


# A Stock Trading Recommender System Based on Temporal Association Rule Mining

SAGE Open  
 April-June 2015: 1–10  
 © The Author(s) 2015  
 DOI: 10.1177/2158244015579941  
 sgo.sagepub.com  


Binoy B. Nair<sup>1</sup>, V. P. Mohandas<sup>1</sup>, Nikhil Nayanar<sup>1</sup>,  
 E. S. R. Teja<sup>1</sup>, S. Vigneshwari<sup>1</sup>, and K. V. N. S. Teja<sup>1</sup>

## Abstract

Recommender systems capable of discovering patterns in stock price movements and generating stock recommendations based on the patterns thus discovered can significantly supplement the decision-making process of a stock trader. Such recommender systems are of great significance to a layperson who wishes to profit by stock trading even while not possessing the skill or expertise of a seasoned trader. A genetic algorithm optimized Symbolic Aggregate approXimation (SAX)–Apriori based stock trading recommender system, which can mine temporal association rules from the stock price data set to generate stock trading recommendations, is presented in this article. The proposed system is validated on 12 different data sets. The results indicate that the proposed system significantly outperforms the passive buy-and-hold strategy, offering scope for a layperson to successfully invest in capital markets.

## Keywords

stock, trading, recommender, association, mining

Trading successfully in stock markets is considered to be a challenging task. Becoming a successful stock trader requires significant stock trading experience and the ability to spot trends in stock price movements. A stock trading recommender system, which can take on the role of an “expert” trader and generate stock buy/sell recommendations, is thus, of great value to a layperson who wishes to profit by investing in stocks. There have been several studies in the past, most notably by Cowles (1933, 1944) and Fama (1965a, 1965b, 1970), which suggested that stock prices follow a random walk and that it is impossible to profit from trading in stocks. However, more recent studies (Atsalakis & Valavanis, 2009; Brabazon & O’Neill, 2006; Nair & Mohandas, 2015a; Nair & Mohandas, 2015b; Nair, Mohandas, & Sakthivel, 2010a) have demonstrated that it is indeed possible to forecast stock prices and generate returns from stocks that are much higher than the traditional passive buy-and-hold (B&H) strategy. It is also observed from Atsalakis and Valavanis (2009), and Nair and Mohandas (2015b) that soft computing based systems perform better when compared with traditional linear models. Soft computing based techniques, such as artificial neural networks (ANNs) (Nair, Patturajan, Mohandas, & Sreenivasan, 2012; Saad, Prokhorov, & Wunsch, 1998; Zahedi, 1993), combination of genetic algorithms (GAs) and ANN (Nair, Sai, et al., 2011), combination of decision trees and rough sets (Nair, Mohandas, & Sakthivel, 2010a), support vector machines

(SVMs) (Huang, Nakamori, & Wang, 2005; Nair, Mohandas, & Sakthivel, 2010b), ant colony optimization (ACO) (Nair, Mohandas, & Sakthivel, 2011), and neuro-fuzzy systems (Nair, Dharini, & Mohandas, 2010; Nair, Minuvarthini, Sujithra, & Mohandas, 2010), have been successfully used for the purpose. A more comprehensive list of such soft computing based systems can be found in Atsalakis and Valavanis (2009). In Nair and Mohandas (2015a), a survey of more than a 100 publications in the field of finance is presented. However, it is observed that association rule mining (ARM) algorithms have not been widely used for generating stock trading recommendations. There has been very little research on application of ARM techniques for financial applications. ARM was used in Srisawat (2011) to discover relationships between individual stocks. ARM was used in Ting, Fu, and Chung (2006) for identification of frequently occurring patterns in a stock time series and for finding interrelationships between stock price movements of pairs of stocks. However, the economic performance of the system was not considered. An ARM-based system was proposed in Kumar and Kalia (2011) to find similarity between stocks traded on Indian

<sup>1</sup>Amrita VishwaVidyaapeetham (University), Coimbatore, India

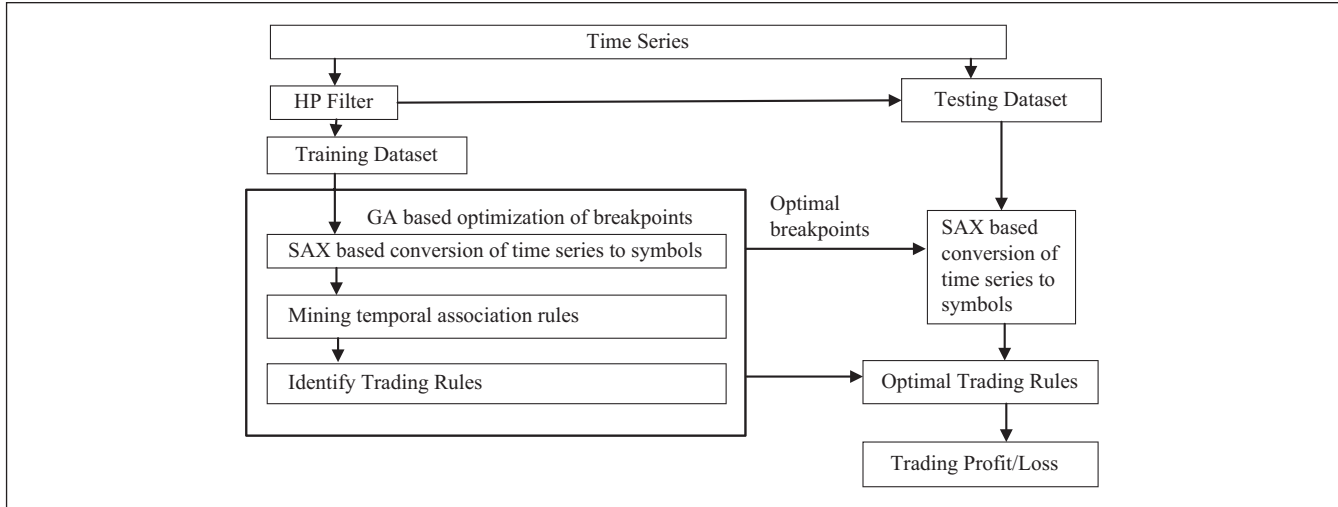
## Corresponding Author:

Binoy B. Nair, Amrita School of Engineering, Amrita VishwaVidyaapeetham (University), Coimbatore Campus, Coimbatore 641112, India.  
 Email: b\_binoy@cb.amrita.edu



Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 3.0 License

(<http://www.creativecommons.org/licenses/by/3.0/>) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access page (<http://www.uk.sagepub.com/aboutus/openaccess.htm>).



**Figure 1.** Block diagram of the proposed system.

Note. HP = Hodrick–Prescott; GA = genetic algorithm; SAX = Symbolic Aggregate approXimation.

stock markets, similar to the one proposed in Ting et al. (2006). However, no trading strategy was proposed.

The primary issue with ARM algorithms is that temporal information is not considered while mining association rules, which makes the task of mining temporal association rules from stock price time series and their incorporation into a trading recommender system all the more challenging.

The stock trading recommender system proposed in this article uses Symbolic Aggregate approXimation (SAX) to obtain a symbolic representation of the stock time series. The symbols are then used to form a transaction database, from which the association rules are obtained. Association rules that can be used for generating trading recommendations are identified from the obtained rules. Based on these rules, at the end of each trading session, recommendation for the next trading day is made. Proposed recommender system is validated on four different stocks, namely, Cipla, Hindustan Unilever (HUL), GlaxoSmithKline (GSK), and Royal Bank of Scotland (RBS), with Cipla and HUL drawn from the emerging market—India (Bombay Stock Exchange)—and GSK and RBS from the mature market—the United Kingdom (London Stock Exchange). Three different time frames are considered for each of these stocks to demonstrate the efficacy of the proposed system under different market conditions such as uptrend, downtrend, and no-trend in the stock price movements. Hence, the proposed system is validated on a total of 12 different data sets. Two variants of the proposed system are considered, and their performance is compared with that of the ARM-based stock trading decision support system proposed in Nair et al. (2013). Results are also compared with the traditional B&H strategy.

The remainder of the article is organized as follows: The following section presents the overall design of the proposed system. The implementation results are presented in the

penultimate section, and the conclusions drawn from the results are presented in the last section.

## System Design

Block diagram illustrating the functioning of the proposed system is presented in Figure 1. The detailed working of the system is as follows:

### Identification of Training and Testing Data

The first step in the design of the proposed stock trading recommender system is the selection of training and testing data sets. The selection process as proposed by Nair et al. (2013) is followed and is described as follows: Initially, the stock price data for 1 year are considered. These data are then separated into their trend and cyclic components using the Hodrick–Prescott (HP) filter (Hodrick & Prescott, 1997).

The HP filter considers the value of a time series at time  $t$ ,  $y(t)$  to be the sum of a trend component  $g(t)$  and a cyclic component  $c(t)$ , that is,

$$y(t) = g(t) + c(t), t = 0, 1, 2, \dots, N-1, \quad (1)$$

where  $N$  is the length of the time series and  $E(c(t)) \rightarrow 0$ .

The HP filter minimizes variance in  $c(t)$  while incorporating a user-defined penalty  $\lambda$  for the lack of smoothness in  $g(t)$ . Hence, the HP filter isolates  $c(t)$  from  $y(t)$  by finding the following:

$$\text{Min}_{\{g(t)\}_{t=1}^N} \left\{ \sum_{t=1}^N \left[ y(t) - g(t) \right]^2 + \lambda \sum_{t=2}^{N-1} \left[ \left[ g(t+1) - g(t) \right] - \left[ g(t) - g(t-1) \right] \right]^2 \right\}. \quad (2)$$

The value of the smoothing parameter is set in this study as 14,400, because it appears to be the most common value (e.g., Doorn, 2001; Iannaccone & Otranto, 2003; Nair et al., 2013).  $c(t)$  is further filtered using a 10-day moving average filter. The frequency content in this filtered component is then identified by computing the Discrete Fourier Transform (DFT) using Fast Fourier Transforms (an implementation of Cooley–Tukey [Cooley & Tukey, 1965] algorithm is used). From the DFT, the dominant frequency is identified from which the duration of the most prominent cycle is identified. Number of samples in the test set (out-of sample) is fixed as  $1 \times \text{Duration of the dominant cycle}$  (rounded to the nearest integer).

It was observed that training sets consisting of number of samples that are multiples of the dominant cycle duration resulted in good out-of-sample performance. A line search was used to identify the multiple for each time frame that generated the best in-sample performance.

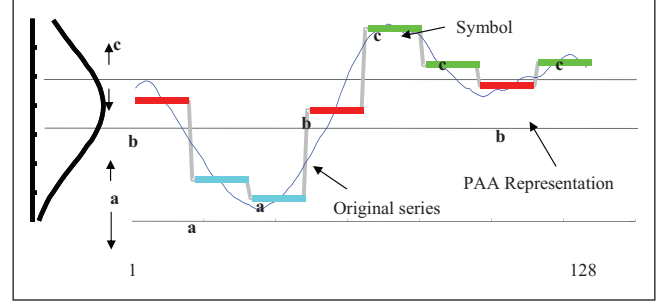
### SAX-Based Conversion of Time Series to Symbols

Once a training data set is selected, the data set is then converted into symbols with the help of SAX algorithm (Lin, Keogh, Wei, & Lonardi, 2007). The SAX algorithm works by first converting the series, say  $\mathbf{Y} = \{y(t), y(t-1), \dots, y(t-N-1)\}$ , consisting of  $N$  samples, into piecewise aggregate approximation (PAA) representation and then converting them into symbols. The algorithm requires two parameters to be specified at the outset: the window size  $D$  (the number of samples that are considered together and converted into a single symbol) and the number of symbols  $S$ . As the first step, the data are normalized. Normalization does not change the original shape of the series. After normalization, PAA is used to reduce the dimensionality of the data set. In PAA, the total number of samples are first divided into  $D$  equal sized bins (windows) with each bin consisting of  $M$  samples, such that  $M \times D = N$ . Each bin  $b_i$  is represented by a single value using Equation 3:

$$b_i = \frac{D}{N} \sum_{j=\frac{N}{D}(i-1)+1}^{\frac{N}{D}} y_j, i=1, 2, \dots, D. \quad (3)$$

The resulting series  $\mathbf{B}$  will have  $D$  number of samples; that is,  $\mathbf{B} = \{b_1, b_2, \dots, b_D\}$ .

To obtain a symbolic representation of the series, the amplitude of the series is divided into  $S$  intervals, with each interval assigned a unique symbol. It is assumed that the samples in the series  $\mathbf{Y}$ , when normalized, follow normal distribution,  $N(0, 1)$ . Hence, for  $S$  symbols, a total of  $S-1$  breakpoints are selected on the normal distribution curve such that equiprobable intervals are produced. After the breakpoints have been fixed, PAA levels are fixed and each segment is assigned with a symbol. The entire process is illustrated in Figure 2 (Lin et al., 2007).



**Figure 2.** SAX representation of a series adapted from Lin, Keogh, Wei, and Lonardi (2007).

Note. SAX = Symbolic Aggregate approXimation; PAA = piecewise aggregate approximation.

In Figure 2, the original series consisted of 128 samples, that is,  $N = 128$ . Also,  $D = 8$  and  $S = 3$ . The symbols were  $\{a, b, c\}$ . As can be seen, the normalized series is divided into eight bins with each bin being represented by its mean value (the PAA representation). Because  $S = 3$ , the series is divided into three segments with each segment represented by a symbol. Now in the final SAX representation, the initial series with  $N = 128$  gets reduced to a sequence of eight symbols: **baabcbcb**.

Because the attempt in this study is to recommend 1-day-ahead trading decision, the window size is chosen to be 1, that is,  $D = N$ . In this study, the number of symbols to be considered for each data set is determined based on the minimum and maximum value of the data (here, the data are the closing value of the stock every day). As a modification to the SAX algorithm, the breakpoints are not fixed, as proposed by Lin et al. (2007), but dynamically vary with each data set. An attempt in this direction was reported in Nair et al. (2014); however, only the number of symbols was optimized. The algorithm used for finding the bounds is presented in the algorithm SAX\_Bounds below.

#### Algorithm SAX\_Bounds

**Input:** max, min

//max: maximum closing price, min: minimum closing price in the given data set

**Output:** Bounds

Begin

1. Temp  $\leftarrow$  min

2. Bounds(1)  $\leftarrow$  min

3.  $i \leftarrow 1$

4. **While** Temp < max **do**

    Temp  $\leftarrow$  Temp + 1.02 \* Temp

$i \leftarrow i + 1$

    Bounds( $i$ )  $\leftarrow$  Temp

**End While**

**End Algorithm**

Identification of optimal breakpoints is accomplished with the help of GA, which finds the optimal breakpoint between each of the two consecutive bounds that maximize the given objective function. Hence, if there are a total of  $\beta$  bounds  $\{\text{Bounds}(1), \text{Bounds}(2), \dots, \text{Bounds}(\beta)\}$ , the GA will find  $\beta - 1$  optimal breakpoints with the upper and lower bounds for each breakpoint in the GA given by the set of tuples  $\{(\text{Bounds}(1), \text{Bounds}(2)), (\text{Bounds}(2), \text{Bounds}(3)), \dots, (\text{Bounds}(\beta - 1), \text{Bounds}(\beta))\}$ . Detailed description of the GA parameters and the objective functions used is presented below.

### GA-Based System Optimization

In the present study, GA is used to identify the optimal breakpoints in SAX. Because the objective of the present study is to obtain maximum trading profits while minimizing the losses incurred due to trading, two different systems were evaluated, the first with objective function = Profit factor, and the second with objective function = Profit per profitable trade. Here,

$$\text{Profit factor} = \frac{\text{Total profit from profit making trades}}{\text{Total loss from loss making trades}} \quad (7)$$

and

$$\text{Profit per trade} = \frac{\text{Total profit from profit making trades}}{\text{Total number of profit making trades}} \quad (8)$$

Hence, the proposed system uses GA-SAX combination to maximize the profit factor and the profit per trade, respectively.

The rest of the parameters for the GA is the same for both the cases and are listed as follows:

The initial population size is 40. Considering  $\beta$  to be the number of variables to be optimized,  $0.05 \times \min(\max(10 \times \beta, 40), 100)$  number of individuals with highest fitness are directly selected for the next generation. Eighty percent of the remaining individuals are subjected to crossover. In the crossover process, initially, a random binary vector with number of bits equivalent to number of genes in each chromosome is generated. For each bit position of the binary vector containing "1," the corresponding gene is taken from one of the parents, and for each "0," corresponding gene is taken from the other parent. Similarly, for another offspring, a new random vector is generated. The remaining individuals in the population are subjected to mutation. The Gaussian mutation technique is used. The GA stops if the value of the objective function value does not improve in 10 consecutive generations.

### Mining Temporal Association Rules

After the stock price series has been converted into symbols, association rules are identified. Apriori ARM (Agrawal &

Ramakrishnan, 1994) algorithm is used in the present study. ARM algorithms are used to detect associations between items in a transaction database. Let  $\mathbf{I} = \{I_1, I_2, \dots, I_m\}$  be the "m" unique items in the transaction database. An association rule then will be of the type  $I_A \Rightarrow I_B$  (i.e.,  $I_A$  "implies"  $I_B$ ) where  $I_A, I_B \subseteq \mathbf{I}$  and  $I_A \cap I_B = \{\}$ . In Apriori (Agrawal & Ramakrishnan, 1994) ARM algorithm, the support for this association rule is given by

$$\text{support}(I_A \Rightarrow I_B) = P(I_A \cup I_B). \quad (4)$$

The confidence is expressed as

$$\text{confidence}(I_A \Rightarrow I_B) = P(I_B | I_A). \quad (5)$$

As can be seen from the above expressions, temporal information is not considered a parameter for mining association rules from the data set. However, while attempting to mine associations from time varying data sets, such as stock price data, time information is of paramount importance. Incorporating temporal information into ARM has been attempted, for example, by Liang, Xinming, and Webliang (2005), and Winarko and Roddick (2007). The temporal ARM system proposed works by incorporating the temporal information into the item sets, as follows:

Time series represented in Equation 1 is converted into its string representation using SAX (Lin et al., 2007) algorithm, with the cut points being determined using GA as described above. The set of symbols can now be represented as  $\mathbf{S} = \{S_1, S_2, \dots, S_\beta\}$ . Representing the time series  $\mathbf{Y}$  using its symbolic representation,  $\mathbf{I} = \{I_1, I_2, \dots, I_n\}, \forall I \in \mathbf{S}$ , the symbols are then used to form the transaction database taking two symbols at a time, that is, two-item sets  $d_1 = \{I_1, I_2\}$ ,  $d_2 = \{I_2, I_3\}, \dots, d_{n-1} = \{I_{n-1}, I_n\}$ . The transaction database can then be represented as  $\mathbf{D}_{\text{temporal}} = \{d_1, d_2, \dots, d_{n-1}\}$ .

For the item set  $d_j$ ,

$$\text{support}(d_j) = \left( \frac{\sum_{i=1}^{n-1} K_i}{n-1} \right), \quad (6)$$

where  $K_i = 1$  if  $d_j \cap K_i = d_j$ ,  
0 otherwise.

Trades are recommended based on the rules obtained from the frequent two-item sets. To illustrate the functioning of the system, assume that the time series has been converted into its symbolic representation, and that there are only two symbols  $S_1$  and  $S_2$  with the price ranges represented by the two symbols being such that price range for  $S_1 <$  price range for  $S_2$ . The rules obtained can then be of three types:

1. Rule is of the type  $S_3 \Rightarrow S_3$ ; that is, there is no significant price change likely from the preceding day to the succeeding day. This is a situation where trading



**Table 1.** Selected Stocks and Time Frames.

	Time frame 1		Time frame 2		Time frame 3	
	In	Out	In	Out	In	Out
Cipla	26/9/2007-3/7/2008	4/7/2008-6/10/2008	11/2/2009-17/5/2010	18/5/2010-13/8/2010	25/4/2011-27/2/2012	28/2/2012-26/4/2012
HUL	16/1/2001-24/1/2002	25/1/2002-11/4/2002	26/5/2009-22/9/2009	23/9/2009-2/11/2009	28/3/2011-2/4/2012	3/4/2012-23/5/2012
GSK	26/12/2002-1/10/2003	2/10/2003-11/11/2003	6/2/2006-30/11/2006	1/12/2006-30/1/2007	14/10/2008-21/8/2009	24/8/2009-6/10/2009
RBS	28/9/1995-25/3/1996	26/3/1996-23/5/1996	16/8/1999-25/1/2000	26/1/2000-25/2/2000	23/7/2004-18/1/2005	19/1/2005-23/2/2005

Note. HUL = Hindustan Unilever; GSK = GlaxoSmithKline; RBS = Royal Bank of Scotland.

will not result in profit. Hence, no trade should be executed.

2. Rule is of the type  $S_2 \Rightarrow S_1$ ; that is, price level on the succeeding day is likely to be lower than price levels on the preceding day. This is a situation where trading will not result in profit. Hence, no trade should be executed.
3. Rule is of the type  $S_1 \Rightarrow S_2$ ; that is, price level on the succeeding day is likely to be higher than price level on the preceding day. This is a situation where trading will result in profit. Hence, stock should be bought at the opening of the very next day of the first one-item set and sold just before the close on the day indicated in the second one-item set.

Only the rules that correspond to the third type above are considered as trading rules and are stored along with their respective support values. Once the trading rules are identified, the system is now ready to generate trading recommendations. The recommendations are generated as follows:

**Step 1:** At the end of the trading session, the closing price of the stock is converted into its symbolic form using SAX, and the temporal information is added to it.

**Step 2:** The resulting one-item set is compared with the selected two-item sets. Only those two-item sets that have the first item matching with the one-item set generated at the day's closing are selected as candidate rules.

**Step 3:** It is checked whether the second item in the candidate two-item sets has the day-of-the-week later than the one-item set. Only the rules that satisfy this criterion are selected. This is to ensure that only those rules, using which trading is possible, are selected.

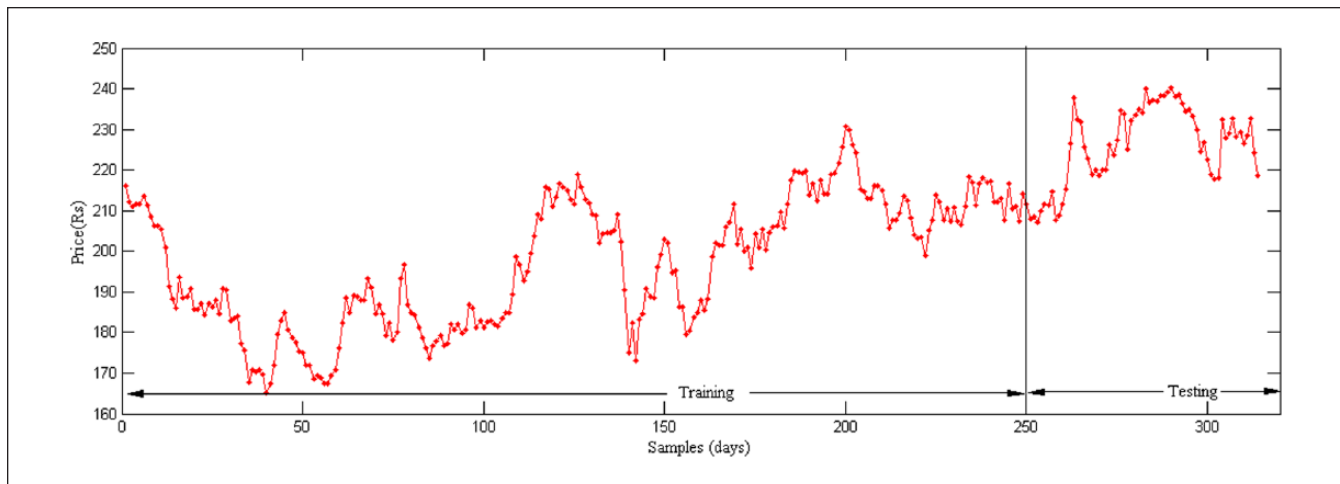
**Step 4:** In case of more than one two-item sets being selected in Step 3, the supports are compared and the two-item set with the highest support is taken as the selected trading rule. A "trade" recommendation is thus generated. In case no two-item set is selected in Step 3, a "no trade" recommendation is generated.

Once a "trade" recommendation has been generated, the user is supposed to follow the recommendation and buy the stock at the opening price on the next day. The stock should be held till the day indicated in the second item of the selected

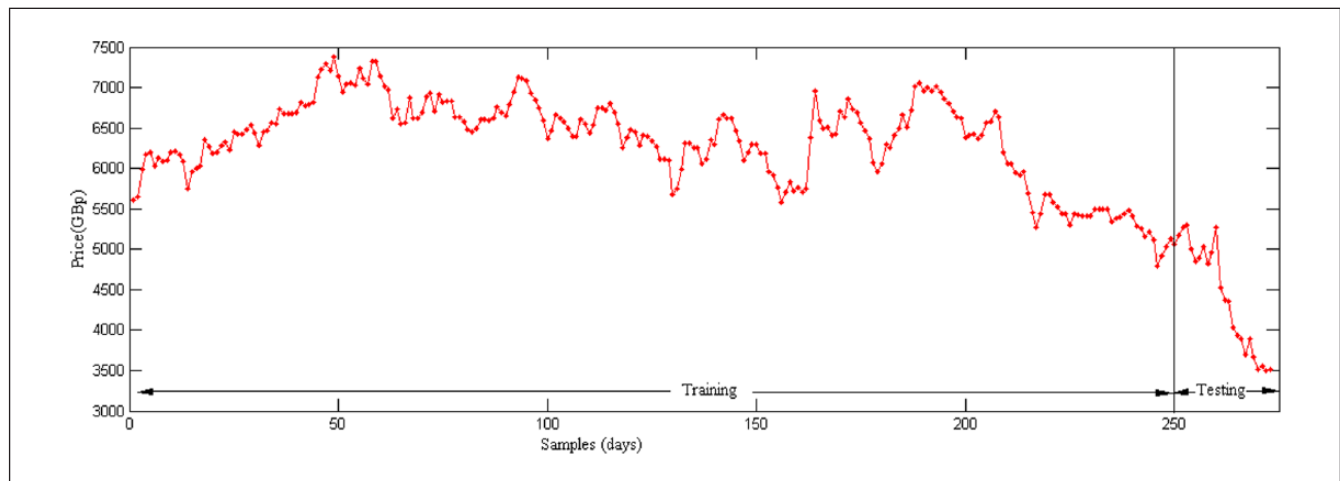
trading rule (see Step 4 above). The stock should then be sold at the closing of the indicated day.

## Results

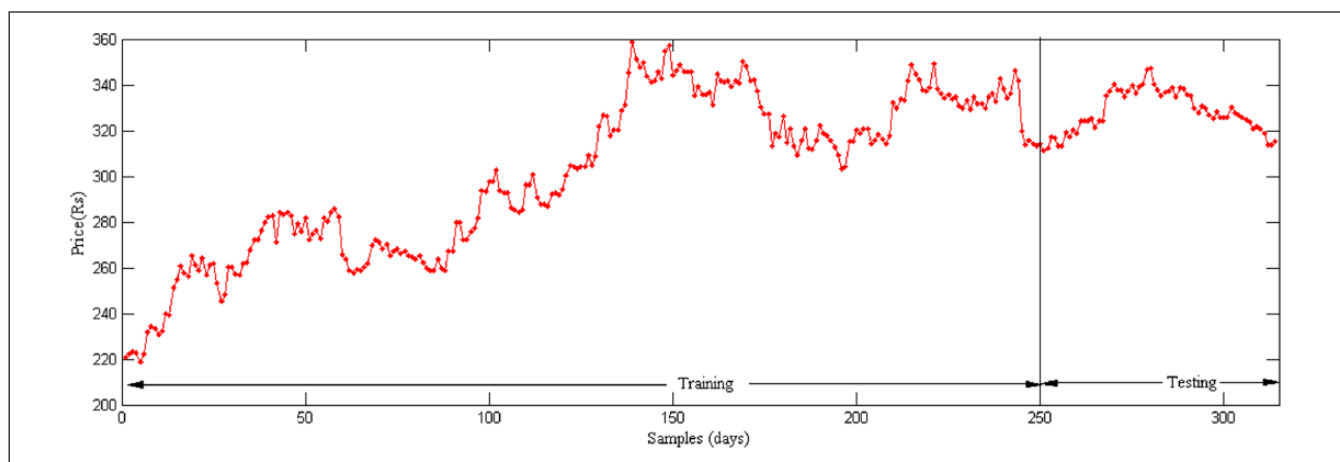
Two different trading systems were evaluated. The first system used profit factor (PF) as the objective function for the GA, whereas the second system used profit per successful trades (PPT) as the objective function. Results are compared with those generated by a similar ARM-based decision support system proposed in Nair et al. (2013), and the returns generated by the B&H strategy. Time frames selected using the technique described earlier are presented in Table 1. The data sets represent three commonly occurring trends in stock price movements, namely, uptrend, downtrend, and no-trend, as can be seen clearly from Figures 3 to 5. Two different variants of the proposed system were validated, based on the objective functions used for identifying the optimal breakpoints. The first variant used PF as the objective function, because maximizing PF implies that the GA attempts to maximize the profit from profitable trades while minimizing the loss from loss-making trades. The second variant uses PPT as the objective function, because maximizing PPT will result in identification of rules with higher profit-generation capability. A transaction cost of 0.5% per trade was considered for all the stocks over all time frames. The results are presented in Tables 2 to 5. The results presented in the tables are separated into in-sample results (In) and out-sample results (Out). It must be noted that in Table 2 (Cipla) and Table 3 (HUL), the currency unit is Indian Rupees (Rs), whereas in Table 4 (GSK) and Table 5 (RBS), the currency unit is Great Britain pence (p). The performances of both the variants of the proposed system are validated on eight different parameters as suggested in Brabazon and O'Neill (2006). In Tables 2 to 5, the performance measure—"maximum drawdown" indicates the worst profit or loss generated by the recommender system for the given data set; hence, in cases where the proposed system does not generate any loss-making trades, the lowest profit generated of all the trades is presented as the maximum drawdown. It must also be noted that "Inf" in Tables 2 to 5 implies a divide-by-zero condition; for example, if there are no loss-making trades over the considered time frame, the PF (which is the ratio of gross profits to gross losses) and the win ratio (which is the ratio of



**Figure 3.** Cipla training and testing closing price time series data for Time Frame 1 indicating an uptrend in prices.



**Figure 4.** Royal Bank of Scotland (RBS) training and testing closing price time series data for Time Frame 2 indicating a downtrend in prices.



**Figure 5.** Cipla training and testing closing price time series data for Time Frame 2 indicating no trend in prices.

**Table 2.** Economic Performance of Cipla Stock for Three Time Frames.

	Time frame 1				Time frame 2				Time frame 3			
	In		Out		In		Out		In		Out	
	PF	PPT	PF	PPT	PF	PPT	PF	PPT	PF	PPT	PF	PPT
B&H	40.75		10.65		122.2		4.2		-6.25		-9.85	
Loss per lossy trade	-1.79	-3.36	-3.43	0	-1.7	0	0	-0.9	-2.73	-0.75	-4.05	-1.72
Maximum drawdown	-3.23	-4.38	3.43	2.24	-2.4	2.89	0	-0.9	-6.27	-0.75	6.67	-4.08
Profit factor	7	2.86	1.29	Inf	12.5	Inf	0	20.94	1.28	7.8	0.51	2.74
Average profit	3.16	3.12	0.5	5.25	4.9	3.01	0	3.57	0.36	1.69	1.19	1.13
Total profit	227.56	187.31	1.99	708.19	294	54.18	0	160.74	50.89	55.82	5.97	99.05
Profit/successful trade	5.22	9.6	4.43	5.25	7.1	3.01	0	4.69	3	5.6	3.08	2.84
Total trades	72	60	4	135	60	18	0	45	143	10	5	88
Win ratio	2.4	1	1	Inf	3	Inf	0	1.3	1.17	2	0.67	1.67

Note. PF = profit factor; PPT = profit per successful trade; B&H = buy-and-hold.

**Table 3.** Economic Performance of HUL Stock for Three Time Frames.

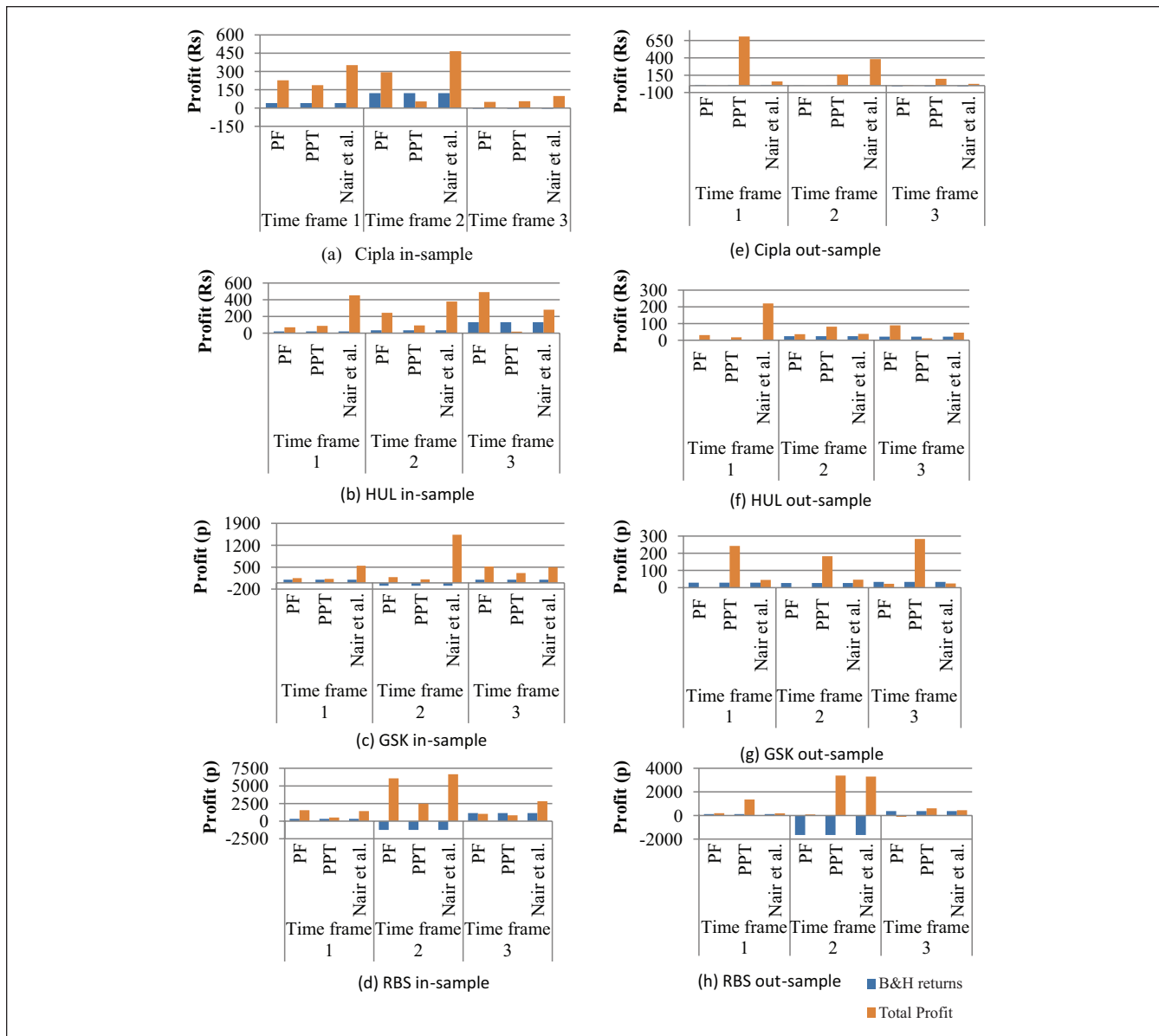
	Time frame 1				Time frame 2				Time frame 3			
	In		Out		In		Out		In		Out	
	PF	PPT	PF	PPT	PF	PPT	PF	PPT	PF	PPT	PF	PPT
B&H	21.55		2.85		34.35		24.75		130.65		21.75	
Loss per lossy trade	-2.14	-7.71	-1.89	0	-3.51	-0.45	-0.45	0	0	-3.78	-3.78	0
Maximum drawdown	-2.94	-7.71	-3.53	2.99	-5.07	-0.45	-0.45	3.98	4.58	-3.78	-3.78	0.45
Profit factor	2.38	2.90	2.03	Inf	4.47	21.67	41.84	Inf	Inf	3.55	2.47	Inf
Average profit	1.27	3.67	0.78	2.99	2.21	3.08	3.05	4.08	10.23	3.22	2.79	1.43
Total profit	70.84	88.06	31.04	17.91	243.28	92.54	36.57	81.59	491.13	18	89.15	12
Profit/successful trade	3.82	7.46	2.55	2.99	3.48	4.85	3.75	4.08	10.23	6.72	9.35	1.43
Total trades	56	24	40	6	110	29	12	20	48	18	32	12
Win ratio	1.33	3	1.5	Inf	4.5	2	5	Inf	Inf	2	1	Inf

Note. HUL = Hindustan Unilever; PF = profit factor; PPT = profit per successful trade; B&H = buy-and-hold.

**Table 4.** Economic Performance of GSK Stock for Three Time Frames.

	Time frame 1				Time frame 2				Time frame 3			
	In		Out		In		Out		In		Out	
	PF	PPT	PF	PPT	PF	PPT	PF	PPT	PF	PPT	PF	PPT
B&H	102		29		-92		27		100		33.5	
Loss per lossy trade	0	-14.26	0	0	-2.99	-6.63	0	-2.99	-16.42	0	-4.91	-4.48
Maximum drawdown	0	-28.86	0	1	-2.99	-11.94	0	-2.99	-16.42	3.98	-6	-4.48
Profit factor	Inf	1.74	0	Inf	8.67	2.4	0	8.67	3.12	Inf	2.3	5.22
Average profit	4.98	3.18	0	8.67	11.44	4.64	0	11.44	11.61	10.45	1.49	6.3
Total profit	149.25	127.36	0	242.78	183.08	111.44	0	183.08	522.38	313.43	22.39	283.58
Profit/successful trade	4.98	10.66	0	8.67	25.87	15.92	0	25.87	25.62	10.45	4.85	11.69
Total trades	30	40	0	28	16	24	0	16	45	30	15	45
Win ratio	Inf	2.33	0	Inf	1	1	0	1	2	Inf	2	2

Note. GSK = GlaxoSmithKline; PF = profit factor; PPT = profit per successful trade; B&H = buy-and-hold.



**Figure 6.** Comparison of profits generated by the two proposed systems with that generated by the B&H strategy and the system proposed in Nair et al. (2013).

Note. (a) to (d) represent the in-sample profits generated by the three systems over the three time frames for Cipla, HUL, GSK, and RBS stocks, whereas (e) to (h) present the out-of-sample profits. B&H = buy-and-hold; PF = profit factor; PPT = profit per successful trade; HUL = Hindustan Unilever; GSK = GlaxoSmithKline; RBS = Royal Bank of Scotland.

number of profitable trades to number of loss-making trades) will both be “Inf.” The overall profits generated by both the variants are compared with the profits generated by the temporal ARM-based recommender system proposed in Nair et al. (2013). The results are presented in Figure 6. It is observed from the results that the proposed system with PPT as the objective function generates higher out-of-sample profits than all the other systems considered in the present study for 9 out of the 12 data sets, whereas the proposed system with PF as the objective function generates highest out-of-sample profits of all the systems considered for 1 out of the remaining 3 data sets.

## Conclusion

In the present study, a stock trading recommender system based on mining of temporal association rules in stock prices is proposed. Performance of the system was optimized using GA. The system was validated on stocks belonging to two different markets—an emerging market (India) and a mature market (the United Kingdom). For each stock, the system was validated over three time frames. Two different variants of the proposed system are presented, based on the objective functions used for identifying the optimal temporal association rules. From the results



**Table 5.** Economic Performance of RBS Stock for Three Time Frames.

	Time frame 1				Time frame 2				Time frame 3			
	In		Out		In		Out		In		Out	
	PF	PPT	PF	PPT	PF	PPT	PF	PPT	PF	PPT	PF	PPT
B&H	335		120		-1,245		-1,655		1,144.7		375.35	
Loss per lossy trade	-23	-164.32	-26.12	-29.87	0	0	-144.19	0	-89.6	0	-52.3	0
Maximum drawdown	-47.3	-164.32	-26.12	-44.82	134.3	124.38	-144.19	169.15	-179.2	119.55	-74.72	75.24
Profit factor	3.3	1.45	3.43	4.25	Inf	Inf	1.66	Inf	1.3	Inf	0.43	Inf
Average profit	17.3	18.68	27.18	48.56	151.7	124.38	31.54	169.15	10.5	119.55	-19.92	75.24
Total profit	1,557	523.07	190.29	1,359.6	6,069.5	2,487.5	94.61	3,383	1,029.8	836.84	-119.55	601.94
Profit/successful trade	37.5	79.68	67.16	126.99	151.7	124.38	119.4	169.15	66.1	119.55	44.82	75.24
Total trades	90	28	7	28	40	20	3	20	98	7	6	8
Win ratio	2	3	1.33	1	Inf	Inf	2	Inf	1.8	Inf	0.5	Inf

Note. RBS = Royal Bank of Scotland; PF = profit factor; PPT = profit per successful trade; B&H = buy-and-hold.

obtained, it can be seen that the proposed system is capable of learning patterns from the stock price data and is able to execute profitable trades resulting in much higher profits than that generated by the traditional benchmark B&H strategy. Hence, it can be said that the proposed stock trading recommender system with PPT as the objective function can be successfully used to generate stock trading recommendations that can help a layperson trade successfully in the stock markets.

### Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

### Funding

The author(s) received no financial support for the research and/or authorship of this article.

### References

- Agrawal, R., & Ramakrishnan, S. (1994). Fast algorithms for mining association rules in large databases. In J. B. Bocca, M. Jarke, & C. Zaniolo (Eds.), *Proceedings of the 20th International Conference on Very Large Data Bases* (pp. 487-499). SantiagoChile, 12-15 September. Massachusetts, USA: Morgan Kaufmann.
- Atsalakis, G., & Valavanis, K. (2009). Surveying stock market forecasting techniques—Part II: Soft computing methods. *Expert Systems With Applications*, 36, 5932-5941.
- Brabazon, A., & O'Neill, M. (2006). *Biologically inspired algorithms for financial modelling* (1st ed.). New York, NY: Springer-Verlag.
- Cooley, J. W., & Tukey, J. W. (1965, April). An algorithm for the machine computation of the Complex Fourier Series. *Mathematics of Computation*, 19, 297-301.
- Cowles, A. (1933). Can stock market forecasters forecast? *Econometrica*, 1, 309-324.
- Cowles, A. (1944). Stock market forecasting. *Econometrica*, 12, 206-214.
- Doorn, D. (2001). Consequences of Hodrick–Prescott filtering for parameter estimation in a structural model of inventory behavior. In *Proceedings of the Annual Meeting of the American Statistical Association* (pp.5-9), 2–9 August. Atlanta, Georgia: American Statistical Association.
- Fama, E. F. (1965a). The behavior of stock-market prices. *The Journal of Business*, 38, 34-105.
- Fama, E. F. (1965b). Random walks in stock market prices. *Financial Analysts Journal*, 21(5), 55-59.
- Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *Journal of Finance*, 25, 383-417.
- Hodrick, R., & Prescott, E. (1997). Postwar U.S. business cycles: An empirical investigation. *Journal of Money, Credit and Banking*, 29(1), 1-16.
- Huang, W., Nakamori, Y., & Wang, S.-Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, 32, 2513-2522.
- Iannaccone, R., & Otranto, E. (2003). *Signal extraction in continuous time and the generalized Hodrick–Prescott filter*. St. Louis, St. Louis, Missouri, USA: WPA, Washington University.
- Kumar, M., & Kalia, A. (2011). Mining of emerging pattern: Discovering frequent itemsets in a stock data. *International Journal of Computer Technology and Applications*, 2, 3008-3014.
- Liang, Z., Xinming, T., Lin, L., & Webliang, J. (2005). *Temporal association rule mining based on T-Apriori algorithm and its typical application*. International Symposium on Spatio-Temporal Modeling, Spatial Reasoning, Analysis, Data Mining, and Data Fusion, 27-29 August. Beijing, China.
- Lin, J., Keogh, E., Wei, L., & Lonardi, S. (2007). Experiencing SAX: A novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15, 107-144.
- Nair, B. B., Dharini, N. M., & Mohandas, V. P. (2010). A stock market trend prediction system using a hybrid decision tree-neuro-fuzzy system. In *International Conference on Advances in Recent Technologies in Communication and Computing* (pp. 381-385), 15-16 October. Kottayam, India: IEEE.
- Nair, B. B., Minuvarthini, M., Sujithra, B., & Mohandas, V. P. (2010). Stock market prediction using a hybrid neuro-fuzzy System. In *International Conference on Advances in Recent*

- Technologies in Communication and Computing* (pp. 243-247). 15-16 October. Kottayam, India: IEEE.
- Nair, B. B., & Mohandas, V. P. (2015a). Artificial intelligence applications in financial forecasting—A survey and some empirical results. *Intelligent Decision Technologies*. Advance online publication. doi:10.3233/IDT-140211
- Nair, B. B., & Mohandas, V. P. (2015b). An intelligent recommender system for stock trading. *Intelligent Decision Technologies*. Advance online publication. doi:10.3233/IDT-140220
- Nair, B. B., Mohandas, V. P., & Sakthivel, N. R. (2010b). A genetic algorithm optimized decision tree-SVM based stock market trend prediction system. *International Journal on Computer Science and Engineering*, 2, 2981-2988.
- Nair, B. B., Mohandas, V. P., & Sakthivel, N. R. (2011). Predicting stock market trends using hybrid ant-colony-based data mining algorithms: An empirical validation on the Bombay Stock Exchange. *International Journal of Business Intelligence and Data Mining*, 6, 362-381.
- Nair, B. B., Mohandas, V. P., Sakthivel, N. R., Nagendran, S., Nareash, A., Nishanth, R., ... Kumar, D. M. (2010). Application of hybrid adaptive filters for stock market prediction. In S. Balamurugan, P. Balasubramanie, & G. Murugesan (Eds.), *International Conference on Communication and Computational Intelligence* (pp. 443-447). Erode, India, 27-29 December, IEEE.
- Nair, B. B., Mohandas, V. P., Varun, G., Chaitanya, I., Krishna, K. S., Karthik, S. M., & Kumar, B. J. V. (2013). A temporal association rule mining based decision support system for stock trading. *International Research Journal of Finance and Economics*, 117(5), 67-79.
- Nair, B. B., Mohandas, V. P., & Sakthivel, N. (2010a). A decision tree-rough set hybrid system for stock market trend prediction. *International Journal of Computer Applications*, 6(9), 1-6.
- Nair, B. B., Patturajan, M., Mohandas, V. P., & Sreenivasan, R. R. (2012). Predicting the BSE Sensex: Performance comparison of adaptive linear element, feed forward, and time delay neural networks. In *International Conference on Power, Signals, Controls and Computation* (pp. 1-5). Trissur, India, 3-6 January. IEEE.
- Nair, B. B., Sai, S. G., Naveen, A. N., Lakshmi, A., Venkatesh, G. S., & Mohandas, V. P. (2011). A GA-artificial neural network hybrid system for financial time series forecasting. In V. Das, G. Thomas, & F. Gaol (Eds.), *Lecture notes in computer science-communications in computer and information science* (Vol. 147, pp. 499-506). Nagpur, India: Springer-Verlag.
- Nair, B. B., Xavier, N., Mohandas, V. P., Sathyapal, A., Anusree, E. G., Kumar, P., & Ravikumar, V. (2014, November). A GA-optimized SAX-ANN based Stock Level Prediction System. *International Journal of Computer Applications*, 106(15), 7-12.
- Saad, E. W., Prokhorov, D., & Wunsch, D. (1998). Comparative study of stock trend prediction using time delay, recurrent, and probabilistic neural networks. *IEEE Transactions on Neural Networks*, 9, 1456-1470.
- Srisawat, A. (2011). *An application of association rule mining based on stock market*. Paper presented at the 3rd International Conference on Data Mining and Intelligent Information Technology Applications (ICMIA) 24-26 October 2011, (pp. 259-262), Macao, China.
- Ting, J., Fu, T.-C., & Chung, F.-L. (2006). Mining of stock data: Intra-and inter-stock pattern associative classification. *Threshold*, 5(100), 95-99.
- Winarko, E., & Roddick, J. (2007). ARMADA—An algorithm for discovering richer relative temporal association rules from interval-based data. *Data & Knowledge Engineering*, 63, 76-90.
- Zahedi, F. (1993). *Intelligent systems for business: Expert systems with neural networks*. Belmont, CA: Wadsworth.

### Author Biographies

**Binoy B Nair**, PhD, is a senior assistant professor with the Dept. of Electronics and Communication Engineering, Amrita VishwaVdyapeetham (University), India. His research interests include data mining and its application in modeling of nonlinear systems and machine condition monitoring.

**V. P. Mohandasis** is a professor with the Dept. of Electronics and Communication Engineering, Amrita VishwaVdyapeetham (University), India. His research interests include data mining and its application in modeling of nonlinear systems and Financial Engineering.

**Nikhil Nayanar** is currently pursuing his B.Tech. (Electronics and Instrumentation Engineering) in Dept. of Electronics and Communication Engineering, Amrita VishwaVdyapeetham (University), India.

**E. S. R. Tejais** currently pursuing his B.Tech. (Electronics and Instrumentation Engineering) in Dept. of Electronics and Communication Engineering, Amrita VishwaVdyapeetham (University), India.

**S. Vigneshwariis** currently pursuing his B.Tech. (Electronics and Instrumentation Engineering) in Dept. of Electronics and Communication Engineering, Amrita VishwaVdyapeetham (University), India.

**K. V. N. S. Tejais** currently pursuing his B.Tech. (Electronics and Instrumentation Engineering) in Dept. of Electronics and Communication Engineering, Amrita VishwaVdyapeetham (University), India.