	0.55
9	1.50	1	100.00	–	–
10	−3.65	71	63.38	0.94	0.54
11	−35.75	42	42.86	0.36	0.48
12	5.40	50	72.00	1.13	0.44
13	−6.10	55	56.36	0.90	0.70
14	55.85	41	58.54	2.45	1.73
15	−11.50	2	50.00	0.11	0.11
16	−24.15	52	32.69	0.63	1.30
17	26.35	101	55.45	1.45	1.17
18	−27.25	38	44.74	0.57	0.70
19	12.30	56	67.86	1.39	0.66
20	1.35	34	55.88	1.05	0.83
21	5.85	53	54.72	1.16	0.96
22	−18.60	2	50.00	0.20	0.20
23	14.10	45	68.89	1.47	0.66
24	3.35	1	100.00	–	–
Average	4.79	46	63.91	0.95	0.62

Table 7
Trading results of *RCTS-Combine* in the testing periods.

No. Window	Profit (pt)	No. trades	Win%	Profit factor	Payoff ratio
1	−3.00	57	59.65	0.95	0.64
2	9.37	60	65.00	1.18	0.64
3	28.30	56	71.43	1.21	0.48
4	18.80	52	67.31	1.32	0.64
5	−3.10	56	64.29	0.95	0.53
6	34.00	15	60.00	7.02	4.68
7	1.05	1	100.00	–	–
8	15.35	48	66.67	1.71	0.86
9	1.50	1	100.00	–	–
10	−14.10	51	47.06	0.66	0.74
11	−26.95	52	48.08	0.50	0.53
12	1.70	47	68.09	1.04	0.49
13	−0.50	57	70.18	0.99	0.42
14	36.60	52	53.85	1.50	1.28
15	−9.35	8	37.50	0.67	1.12
16	−0.45	29	41.38	0.99	1.40
17	−15.00	17	58.82	0.50	0.35
18	−36.05	50	66.00	0.50	0.26
19	5.85	54	74.07	1.16	0.41
20	−11.60	46	45.65	0.70	0.84
21	20.00	2	100.00	–	–
22	−18.60	2	50.00	0.20	0.20
23	14.85	22	68.18	1.75	0.82
24	3.35	1	100.00	–	–
Average	2.17	35	65.97	1.06	0.69

with a 6-month period, and tested with a 3-month period. For each subsequent experiment, both the training and the testing period were moved 3 months forward (for a total of 24 windows). The training and testing periods included trending up, trending down, and flat markets, providing an experimental data set that was appropriate for the performance evaluation.

In order to explore the impact of rule changes on the performance of the *RCTS*, this study examined the different rule change strategies shown in Table 3. The terms *Fix, Discretization, Reducts, Combine*, and *Combine with GA* identify the criteria of rule change. To compare the performance of the *RCTS* with these various rule change strategies, the net profit was calculated. Furthermore, the payoff ratio and a profit factor were considered for comparing trading results. The profit factor is the system's gross profit divided by the gross loss. This calculation indicates how much the profit exceeded the loss. The higher the payoff ratio and profit factor, the better the trading system (Wang et al., 2014).

The main aim of designing the experiments is to examine the usage of trading rules that were generated by combining the discretization and reducts based on rough set analysis, as compared against a buy-and-hold strategy, for the KOSPI 200 index futures market.

4.2. Experimental results

As described previously, both the trading rules extracted in the training period, along with the buying and selling signals, are generated by the extracted rules in the testing period, allowing

Table 8
Trading results of *RCTS-Combine with the GA* in the testing periods.

No. Window	Profit (pt)	No. trades	Win%	Profit factor	Payoff ratio	Thresholds (α,β)
1	−2.10	35	62.86	0.95	0.56	0.39, −0.41
2	50.88	48	77.08	2.84	0.84	0.79, −0.28
3	40.70	40	72.50	1.40	0.53	1.41, −1.01
4	27.75	38	65.79	1.63	0.85	0.72, −0.44
5	18.25	18	55.56	1.95	1.56	0.38, −0.19
6	1.95	65	53.85	1.05	0.90	0.41, −0.39
7	−1.55	24	66.67	0.94	0.47	0.23, −0.25
8	1.10	55	61.82	1.03	0.64	0.35, −0.34
9	−17.10	41	31.71	0.65	1.40	0.31, −0.49
10	−10.65	36	50.00	0.69	0.69	0.24, −0.35
11	3.35	18	50.00	1.19	1.19	0.21, −0.18
12	−2.50	16	62.50	0.89	0.53	0.20, −0.23
13	−5.75	43	67.44	0.91	0.44	0.58, −0.64
14	−21.65	32	65.63	0.77	0.40	0.72, −0.94
15	−8.55	21	57.14	0.84	0.63	0.44, −0.53
16	3.20	43	41.86	1.08	1.50	0.44, −0.41
17	4.50	22	27.27	1.14	3.03	0.37, −0.33
18	7.50	3	33.33	3.05	6.11	0.11, −0.04
19	20.20	30	73.33	2.22	0.81	0.37, −0.17
20	5.50	38	71.05	1.20	0.49	0.32, −0.27
21	12.35	47	59.57	1.40	0.95	0.43, −0.31
22	−7.90	35	62.86	0.77	0.46	0.26, −0.34
23	10.95	24	62.50	1.45	0.87	0.35, −0.25
24	10.10	9	66.67	2.49	1.24	0.17, −0.07
Average	5.86	33	58.29	1.36	1.11	

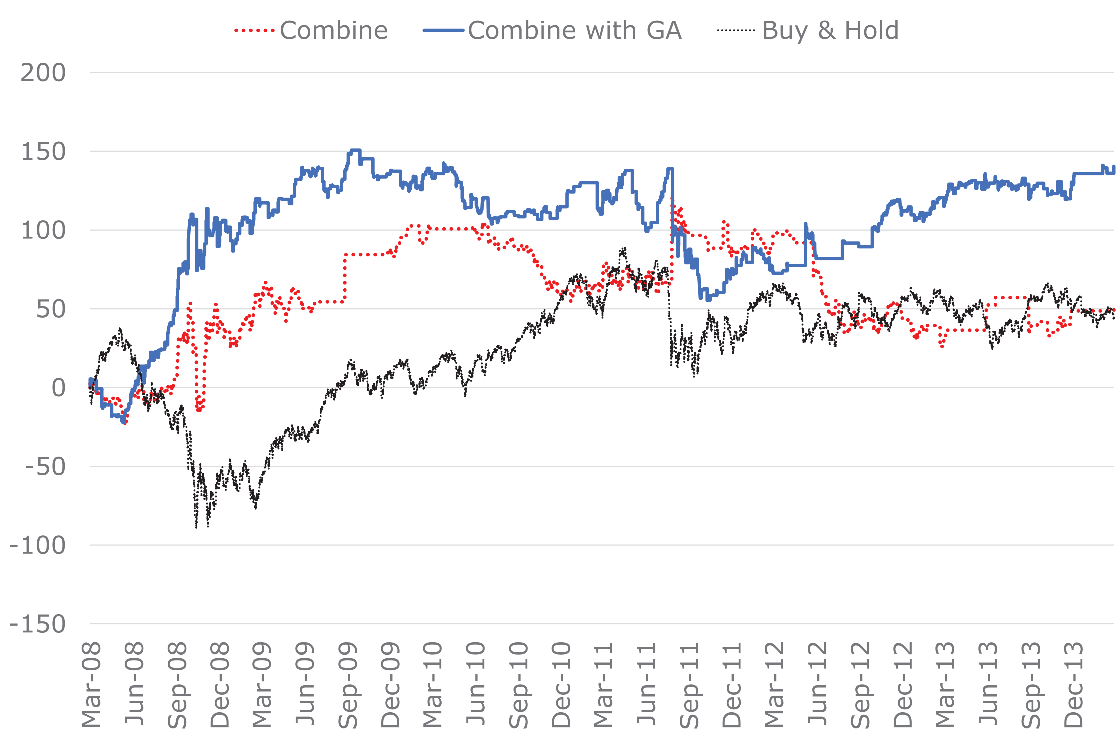

Fig. 2. Cumulative profit for *RCTS-Combine* and *RCTS-Combine* with *GA* in the testing period.

both the effectiveness of the rule change strategy and the trading results to be analyzed. The rule change mechanism is examined for the four strategies that may influence trading performance (*RCTS–Fix, -Discretization, -Reducts, -Combine*). Furthermore, a trading simulation was performed to examine whether the GA can help to improve the *RCTS* in order to acquire higher profits.

The results are compared against the performance generated from *RCTS-Fix, -Discretization, -Reducts, -Combine*, as well as the buy-and-hold trading strategy. Tables 4–7 present the profitability results over the testing periods, including the profit of the trading based on one contract of the KOSPI 200 futures, the percentage of winning trades (Win%), and the number of trades (No. trades). During the testing periods, the average profit of the buy-and-hold strategy was 1.86pt (standard deviation: 19.16), with total profit of 44.60pt (see Table 4 where 1pt = ₩500,000 (USD $500)). It can be seen in the tables that *RCTS* (*-Reduct, -Combine, -Combine with GA*) based on rough sets analysis performs better than the buy-and-hold strategy since it generates more total profit and average profit. This result indicates that the rough set approach is useful for extracting trading rules, as well as illustrates the benefits of the rule change mechanism.

Tables 4–7 also show that *RCTS-Reducts* generate the highest total profit (114.97pt) and average profit (4.79pt, standard deviation: 19.33), while *RCTS-Discretization* produced the lowest total profit (35.71pt) and average profit (1.49pt, standard deviation: 21.43) among the rule change strategies. This result indicates that the *RCTS-Reducts* strategy is a more efficient use of the rule change mechanism, and that the rule change mechanism is appropriate to change the cardinality of reducts more than changing the number of discretization. When compared with *RCTS-Combine, RCTS-Reducts* generated higher total profit and average profit. However, although the *RCTS-Combine* strategy could not generate higher total profit (52.02pt) and average profit (2.17pt, standard deviation: 17.76) as compared to the *RCTS-Reducts* strategy for the defined testing period, it is a profitable strategy when a profit factor measure is applied. The *RCTS-Combine* strategy obtained an average profit factor of 1.06, which is higher than that generated by the *RCTS-Reducts* (0.95). A profit factor greater than 1 implies that the strategy is profitable, while less than 1 implies a profitless strategy (Stridsman, 2001). Therefore, although the trading strategy based on *RCTS-Combine* had a lower total profit, it can earn more profit per unit of loss in all market conditions.

However, these strategies have one problem: they always have a position due to the thresholds ($\alpha = \beta = 0$). In some testing periods (see Tables 4–7), trading measures such as the profit factor and payoff ratio are unable to be evaluated since the rule change mechanism does not activate. Thus, these trading strategies may not be practical for all traders. Therefore, a genetic algorithm was used to search the threshold for market timing (buying and selling signals) to obtain better trading performance. The results of the *RCTS combine with GA* are quite impressive, generating improved trading performance in all aspects (see Table 8). This approach generally outperforms the benchmark strategies, both in terms of overall profitability with an average profit of 5.86pt (standard deviation: 16.45) and total profit of 140.53pt, and also in terms of risk performance with an average profit factor of 1.36. In spite of the relative low percentage of wining trades, the average profit and total profit are the highest, and the profit factor and payoff ratio are also improved. This indicates that the threshold searched from the *RCTS-Combine with GA* system helps to make better trading decisions when market timing. These results suggest that *RCTS-Combine with GA* outperforms, and that this strategy is better suited for the rule change mechanism during practical trading. A comparison of the trading performance results is also presented in Fig. 2.

When a GA was applied to search the thresholds of *RCTS-Combine*, the system achieved the best trading results over the testing periods. This demonstrates that the rule change mechanism can help to eliminate whipsaw signals, while the GA can help to provide the correct buying and selling signals.

5. Concluding remarks

For this study a unique rule change mechanism for usage with decision rules generated by applying rough set analysis on the futures market was proposed. The resulting rule change trading system, or RCTS, which consists of trading rules combined with reducts by changing numerous technical trading rules generated by rough set analysis, was developed and tested utilizing real data from the KOSPI 200 index futures market to prove its practicability. Testing has shown that the system provides impressive profits with both a high average profit factor and a high payoff ratio. The results also showed that the proposed *RCTS-Combine with GA* system can help to provide optimal trading decisions to generate better profitability when compared to systems using *RCTS-Fix, -Discretization, -Reducts*, and *-Combine*, as well as a traditional buy-and-hold trading strategy. Often, systems using generated trading rules based on rough sets analysis, which may consider too many intervals for discretization or cardinality of reducts, makes them unsuited for trading given that the trading rules are too elaborate or optimal to identify trading opportunities. Furthermore, trading rules that generate too few intervals or cardinality of reducts can risk losing core information. The proposed system provides a successful approach to deal with both issues.

For further study, the trading performance of the proposed system could possibly be improved by considering other combinations of data discretization methods, or by considering other intelligent techniques for generating trading rules. In addition, the proposed trading system can be simulated in different futures markets, such as the commodity and foreign exchange (FX) markets, to study both its robustness and to identify any changes that might be required for such markets.

Appendix. Oscillators

Notes: H, L, C, and *V* denote the high, low, closing price and volume during $(t-1, t]$, respectively. Further define $y_{n,\max} = \max(y_t, \cdots, y_{t-n+1})$,$y_{n,\min} = \min(y_t, \cdots, y_{t-n+1})$,$\bar{y}(n) = \frac{1}{n}\sum_{i=0}^{n-1} y_{t-i}$. MA means moving average; EMA means exponential moving average.

Oscillator	Formula	Values
%D (Stochastics %D)	$\%D(n) = \sum_{i=0}^{n-1} \%K_{t-i}/n$	$n = 12$
%K (Stochastics %K)	$\%K(n) = \frac{C-L_{n,\min}}{H_{n,\max}-L_{n,\min}} \times 100$	$n = 12$
Band %b	$Band\ \%b(n) = \frac{C-L(n)}{U(n)-L(n)}$	
	$U(n) =$	$n = 20$
	$\bar{C}(n) + [\alpha \times \sqrt{\frac{\sum_{i=0}^{n-1}(C_{t-i}-\bar{C}(n))^2}{n}}]$	
	$L(n) =$	$\alpha = 2$
	$\bar{C}(n) - [\alpha \times \sqrt{\frac{\sum_{i=0}^{n-1}(C_{t-i}-\bar{C}(n))^2}{n}}]$	
Band width	$Band\ width(n) = \frac{U(n)-L(n)}{\bar{C}(n)}$	$n = 20$
CCI (Commodity Channel Index)	$CCI(n) = \frac{M-\bar{M}(n)}{d(n)\times 0.015}$	
	$M = \frac{H+L+C}{3}$	$n = 9$
	$d(n) = \frac{1}{n}\sum_{i=0}^{n-1} \lvert M_{t-i} - \overline{M}_t(n)\rvert$	
CO (Chaikin's Oscillator)	$CO(m, n) =$ $EMA(AD, m) - EMA(AD, n)$	$m = 3$
	$AD = AD_{t-1} + (\frac{(C-L)-(H-C)}{(H-L)}) \times V$	$n = 10$
Disparity	$Disparity(n) = \frac{C}{MA(C,n)} \times 100$	$n = 20, 60$
DPO (Detrended Price Oscillator)	$DPO(n) = C - MA(C, (n/2) + 1)$	$n = 14$
EOM (Ease of Movement)	$EOM = (\frac{H+L}{2} - \frac{H_{t-1}+L_{t-1}}{2})/\frac{V}{H-L}$	
MACD (Moving Average Convergence-Divergence)	$MACD(m, n) =$ $EMA(C, m) - EMA(C, n)$	$m = 12$
		$n = 26$
MAO (MA Oscillator)	$MAO(m, n) =$ $MA(C, m) - MA(C, n)$	$m = 5$
		$n = 20$
MFI (Money Flow Index)	$MFI = 100 - \frac{100}{(100+MR)}$	
	$TP = \frac{H+L+C}{3}$, $MF = TP \times V$	
	$MR = \frac{Positive\ MF}{Negative\ MF}$	
MI (Mass Index)	$MI(n) = \sum_{i=0}^{n-1} \frac{EMA_{t-i}(r,n)}{EMA^2_{t-i}(r,n)}$, $r = H - L$	$n = 9$
Momentum	$Momentum(n) = \frac{C}{C_{t-n}} \times 100$	$n = 10$
NCO (Net Change Oscillator)	$NCO(n) = C - C_{t-n}$	$n = 12$
PO (Price Oscillator)	$PO(m, n) = \frac{MA(C,m)-MA(C,n)}{MA(C,m)}$	$m = 5$
		$n = 10$

(*continued on next page*)

Oscillator	Formula	Values
Psychology	$Psy(n) = \frac{\text{the number of up trend}}{n}$	$n = 10$
ROC (Rate of Change)	$ROC(n) = (\frac{C}{C_{t-n}} - 1) \times 100$	$n = 12$
RSI (Relative Strength Index)	$RSI(n) = 100 - \frac{100}{1+RS(n)}$ $RS(n) = \sum_{i=0}^{n-1} Up_{t-i} / \sum_{i=0}^{n-1} Down_{t-i}$ where $Up_t(Down_t)$ is upward (downward) price change	$n = 14$
S-ROC	$SROC(m, n) = \frac{EMA(C,n)}{EMA(C,m)} \times 100$	$m = 10$ $n = 20$
Slow%d (Stochastics slow%d)	$\%d(n) = \sum_{i=0}^{n-1} \%D_{t-i}/n$	$n = 12$
SO (Stochastics Oscillator)	$SO(n) = \sum_{i=0}^{n-1} \%K_{t-i} \times 100$	$n = 12$
Sonar	$Sonar(n) =$ $EMA(C, n) - EMA_{t-n}(C, n)$	$n = 25$
Sonar Signal	$Sonar\ Signal(m, n) =$ $EMA(Sonar(n), m)$	$m = 9$ $n = 25$
TRIX	$TRIX(n) = \frac{EMA^3(C,n) - EMA^3_{t-1}(C,n)}{EMA^3_{t-1}(C,n)}$	$n = 12$
VO (Volume Oscillator)	$VO(m, n) = \frac{\overline{V}(m) - \overline{V}(n)}{\overline{V}(n)} \times 100$	$m = 12$ $n = 26$
VROC (Volume Rate of Change)	$VROC(n) = (\frac{V_t}{V_{t-n}} - 1) \times 100$	$n = 14$
Williams%R	$Williams\%R(n) =$ $\frac{H_{n,\max} - C}{H_{n,\max} - L_{n,\min}} \times 100$	$n = 14$
Z-score	$Zscore(n) = \frac{C - MA(n)}{SD(C,n)}$ where $SD(C, n)$ is the standard deviation of closing prices for n periods	$n = 20$

References

Bagheri, A., Peyhani, H. M., & Akbari, M. (2014). Financial forecasting using ANFIS networks with quantum-behaved particle swarm optimization. *Expert Systems with Applications, 41*, 6235–6250.

Bao, D. (2008). A generalized model for financial time series representation and prediction. *Applied Intelligent, 29*, 1–11.

Bao, D., & Yang, Z. (2008). Intelligent stock trading system by turning point confirming and probabilistic reasoning. *Expert Systems with Applications, 34*, 620–627.

Barak, S., Dahooie, J. H., & Tichý, T. (2015). Wrapper ANFIS-ICA method to do stock market timing and feature selection on the basis of Japanese Candlestick. *Expert Systems with Applications, 42*, 9221–9235.

Bogullu, V. K., Enke, D., & Dagli, C. (2002). Using neural networks and technical indicators for generating stock trading signals. *Intelligent Engineering Systems Through Artificial Neural Networks, 12*, 721–726.

Booth, A., Gerding, E., & McGroarty, F. (2014). Automated trading with performance weighted random forests and seasonality. *Expert Systems with Applications, 41*, 3651–3661.

Cervelló-Royo, R., Guijarro, F., & Michniuk, K. (2015). Stock market trading rule based on pattern recognition and technical analysis: Forecasting the DJIA index with intraday data. *Expert Systems with Applications, 42*, 5963–5975.

Chavarnakul, T., & Enke, D. (2008). Intelligent technical analysis based equivolume charting for stock trading using neural networks. *Expert Systems with Applications, 34*, 1004–1017.

Chavarnakul, T., & Enke, D. (2009). A hybrid stock trading system for intelligent technical analysis-based equivolume charting. *Neurocomputing, 72*, 3517–3528.

Chen, T. L., & Chen, F. Y. (2016). An intelligent pattern recognition model for supporting investment decisions in stock market. *Information Sciences, 346*, 261–274.

Cheng, C.-H., Chen, T.-L., & Wei, L.-Y. (2010). A hybrid model based on rough sets theory and genetic algorithms for stock price forecasting. *Information Sciences, 180*, 1610–1629.

Enke, D., & Mehdiyev, N. (2013). Stock market prediction using a combination of stepwise regression analysis, differential evolution-based fuzzy clustering, and a fuzzy inference neural network. *Intelligent Automation and Soft Computing, 19*(4), 636–648.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*. Boston, MA: Addison-Wesley.

Holland, J. H. (1975). Adaptation in natural and artificial systems. University of Michigan Press.

Kim, K., & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications, 19*(2), 125–132.

Kotsiantis, S., & Kanellopoulos, D. (2006). Discretization techniques: A recent survey. *GESTS International Transactions on Computer Science and Engineering, 32*(1), 47–58.

Lee, S. J., Ahn, J. J., Oh, K. J., & Kim, T. Y. (2010). Using rough set to support investment strategies of real-time trading in futures market. *Applied Intelligence, 32*, 364–377.

Lee, S. J., Oh, K. J., & Kim, T. Y. (2012). How many reference patterns can improve profitability for real-time trading in futures market? *Expert System with Applications, 39*, 7458–7470.

Mabu, S., Hirasawa, K., Obayashi, M., & Kuremoto, T. (2013). Enhanced decision making mechanism of rule-based genetic network programming for creating stock trading signals. *Expert Systems with Applications, 40*, 6311–6320.

Menkhoff, L. (2010). The use of technical analysis by fund managers: International evidence. *Journal of Banking & Finance, 34*(11), 2573–2586.

Murphy, J. J. (1999). *Technical analysis of the financial markets*. Paramus, NJ: New York Institute of Finance.

Nguyen, S. H., & Skowron, A. (1995). Quantization of real value attributes—Rough set and Boolean reasoning approach. In *Proceedings of the second joint annual conference on information sciences, Wrightsville Beach, NC* (pp. 34–37).

Oh, K. J., Kim, T. Y., Min, S.-H., & Lee, H. Y. (2006). Portfolio algorithm based on portfolio beta using generic algorithm. *Expert Systems with Applications, 30*, 527–534.

Øhrn, A. (1999). *Discernibility and rough sets in medicine: Tools and applications. PhD thesis*. Trondheim: Norwegian University of Science and Technology.

Pardo, R. (2008). *The evaluation and optimization of trading strategies*: Vol. 314. Hoboken, NJ: Wiley.

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock market index using fusion of machine learning techniques. *Expert Systems with Applications, 42*, 2162–2172.

Pawlak, Z. (1982). Rough sets. *International Journal of Computer and Information Sciences, 11*, 341–356.

Pawlak, Z. (1997). Rough set approach to knowledge-based decision support. *European Journal of Operational Research, 99*(1), 48–57.

Pawlak, Z., Grzymala-Busse, J., Sloinski, R., & Ziarko, W. (1995). Rough set. *Communications of the ACM, 38*(11), 88–95.

Peters, E. E. (1994). *Fractal market analysis: Applying chaos theory to investment and economics*. New York: Willey & Sons.

Pring, M. (1991). *Technical analysis explained: The successful investor's guide to spotting investment trends and turning points*. New York, NY: McGraw-Hill.

Rahman, M. G., & Islam, M. Z. (2016). Discretization of continuous attributes through low frequency numerical values and attribute interdependency. *Expert Systems with Applications, 45*, 410–423.

Rather, A. M., Agarwal, A., & Sastry, V. N. (2015). Recurrent neural network and a hybrid model for prediction of stock returns. *Expert Systems with Applications, 42*, 3234–3241.

Shen, L., & Loh, H. T. (2004). Applying rough sets to market timing decisions. *Decision Support Systems, 37*, 583–597.

Shin, K. S., & Han, I. (1999). Case-based reasoning supported by genetic algorithms of corporate bond rating. *Expert Systems with Applications, 16*, 85–95.

Slowinski, R., Zopounidis, C., & Dimitras, A. I. (1997). Prediction of company acquisition in Greece by mean of the rough set approach. *European Journal of Operational Research, 100*(1), 1–15.

Stridsman, T. (2001). *Trading systems that work: Building and evaluating effective trading systems*. New York, NY: McGraw Hill Professional.

Tay, F. E. H., & Shen, L. (2002). Economic and financial prediction using rough sets model. *European Journal of Operational Research, 141*, 641–659.

Thawornwong, S., Enke, D., & Dagli, C. (2001). Using neural networks and technical analysis indicators for predicting stock trends. *Intelligent Systems Through Artificial Neural Networks, 11*, 739–744.

Vanstone, B., & Tan, C. N. W. (2003). A survey of the application of soft computing to investment and financial trading. In *Proceedings of the eighth Australian & New Zealand intelligent information systems conference (ANZIIS 2003)* December 10–12.

Vanstone, B., & Tan, C. N. W. (2005). Artificial neural networks in financial trading. In M. Khosrow-Pour (Ed.), *Encyclopedia of information science and technology* (pp. 163–167). Hershey, PA: Idea Group.

Wang, F., Yu, P. L. H., & Cheung, D. W. (2014). Combining technical trading rules using particle swarm optimization. *Expert Systems with Applications, 41*, 3016–3026.

Wong, F., & Tan, C. (1994). Hybrid neural, genetic and fuzzy systems. In G. J. Devoeck (Ed.), *Trading on the edge* (pp. 243–261). New York: John Wiley.

Yao, J., & Herbert, J. P. (2009). Financial time-series analysis with rough sets. *Applied Soft Computing, 9*, 1000–1007.