JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 30, 571-585 (2014)

DOI:10.1688/JISE.2014.30.3.2

A Novel Episode Mining Methodology for Stock Investment*

YU-FENG LIN[1], CHIEN-FENG HUANG[2] AND VINCENT S. TSENG[1]
[1]*Department of Computer Science and Information Engineering*
National Cheng Kung University
Tainan, 701 Taiwan
E-mail: tsengsm@mail.ncku.edu.tw
[2]*Department of Computer Science and Information Engineering*
National University of Kaohsiung
Kaohsiung, 811 Taiwan

In this paper, we present a novel methodology for stock investment using episode mining and technical indicators. The time-series data of stock price and the derived moving average, a class of well-known technical indicators, are used for the construction of complex episode events and rules. Our objective is to devise a profitable episode-based investment model to mine associated events in the stock market. Using Taiwan Capitalization Weighted Stock Index (TAIEX), the empirical results show that our proposed model significantly outperforms the benchmark in terms of cumulative total returns. We also show that the level of the precision by our model is close to 60%, which is better than random guessing. Based upon the results obtained, we expect this novel episode-based methodology will advance the research in data mining for computational finance and provide an alternative to stock investment in practice.

Keywords: episode mining, technical indicators, stock investing strategy, complex event sequence, cross validation

1. INTRODUCTION

One major goal of investment in the stock market is to achieve above-average returns while maintaining acceptable risk levels along the course of investment. This objective has been recognized as a difficult task in research and practice. Researchers in Computer Science have been using various soft computing and data mining models in order to gain useful insight and information for investment. For instance, Quah and Srinivasan [1] employed a neural network (NN) model for stock selection to choose stocks that are top-ranked performers. Kim and Han [2] proposed a genetic algorithm (GA) approach to feature discretization and the determination of connection weights for a NN model to predict the stock price index. Caplan and Becker [3] employed genetic programming (GP) to develop a stock ranking model for the high technology manufacturing industry in the U.S. Chu *et al.* used fuzzy multiple attribute decision analysis to select stocks for portfolio construction [4]. More recently, Huang [5] and Huang *et al.* [6] developed support vector regression and fuzzy theory-based models, respectively, for the determination of relative quality of stocks; and GA was employed for simultaneous treatment of feature selection and optimization of model parameters, which was shown to be effective for stock selection.

Time-series forecasting of stock price is a different line of research in the invest-

Received February 28, 2013; accepted June 15, 2013.
Communicated by Hung-Yu Kao, Tzung-Pei Hong, Takahira Yamaguchi, Yau-Hwang Kuo, and Vincent Shin-Mu Tseng.
* This research was supported by National Science Council, Taiwan, under Grant No. NSC101-2221-E-006-255-MY3.

ment area. As the Efficient Market Hypothesis (EMH) [7] asserts that a stock's price incorporates all the information available to investors, the implication is that predicting the price movement of a stock is impractical. Instead of precise prediction for stock prices, researchers tend to develop various tools to reveal patterns in the movement of stock price. In recent years, episode mining has emerged as a new technique for this purpose. For instance, Ng and Fu [8] proposed a method that extracts events from the financial news of Chinese newspapers to mine frequent episodes for discovering the relationship between news and stock market movements. The basic idea was to classify various semantic types of events from news and generate episodes by a combination of multiple event types. If the same episode occurs frequently, it is a frequent episode; and the goal was to identify frequent episodes for relevant stock market movements. While the proposed idea in [8] is interesting, the method only considered mining frequent episodes about the relationship between events of financial news and the rise or fall in the stock price. If these frequent episodes can be used to further generate episode rules (ER) for the detection of patterns hidden in the stock price movement, it shall be more useful for stock investing.

Dattasharma and Tripathi [9] proposed another episode-mining method using events and episodes to encode the time series data of stocks using binary alphabet by which one can calculate the similarity between two stocks using string distance metrics. However, the method in [9] cannot be applied readily to complex criteria that used more than one event in the time series. Dattasharma *et al*. [10] then proposed an improved method in order to generate multiple events and combine them effectively. Given a user defined criterion involving multiple events, they showed that it is possible to encode any stock's time series data using a finite alphabet.

Although Dattasharma *et al.* [10] proposed an interesting framework for modeling multi-event episodes using the time series data of stock price, in their method one has to specify multi-event episodes beforehand and determine if these episodes are frequent. However, in the real-world financial applications one may not be aware of what the multi-event episodes are in advance, so that the method in [10] is not directly applicable to these problems. In light of this deficiency, we propose a new episode-mining method to model directly with complex event sequences in the stock market. Our objective is to devise an episode-based investment model that is able to automatically determine multi-event episodes and associated profitable complex events embedded in the stock price data. Using the time-series data of stock price and the derived moving average [11], a class of well-known technical indicators, are used for the construction of complex episode events and rules. By a proper integration of methods for event extraction, technical indicators as well as statistical validation, we show that our method outperforms the benchmark significantly through empirical evaluation using Taiwan Capitalization Weighted Stock Index (TAIEX).

The remainder of this paper is organized as follows. We provide relevant definitions and discuss related works in Section 2. Session 3 presents our proposed approach. Then, the empirical results are shown in Session 4. Session 5 concludes the paper.

2. RELATED WORKS AND DEFINITIONS

2.1 Problem Definition

We begin this subsection by introducing several relevant definitions for episode dis-

covery using sliding windows [12, 13].

Let $\varepsilon = \{E_1, E_2, \ldots, E_m\}$ be a finite set of event types, and an event is a pair (E, T), where $E \in \varepsilon$ and $T \in \mathbb{N}$. A simple event sequence is denoted as $SES = \langle (E_1, T_1), \ldots, (E_2, T_2), \ldots, (E_n, T_n) \rangle$ such that $T_i < T_j$ for all $1 \le i < j \le n$.

Definition 1 (*Simultaneous event set*) A simultaneous event set, is denoted as $SE = (E_1, E_2, \ldots, E_m)$ such that $E_i \in \varepsilon$ for all $1 \le i \le m$, means each event in SE occurs at the same time point.

Definition 2 (*Complex Event Sequence*) The complex event sequence is denoted as $CES = \langle (SE_1, T_1), \ldots, (SE_2, T_2), \ldots, (SE_n, T_n) \rangle$ such that $T_i < T_j$ for all $1 \le i < j \le n$. The length of CES is denoted by $|CES|$ and is equal to the number of time points in CES.

Definition 3 (*Episode*) An episode α is an ordered collection of simultaneous event set with the form $\langle SE_1, SE_2, \ldots, SE_k \rangle$, where SE_i appears before SE_j for all $1 \le i < j \le k$.

Definition 4 (*Window Size*) A window size is a number of *WinS* consecutive records in a complex event sequence. Given a complex event sequence *CES*, the number of windows is equal to $(|CES| - WinS + 1)$.

Definition 5 (*Support/Frequent Episode*) Given a complex event sequence *CES*, the support of an episode α in *CES*, denoted by $sup(\alpha)$, is equal to the number of windows which contain events in the episode. An episode is called frequent, iff its support is no less than minimum support threshold min_{sup}.

Definition 6 (*Prefix and Suffix of an Episode*) Given an episode $\alpha = \langle SE_1, SE_2, \ldots, SE_k \rangle$, the suffix of α, denoted by $suffix(\alpha)$, is defined as an episode composed only by the last element, *i.e.*, $suffix(\alpha) = \langle SE_k \rangle$. The prefix of α is defined as $prefix(\alpha) = \langle SE_1, SE_2, \ldots, SE_{k-1} \rangle$.

Definition 7 (*Episode rule/Confidence*) Assume α and β are episodes such that $prefix(\beta) = \alpha$. The confidence of an episode rule $\alpha \rightarrow suffix(\beta)$is defined as

$$confidence = \frac{support(\beta)}{support(\alpha)}.$$

Problem statement of mining episode rules Given a complex event sequence and a user-specified minimum confidence threshold *min_conf*, the aim of mining episode rules is at discovering all the episode rules with confidence no less than *min_conf*.

2.2 Background

Frequent episode mining in event sequence is an important and fundamental technique in knowledge discovery area. Episode mining, first introduced by Mannila *et al.* [12], aims to find the relationships between events, and episode rules are used to model associations between occurrence orders of events. Several efficient methods have been

proposed for mining significant episodes or episode rules in event sequences [13-15]. The application areas of episode mining include, for example, telecommunication network alarms [16, 17], occurrences of recurrent illnesses [18, 19], financial data [8-10], to name a few.

Technical indicators have been frequently used for the construction of trading models. These methods typically extract trading signals from prices or volumes of stocks. Moving average (abbreviated as *MA*) [11] is a well-known technical indicator, which is used to get rid of the short-term fluctuations in order to reveal the long-term trends hidden in the signals. Since these long-term trends may consist of informative patterns such as bearish or bullish signs of the market, these patterns can be further used to generate episode rules. In this study, we thus propose to use MA as a component for our episode mining models.

Here we propose a methodology, Stock Investment Strategy using Technical indicator and Episode Mining (abbreviated as *SISTEM*), to construct frequent episode rules for stock investment. *SISTEM* is composed of three ingredients – (1) the extraction of events from stock price data; (2) the employment of episode mining techniques to discover frequent episodes and episode rules; and (3) the construction of investment strategies using the episode rules generated.

3. PROPOSED METHOD

In this section, we describe the details of our proposed *SISTEM* approach for stock investing. As displayed in Fig. 1, *SISTEM* consists of three components: events extraction, episode mining, and investment strategy.

Fig. 1. The framework of *SISTEM*.

In the following, we first show how to extract events from stock data and convert these events to complex event sequences. Then, we describe how to mine frequent episode rules and the design of investment strategies is given in Sections 3.2 and 3.3, respectively.

3.1 Event Extraction

The stock data we use here is a time series dataset that describes a stock's opening price, highest price, lowest price and closing price in each day. Table 1 shows an example for the price data of Taiwan Capitalization Weighted Stock Index (TAIEX). Events can be extracted from these prices by applying technical indicators, which can be further transformed into complex event sequence as shown in Fig. 2.

For example, assuming there are four types of prices at each of the 10 consecutive time points in Table 1, a basic event type, as shown in Table 2, can be defined as a 4-symbol code such as *COSy*, *OCDy*, *CCDy*, and so forth, according to the price variables at the same time or consecutive time points. The first two elements of the code indicate the price variables to be used for the later calculation of price difference. For instance, symbol *C*, *H*, *L* and *O* correspond to the closing price, the highest price, the lowest price and the opening price, respectively. The third element of the code consists of two symbols, *S* and *D*, which indicates whether the price difference is calculated using the same time (*S*) or the consecutive time points (*D*), respectively. The fourth element of the code consists of two outcomes, 1 and 0, which indicate whether the price difference is greater than or less than 0, respectively. For example, in our case *OCD*0 means that the difference between the closing price at time *i* and the opening price at time $(i - 1)$ is less than zero. Analogously, *OCD*1 means the difference between these two prices is larger than zero.

Here we also employ the moving average (abbreviated as *MA*) to construct complex event sequences. The definition of the moving average is as follows.

Table 1. Example for the price data of Taiwan weighted index.

*i*th time point	Date	Opening Price	Highest Price	Lowest Price	Closing Price
1	12/26/2012	7646.51	7683.09	7633.83	7634.19
2	12/27/2012	7634.8	7661.01	7622.73	7648.41
3	12/28/2012	7700.87	7708.31	7665.04	7699.5
4	01/02/2013	7738.05	7793.48	7715.26	7779.22
5	01/03/2013	7826.34	7855.16	7815.29	7836.84
6	01/04/2013	7818.27	7818.27	7773.06	7805.99
7	01/07/2013	7797.05	7797.05	7725.47	7755.09
8	01/08/2013	7737.07	7751.8	7692.98	7721.66
9	01/09/2013	7726.4	7763.97	7703.23	7738.64
10	01/10/2013	7780.98	7825.06	7760.09	7811.64

Definition 8 (*Moving Average*) Let P_t be the price of a stock at time t. A moving average at time t, the mean of the prices of the most recent k time periods, is defined as

$$MA_{t,k} = \frac{1}{k}\sum_{i=0}^{k-1} P_{t-i}.$$

For example, in our case the event of *MA* is defined as

$$\begin{cases} MA_k 0, \text{ if } (MA_{i,k} - MA_{i-1,k}) \le 0 \\ MA_k 1, \text{ if } (MA_{i,k} - MA_{i-1,k}) > 0 \end{cases}, k \le i \le T_n.$$

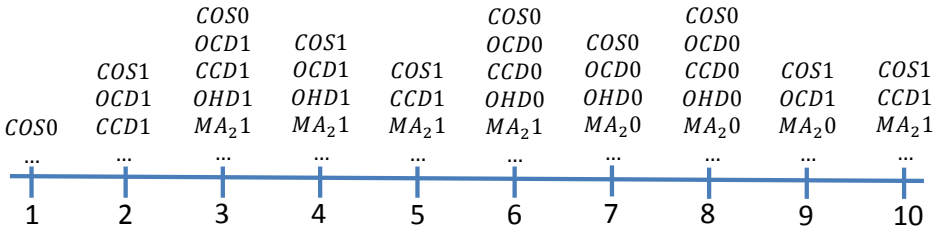

Fig. 2. An example for complex event sequence.

Table 2. Basic event types from Taiwan weighted index.

		*i*th time point				(i – 1)th time point			
		O	*H*	*L*	*C*	*O*	*H*	*L*	*C*
*i*th time point	*O*	–	–	–	–	*OODy*	*OHDy*	*OLDy*	*OCDy*
	H	*HOSy*	–	–	–	*HODy*	*HHDy*	*HLDy*	*HCDy*
	L	*LOSy*	*LHSy*	–	–	*LODy*	*LHDy*	*LLDy*	*LCDy*
	C	*COSy*	*CHSy*	*CLSy*		*CODy*	*CHDy*	*CLDy*	*CCDy*

As an illustration, in Fig. 2, k for Definition 8 is set to 2. Then using the closing price, *MA* at time 2 is $MA_{2,2} = \frac{7634.19+7648.41}{2} = 7641.3$; and *MA* at time 3 is $MA_{3,2} = \frac{7648.41+7699.5}{2} = 7679.955$. Because $MA_{3,2}$ is larger than $MA_{2,2}$, the event of *MA* at time 3 is denoted $MA_2 1$. The rationale behind the designation of these events is to examine if the market can be regarded bullish or bearish. For instance, if the closing price of a stock is larger than its opening price on the same day, one may regard this as a bullish sign for the stock.

3.2 Episode Mining

The pseudo code of the frequent episode rule mining algorithm employed in our study is shown in Fig. 3 [13]. The mechanism of the algorithm works as follows. It first generates the candidate set of episode Can_1 (line 1), and scans the complex event sequence *CES* to generate all windows (lines 3-5). The main loop of the algorithm is for the construction of episode rules (lines 7-19), which is implemented as an Apriori-like level-wise searching approach with respect to each occurrence of candidate episode c in window w, and the support of c is increased by 1 (lines 7-10). Then, the algorithm adds frequent episodes to FE_i (lines 11-12) and generates episode rules and their confidence (lines 13-15). Lastly, the algorithm outputs episode rules ERs_{WinS} and their corresponding confidence (lines 18).

3.3 Investment Strategy

In this subsection we propose an algorithm for investment in the stock market. The algorithm aims to predict whether the stock market price rises or falls at time *WinS* when we have a complex event sequence *CES* for previous *WinS* – 1 consecutive time points. The algorithm, as shown in Fig. 4, can be explained as follows. Events in *ST* will be fully expanded, and then the algorithm first determines whether the prefix of the rules occurs

in *CES*. In lines 1-6, if the episode rule occurs, the algorithm provides an investment scheme by using proposed model: We will buy a stock at time *WinS* − 1 as *SISTEM* predicts its price will rise, and then sell it at time *WinS*. On the other hand, we can sell a stock at time *WinS* − 1 if *SISTEM* predicts its price may fall, and then buy it back at time *WinS*. In finance, the former strategy is called longing a stock, and the latter shorting a stock.

Algorithm 1:
Input: A complex event sequence *CES*, a window size *WinS*, and a minimum support *min_sup*
Output: The episode rules and their corresponding confidence

1. $Can_1 := \{E|E \in \varepsilon\}$;
2. $i := 1, j := 1$;
3. **for all** ***SE*** in ***CES*** **do**
4. **for all** $z := j$ **to** $j + WinS - 1$ **do**
5. add *SE* to Win_j; //generate all windows
6. $j = j + 1$;
7. **while** $Can_i \neq \phi$ **do**
8. **for all** ***c*** in Can_i **do**
9. **for all** ***w*** in ***Win*** **do**
10. **for all** occurrence of *c* in *w*, add 1 to the support of *c*;
11. **if** $\frac{support(c)}{T_n - WinS + 1} \geq \min\_\sup$ **then**
12. add *c* to FE_i;
13. **if** $i \geq 2$ **then**
14. $fe.Confidence = \frac{support(fe)}{support(prefix(fe))}$;
15. add $(prefix(fe) \rightarrow suffix(fe))$ to ER_{S_i};
16. build Can_{i+1} from FE_i; //generate candidate episodes
17. $i := i + 1$;
18. **Output** ERs_{WinS};

Fig. 3. Pseudo code for generating episode rules.

Algorithm 2:
Input: The episode rules ERs_{WinS} and their corresponding confidence, a complex event sequence of (*WinS* − 1) consecutive time points *SubCES*, a window size *WinS*, a threshold δ
Output: Decision of buying or selling a stock at time *WinS* − 1

1. **for all** ***R*** in ERs_{WinS} **do**
2. **if** ***R.prefix*()** $\in$ ***SubCES*** **then**
3. **if** ***R.suffix*() = *rise*** and ***R.Confidence*** $\geq \delta$ **then**
4. **Output** Buy;
5. **if** ***R.suffix*() = *fall*** and ***R.Confidence*** $\geq \delta$ **then**
6. **Output** Sell;

Fig. 4. Pseudo code for the investment strategy.

4. EMPIRICAL RESULTS

In this study, we use the price dataset of TAIEX from January 6, 1967 to January 15, 2013.[1] In order to provide a statistical (cross) validation (abbreviated as CV) for our proposed model, we split the data into two parts: the first *n* percentage of the data is used to train episode rules and the remaining data is used to validate the models. As shown in Fig. 5, each CV consists of the training and testing data, respectively; *e.g.*, in CV1 the gray area indicates the first 5% of the data is used for training, and the model learned from this is examined by the testing data in the black area. In the second CV, the first 10% of the data was used for training, and the remaining data is used for testing, and so on. For this study, we have 19 CVs for the dataset of TAIEX and the dates of the training/ testing datasets in different CVs are shown in Table 3.

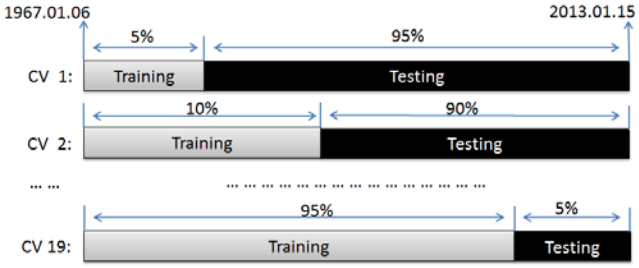

Fig. 5. Model validation: training period (gray) and testing period (black).

Table 3. Training/testing period of time in individual CVs.

CV	Training Period	Testing Period	CV	Training Period	Testing Period
1	01/06/1967 - 03/01/1969	03/03/1969 - 01/15/2013	11	01/06/1967 - 12/15/1990	12/17/1990 - 01/15/2013
2	01/06/1967 - 04/26/1971	04/27/1971 - 01/15/2013	12	01/06/1967 - 03/15/1993	03/16/1993 - 01/15/2013
3	01/06/1967 - 06/22/1973	06/23/1973 - 01/15/2013	13	01/06/1967 - 05/26/1995	05/27/1995 - 01/15/2013
4	01/06/1967 - 08/23/1975	08/25/1975 - 01/15/2013	14	01/06/1967 - 08/12/1997	08/13/1997 - 01/15/2013
5	01/06/1967 - 10/29/1977	11/01/1977 - 01/15/2013	15	01/06/1967 - 03/24/2000	03/27/2000 - 01/15/2013
6	01/06/1967 - 01/04/1980	01/05/1980 - 01/15/2013	16	01/06/1967 - 10/25/2002	10/28/2002 - 01/15/2013
7	01/06/1967 - 03/11/1982	03/12/1982 - 01/15/2013	17	01/06/1967 - 05/16/2005	05/17/2005 - 01/15/2013
8	01/06/1967 - 05/11/1984	05/12/1984 - 01/15/2013	18	01/06/1967 - 12/06/2007	12/07/2007 - 01/15/2013
9	01/06/1967 - 07/17/1986	07/18/1986 - 01/15/2013	19	01/06/1967 - 07/01/2010	07/02/2010 - 01/15/2013
10	01/06/1967 - 09/24/1988	09/29/1988 - 01/15/2013			

[1] TSEC: Taiwan Stock Exchange Corporation, http://www.twse.com.tw/en/.

Notice that this setup is different from the regular cross-validation procedure, where the data is split into two independent sets, with the process randomly repeated several times without taking into account the data's temporal order. However, in the study for stock investment here, temporal order is critical as one would like to use all available data so far to train the model and hope to apply the models in the future to gain real profits.

In this study we will use the cumulative total (compounded) return (abbreviated as *CTR*) to evaluate the performance of the model, where *CTR* is defined as the product of the model return R_t's over n consecutive time periods as follows:

$$CTR = \prod_{t=1}^{n} R_t.$$

Here we employ the moving average to illustrate how to evaluate the performance of a trading strategy by *CTR*. Using the moving averages, we expect the stock price shall rise at certain point in the future if the slope of the *MA* changes from the negative sign to the positive one because this can then be regarded as bullish for trading.

The rule above may be used for the design of an investment strategy. For example, a simple short-term scheme is to buy the stock if the condition above is satisfied, and then sell the stock one time period right after that. To illustrate the performance of this MA model, price data of TAIEX during the training period of CV 11 is used for training to search for the optimal value (corresponding to the maximum *CTR*) of k as shown in Definition 8. Fig. 6 provides the results for the *CTR* computed by various values of k where, as can be seen, the maximum *CTR* is achieved when x equals to 37. Then, this optimal parameter value will be used to examine the performance of our model in the testing data.

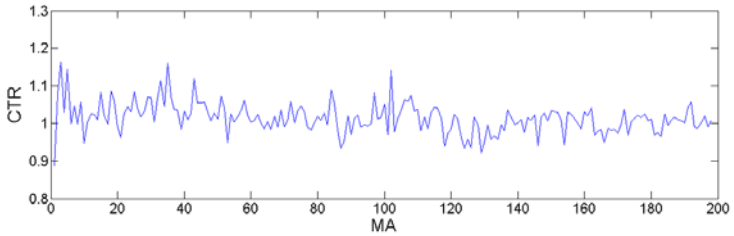

Fig. 6. CTR of the MA model for CV 11.

As for our proposed method, the parameters used in this study are displayed in Table 4. In order to test the effect of short-term (*e.g.*, one or two days for the period of moving average or window size of episode mining) and mid-term price fluctuations, we use a range of *Period* and *WinS* for the experiments. In addition, we chose the level of minimum confidence at 51% to be slightly larger than the random level (50%). As for the value of minimum support, we selected the range of 0.1% to 3% because we used several tests for various CV and found that this range would typically generate good rule sets for investment.

Here we first examine the precision of our proposed models, where the precision is defined as

$$Precision = \frac{TP}{TP + FP}.$$

In this definition, *TP* and *FP* denote the number of true positives and false negatives, respectively. In this study, a true positive occurs when the stock's return is predicted positive and it turns out to be positive; otherwise, the model generates a false positive.

Table 4. Parameters used for the proposed method.

Period	2 – 21 days
WinS	2 – 5
Minimum Support	0.1% – 3%
Minimum Confidence	51%

Table 5. Performance comparison of the benchmark and our proposed model for TOP-1 rule set in each CV.

CV	Training		Testing		
	Annualized Benchmark Return (%)	Annualized Model Return (%)	Annualized Benchmark Return (%)	Annualized Model Return (%)	Precision
1	4.91%	9.90%	10.34%	1.05%	50.00%
2	5.20%	11.69%	10.66%	7.99%	64.12%
3	22.90%	28.82%	7.85%	7.61%	57.18%
4	16.54%	27.16%	8.55%	19.89%	75.49%
5	13.31%	24.59%	8.93%	9.52%	59.57%
6	13.43%	26.04%	8.58%	19.11%	73.81%
7	10.95%	16.86%	9.66%	19.87%	64.76%
8	13.38%	19.11%	7.96%	19.17%	58.73%
9	12.11%	17.80%	8.56%	20.27%	58.98%
10	21.72%	27.09%	–0.27%	11.53%	56.48%
11	16.48%	28.40%	3.23%	8.68%	55.09%
12	15.13%	28.30%	3.06%	6.69%	54.64%
13	14.79%	27.24%	2.10%	5.60%	54.79%
14	15.58%	25.11%	–1.69%	-0.36%	55.41%
15	14.12%	24.52%	–1.85%	3.10%	60.11%
16	11.06%	23.23%	5.96%	2.82%	61.97%
17	11.23%	22.23%	4.06%	1.24%	62.04%
18	11.51%	19.95%	–1.28%	1.36%	57.38%
19	10.54%	18.47%	0.74%	2.59%	48.15%

In Table 5, we display the performance comparison between the benchmark[2] (the traditional buy-and-hold scheme) and our proposed model for each CV, and the reported results of our model in the testing phase were obtained by using the best model, named TOP-1 rule set, learned from the training phase. These results show that, in terms of the annualized return, our proposed model outperforms the benchmark in 14 out of 19 CVs; and the precision of the proposed model is greater than 50% in 18 out of 19 CVs.

[2] Beating the benchmark has been a challenging task as discussed in [21, 22]; therefore, in this paper, we compare the performance of our model with benchmark.

Furthermore, in order to illustrate the performance discrepancy between our proposed method and the benchmark, Fig. 7 displays the cumulative total returns of the two for CV 11 in the testing phase. This illustration clearly shows that our method can significantly outperform the benchmark in the long run. In addition, a further inspection shows that most of the CTR accumulation of our model appeared in the duration of August 1997 through the end of 2007. Furthermore, in the financial crisis (2008 through 2009), there is a trough in the CTR curve by either of our model and the benchmark around 2008/08/16. It appears that our model lost more than the benchmark, but also made up the loss quickly. Table 6 shows the detailed return of investment for each buy/sell pair for CV11 during this period of time. As can be seen, the total return by our model after this period of time is greater than that of the benchmark, thereby indicating that our model is more capable of making up the loss than the benchmark.

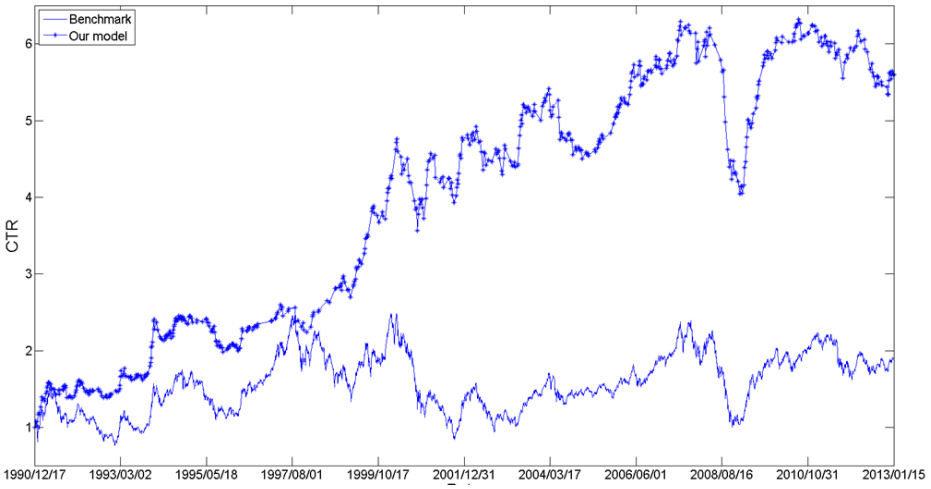

Fig. 7. Wealth accumulation for CV11 in the testing phase.

Table 6. Return of each buy/sell pair for CV11 during the 2008-2009 financial crisis.

SISTEM					
Buying date	Selling date	Return	Buying date	Selling date	Return
03/27/2008	04/21/2008	1.009707	02/12/2009	03/12/2009	1.005669
04/21/2008	08/01/2008	0.931521	03/12/2009	03/30/2009	1.118935
08/13/2008	09/04/2008	0.917555	04/03/2009	04/21/2009	1.073399
09/10/2008	10/22/2008	0.827451	04/21/2009	05/26/2009	0.993902
11/03/2008	12/01/2008	1.018007	05/26/2009	07/07/2009	1.069712
12/01/2008	12/23/2008	0.962101	07/07/2009	07/23/2009	1.029733
01/06/2009	02/12/2009	0.960949	07/23/2009	09/14/2009	1.05346
Total return (*SISTEM*)					0.936286
Benchmark					
Buying date	Buying price		Selling date	Selling price	
03/27/2008	8605.95		09/14/2009	7256.95	
Total return (Benchmark)					0.843248

Table 7. Performance comparison of the benchmark and our TOP-*k* models.

Top-*k* models	Annualized benchmark return (%)	Avg. Annualized model return (%)	Avg. Precision (%)	Number of CVs proposed method outperforms the Benchmark
$k = 1$	5.01%	8.83%	59.41%	14
$k = 3$	5.01%	10.06%	58.08%	14
$k = 5$	5.01%	8.40%	57.35%	14
$k = 10$	5.01%	7.93%	57.35%	14

It is worth mentioning that we obtained the results reported in Table 7 for testing by using the best episode mining model learned from the training data. Here we are further interested in the performance of the top *k* models in the testing phase. Our goal is to investigate if the first several good models have similar performance and it is an indication of the robustness of our models. We hence test Top-*k* models in the testing phase for each CV and compute the average model return and precision. The results for TOP-*k* models ($k = 1$, 3, 5 and 10) are shown in Table 6. For comparison, the benchmark return and the number out of 19 CVs in which our model outperformed the benchmark are displayed, as well. It can be seen that the performance of top 1, 3, 5 and 10 models are similar, thereby indicating the models generated by our proposed method are robust.

5. CONCLUSIONS

In this paper, we present a novel approach named *SISTEM* for stock investment by an integrated methodology using episode mining and technical indicators. The time-series data of stock price and the derived moving average are used to reveal the complex, associated events embedded in the stock market and construct episode events and rules. The empirical results of cross validation show that our proposed models can significantly outperform the benchmark. We also show how our method generates robust models by testing the varied number of top models. In future work, we intend to investigate this methodology more thoroughly by using various time-series datasets, such as Dow, S&P500, *etc.* To enrich our toolset for the construction of complex events, we also intend to employ other technical indicators, including volume, stochastic oscillator (KD) and Moving Average Convergence Divergence (MACD).

REFERENCES

1. T. S. Quah and B. Srinivasan, "Improving returns on stock investment through neural network selection," *Expert Systems with Applications*, Vol. 17, 1999, pp. 295-301.
2. K. J. Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," *Expert Systems with Applications*, Vol. 19, 2000, pp. 125-132.
3. M. Caplan and Y. Becker, "Lessons learned using genetic programming in a stock

picking context," in U.-M. O'Reilly, T. Yu, R. Riolo, B. Worzel, eds., *Genetic Programming Theory and Practice II*, Springer, Chapter 6, 2004, pp. 87-102.
4. T. C. Chu, C. T. Tsao, and Y. R. Shiue, "Application of fuzzy multiple attribute decision making on company analysis for stock selection," in *Proceedings of Soft Computing on Intelligent Systems and Information Processing*, 1996, pp. 509-514.
5. C.-F. Huang, "A hybrid stock selection model using genetic algorithms and support vector regression," *Applied Soft Computing*, Vol. 12, 2012, pp. 807-818.
6. C.-F. Huang, B. R. Chang, D.-W. Cheng, and C.-H. Chang, "Feature selection and parameter optimization of a fuzzy-based stock selection model using genetic algorithms," *International Journal of Fuzzy Systems*, Vol. 14, 2012, pp. 65-75.
7. H. White, "Economic prediction using neural networks: The case of IBM daily stock returns," in *Proceedings of the 2nd Annual IEEE Conference on Neural Networks*, Vol. 2, 1998, pp. 451-458.
8. A. Ng and A. W.-C. Fu, "Mining frequent episodes for relating financial events and stock trends," in *Proceedings of the 7th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, 2003, pp. 27-39.
9. A. Dattasharma and P. K. Tripathi, "Identifying stock similarity based on episode distances," in *Proceedings of IEEE International Workshop on Data Mining and Artificial Intelligence*, 2008, pp. 28-35.
10. A. Dattasharma, P. K. Tripathi, and G. Sridhar, "Identifying stock similarity based on multi-event episodes," in *Proceedings of the 7th Australasian Data Mining Conference*, 2008, pp. 153-162.
11. W. Brock, J. Lakonishok, and B. LeBaron, "Simple technical trading rules and the stochastic properties of stock returns," *Journal of Finance*, Vol. 47, 1992, pp. 1731-1764.
12. H. Mannila, H. Toivonen, and A. I. Verkamo, "Discovery of frequent episodes in event sequences," *Data Mining and Knowledge Discovery*, Vol. 1, 1997, pp. 259-289.
13. K.-Y. Huang and C.-H. Chang, "Efficient mining of frequent episodes from complex sequences," *Information Systems*, Vol. 33, 2008, pp. 96-114.
14. N. Tatti and J. Vreeken, "The long and the short of it: Summarizing event sequences with serial episodes," in *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 462-470.
15. S. Laxman, P. S. Sastry, and K. P. Unnikrishnan, "A fast algorithm for finding frequent episodes in event streams," in *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2007, pp. 410-419.
16. R. Gwadera, M. J. Atallah, and W. Szpankowski, "Reliable detection of episodes in event sequences," *Knowledge and Information System*, Vol. 7, 2005, pp. 415-437.
17. L. Wan, J. Liao, and X. Zhu, "A frequent pattern based framework for event detection in sensor network stream data," in *Proceedings of the 3rd International Workshop on Knowledge Discovery from Sensor Data*, 2009, pp. 87-96.
18. N. Meger, C. Leschi, N. Lucas, and C. Rigotti, "Mining episode rules in STULONG dataset," in *Proceedings of the ECML/PKDD2004 Discovery Challenge*, 2004, pp. 1-12.
19. D. Patnaik, P. Butler, N. Ramakrishnan, L. Parida, B. J. Keller, and A. Hanauer, "Experiences with mining temporal event sequences from electronic medical rec-

ords," in *Proceedings of ACM SIGKDD Conference on Advances in Knowledge Discovery and Data Mining*, 2011, pp. 360-368.
20. Yahoo! Finance, http://finance.yahoo.com/.
21. B. A. Sensoy, "Performance evaluation and self-designated benchmark indexes in the mutual fund industry," *Journal of Financial Economics*, Vol. 92, 2009, pp. 25-39.
22. R. Shukla and S. Singh, "A performance evaluation of global equity mutual funds: Evidence from 1988-95," *Global Finance Journal*, Vol. 8, 1997, pp. 279-293.

Yu-Feng Lin (林鈺峰) is currently pursuing the Ph.D. degree in the Department of Computer Science and Information Engineering at National Cheng Kung University in Tainan City, Taiwan. He received the M.S. degree from the Department of Computer Science and Information Engineering at National University of Kaohsiung, in 2010. His current research interests include data mining, episode mining and financial computing.

Chien-Feng Huang (黃健峯) is currently an Assistant Professor of the Department of Computer Science and Information Engineering at National University of Kaohsiung in Kaohsiung City, Taiwan. In 2002, he earned his doctoral degree from the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, Michigan, USA. He has published more than 70 scientific papers on various journals, conference proceedings and books. His current research interests include computational finance, data mining, intelligent computing and data compression.

Vincent S. Tseng (曾新穆) holds the position of Distinguished Professor at Department of Computer Science and Information Engineering at National Cheng Kung University (NCKU), Taiwan. Before this, he was a Postdoctoral Research Fellow in Computer Science Division of University of California at Berkeley during January 1998 and July 1999. He has served as the President of Taiwanese Association for Artificial Intelligence during 2011-2012 and acted as the Director for Institute of Medical Informatics of NCKU during 2008-2011. He has also served as the chair for IEEE CIS Tainan Chapter. Dr. Tseng has a wide variety of research interests covering data mining, biomedical informatics, mobile and Web technologies, multimedia databases. He has published more than 250 research papers in referred journals and conferences and also held (or filed) more than 15 patents. Dr. Tseng is a honorary member of Phi Tau Phi Society and has served as chairs/program committee for a number of premier interna-

tional conferences like SIGKDD, ICDM, CIKM, IJCAI, PAKDD. He has served on the editorial board of a number of journals including IEEE Transactions on Knowledge and Data Engineering and ACM Transactions on Knowledge Discovery from Data.