Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm

Article *in* Applied Soft Computing · March 2011

DOI: 10.1016/j.asoc.2010.09.007 · Source: dx.doi.org

3 authors:

Tsung-Jung Hsieh
Industrial Technology Research Institute

Hsiao-Fen Hsiao
MingDao University

Wei-Chang Yeh
National Tsing Hua University

Applied Soft Computing 11 (2011) 2510–2525

Contents lists available at ScienceDirect

Applied Soft Computing

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

Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm

Tsung-Jung Hsieh[a,*], Hsiao-Fen Hsiao[b,**], Wei-Chang Yeh[a]

[a] Department of Industrial Engineering and Engineering Management, National Tsing Hua University, P.O. Box 24-60, Hsinchu 30013, Taiwan, ROC
[b] Department of Finance, Mingdao University, 369 Wen-Hua Rd., Peetow, Changhua 52345, Taiwan, ROC

ARTICLE INFO

Article history:
Received 6 February 2010
Received in revised form 10 June 2010
Accepted 1 September 2010
Available online 30 October 2010

Keywords:
Wavelet transform
Stepwise regression-correlation selection (SRCS)
Recurrent neural network (RNN)
Artificial bee colony algorithm (ABC)

ABSTRACT

This study presents an integrated system where wavelet transforms and recurrent neural network (RNN) based on artificial bee colony (abc) algorithm (called ABC-RNN) are combined for stock price forecasting. The system comprises three stages. First, the wavelet transform using the Haar wavelet is applied to decompose the stock price time series and thus eliminate noise. Second, the RNN, which has a simple architecture and uses numerous fundamental and technical indicators, is applied to construct the input features chosen via Stepwise Regression-Correlation Selection (SRCS). Third, the Artificial Bee Colony algorithm (ABC) is utilized to optimize the RNN weights and biases under a parameter space design. For illustration and evaluation purposes, this study refers to the simulation results of several international stock markets, including the Dow Jones Industrial Average Index (DJIA), London FTSE-100 Index (FTSE), Tokyo Nikkei-225 Index (Nikkei), and Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX). As these simulation results demonstrate, the proposed system is highly promising and can be implemented in a real-time trading system for forecasting stock prices and maximizing profits.

1. Introduction

Mining stock market tendencies is a challenging task. Numerous factors influence stock market performance, including political events, general economic conditions, and trader expectations. Though stock and futures traders rely heavily on various types of intelligent systems to make trading decisions, to date their success has been limited [1]. Even financial experts find it difficult to make accurate predictions, because stock market trends tend to be nonlinear, uncertain, and non-stationary. No consensus exists among experts as to the effectiveness of forecasting a financial time series.

One model that may be more efficient than others in stock prediction is the artificial neural network (ANN) [8]. Several studies have shown that the ANN outperforms statistical regression models [2] and discriminant analysis [3]. Generally, two different methodologies exist for stock price prediction using ANNs [4]. The first methodology considers stock price variations as a time series and predicts future prices using past data. This approach uses ANNs as predictors [5–7]. These prediction models suffer limitations owing to the enormous noise and high dimensionality of stock price data. Consequently, none of the existing prediction models has satisfactory performance, as Zadeh [8] and Marmer [9] have observed. A second approach for stock price prediction has been proposed that considers technical indices and qualitative factors, such as political effects in stock market forecasting and trend analysis. The idea here is that merging technical indicators permits the exploitation of tolerance for imprecision, uncertainty, and partial truth to achieve tractability, robustness, and low-cost solutions.

Other attempts have been made to forecast financial markets that range from traditional time series approaches to artificial intelligence techniques, including, ARCH-GARCH models [32], ANNs [8], and evolutionary computation methods [28–31]. However, the main disadvantage of both ANNs and these other black-box techniques is the enormous difficulty of interpreting the results. This study diverges from previous attempts at forecasting stock prices by proposing a method that uses the Artificial Bee Colony (ABC) algorithm combined with the ANN optimization process to create a transparent architecture. The operators of the ABC algorithm provide diverse initial weights and biases for the network under a parameter matrix design, where parameters contain the biases and weight connected neurons.

The ABC algorithm is a new meta-heuristic approach, proposed by Basturk and Karaboga [10]. Because it has the advantages of memory, multi-characters, local search, and a solution improvement mechanism, it can be used to identify a high quality optimal solution and offer a balance between complexity and performance, thus optimizing forecasting effectiveness. In this study, the param-

* Corresponding author.
** Corresponding author.
E-mail addresses: tsungjung.hsieh@gmail.com (T.-J. Hsieh), fen@mdu.edu.tw (H.-F. Hsiao).

1568-4946/$ – see front matter. Crown Copyright © 2010 Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2010.09.007

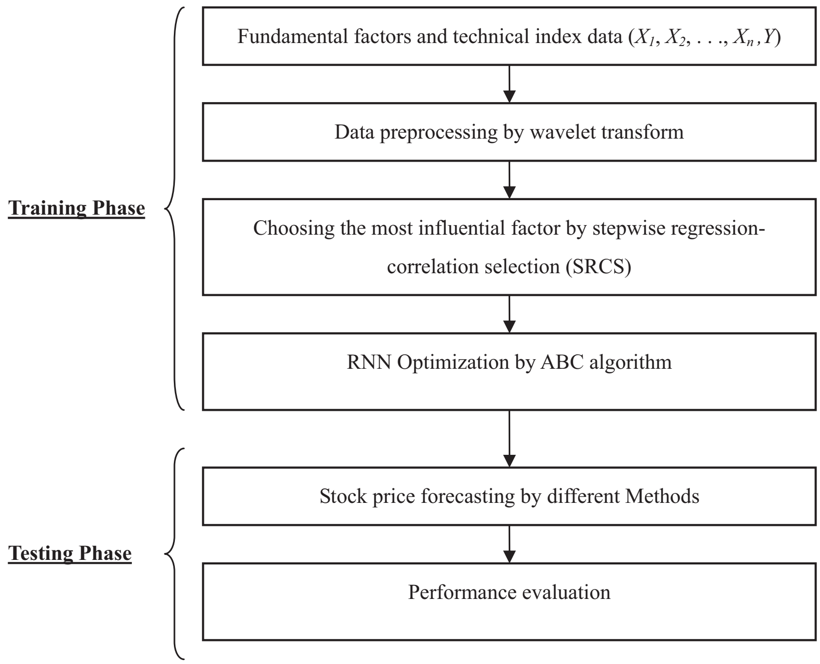

Fig. 1. Framework of ABC-RNN.

eter matrix is optimized via the powerful heuristic method, the ABC algorithm.

First, however, wavelet analysis is applied. Wavelet analysis is a relatively new field in signal processing [11]. Wavelets are mathematical functions that decompose data into different frequency components, after which each component is studied with a resolution matched to its scale, where a scale denotes a time horizon [12]. Wavelet filtering is closely related to the volatile and time-varying characteristics of the real-world time series and is not limited by the stationarity assumption [13]. The wavelet transform decomposes a process into different scales, making it useful in distinguishing seasonality, revealing structural breaks and volatility clusters, and identifying local and global dynamic properties of a process at specific timescales [14]. Wavelet analysis has been shown to be particularly useful in analyzing, modeling, and predicting the behavior of financial instruments as diverse as stocks and exchange rates [15,16]. This study applies wavelet transform using the Haar wavelet to decompose the time series.

Numerous practical problems involve a possible quite large number of input variables that can be quite large. Moreover, these input variables may contain considerable redundancy. Because of this redundancy, eliminating redundant variables may improve prediction performance. Besides, the interpretability of the predictive model can be enhanced by reducing the data dimensionality [17]. Consequently, this study employs Stepwise Regression-Correlation Selection (SRCS) to choose the input features that most strongly influence the response variable. The framework combines several statistical methods and soft computing techniques such as RNN, wavelet transform, ABC algorithm, and SRCS to extract the input feature subset. Besides applying wavelet-based data representation, SRCS to data processing before generating the RNN, and the parameter matrix was optimized via a powerful heuristic method—Artificial Bee Colony algorithm.

We tested our method on the Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX) for the period 1997–2003, and found that it was more predictable than other methods. To ensure our work was sufficiently robust and workable, we conducted simulation studies on other international stock markets, including the Dow Jones Industrial Average Index (DJIA), London FTSE-100 Index (FTSE), and Tokyo Nikkei-225 Index (Nikkei), for two periods: 1997–2003 and 2002–2008. The results indicate that the proposed integrated system is viable and useful, and it may also result in large profits.

The remainder of this paper is organized as follows: Section 2 describes the stock price prediction approach used in this study. Section 3 presents the results and compares the proposed method to previous approaches. Section 4 summarizes the findings and provides suggestions for further research.

2. Methodology

Previous studies have used statistics, technical analysis, fundamental analysis, and linear regression to predict market direction [4]. However, price forecasting is generally conducted using technical analysis or fundamental analysis. Technical analysis concentrates on market action, while fundamental analysis concentrates on the forces of supply and demand that drive price movements. The basic assumption of this study is supported by studies of the financial time series, which indicate that price movements are closely related to market returns during periods of volatility, but also to fundamental factors. The output of these factors is stock price.

To study the relations among the financial time-series variables, this work presents a hybrid method that integrates a wavelet and the ABC-RNN-based forecasting scheme. Fig. 1 shows the main procedures of this approach. The inputs and outputs of each block are further detailed in the material that follows.

2.1. Data preprocessing using wavelet transform

Wavelet theory is applied for data preprocessing, since the representation of a wavelet can deal with the non-stationarity involved in the economic and financial time series [15]. The key property of wavelets for economic analysis is decomposition by time scale. Economic and financial systems contain variables that operate on various time scales simultaneously; thus, the relations between variables may differ across time scales. One of the benefits of the wavelet approach is that it is flexible in handling highly irregular data series [13].

Table 1
All variables considered.

ID	Feature description	Computational formula or statement
1	Moving average (6) (MA (6))	$\text{MA}(m)_t = (\sum_{i=n-m}^{n} x_i)/m$, $m=6$
2	Demand index (DI)	$\text{DI}_t = (H_t + L_t + 2C_t)/4$
3	Exponential moving average (12) (EMA (12))	$\text{EMA}(m)_i = (1/m) \times \sum_{i=t-m}^{n} \text{DI}_i$, $m=12$
4.	Exponential moving average (26) (EMA (26))	$\text{EMA}(m)_i = (1/m) \times \sum_{i=t-m}^{n} \text{DI}_i$, $m=26$
5	RSI (6)	$\text{RS}(6)_i = \frac{\text{UP}(6)_{\text{avg}}}{\text{DOWN}(6)_{\text{avg}}}$, $\text{RSI}(6)_i = 100 - \left(\frac{100}{1+\text{RS}(6)_i}\right)$
		UP_{avg}: The average closed price of those arising days in a period 6.
		DOWN_{avg}: The average closed price of those dropping days in a period 6.
6	RSV (9)	$\text{RSV}(9) = 100(C(9) - L(9))/(H(9) - L(9))$
		$C(9)$: The closed price on the 9th day.
		$L(9)$: The lowest closed price in 9 days.
		$H(9)$: The highest closed price in 9 days.
7	$K(9)$	$K(9)_i = \frac{2}{3}K(9)_{t-1} + \frac{1}{3}\text{RSV}(9)_t$
8	$D(9)$	$D(9)_i = \frac{2}{3}D(9)_{t-1} + \frac{1}{3}K(9)_t$
9	MACD	$\text{MACD}(9)_i = \frac{1}{9}\sum_{t=i-9}^{i}(\text{EMA}(12)_t - \text{EMA}(26)_t)$,
10	PSY (13)	PSY (13) = [(The days of those closed prices arising in a period 13)/13] × 100
11	The close price one day ago (C)	Cx_t
12	The open price one day ago (O)	Ox_t
13	The highest price one day ago (H)	Hx_t
14	The lowest price one day ago (L)	Lx_t

This study applies the Haar wavelet as the main wavelet transform tool. A wavelet not only decomposes the data in terms of times and frequency, but also significantly reduces the processing time. Let n denote the time series size, then the wavelet decomposition used in this study can be determined in $O(n)$ time [18].

Wavelets theory is based on Fourier analysis, which represents any function as the sum of the sine and cosine functions. A wavelet $\psi(t)$ is simply a function of time t that obeys a basic rule, known as the wavelet admissibility condition [19]:

$$C_\psi = \int_0^\infty \frac{|\psi(f)|}{f} df < \infty \tag{1}$$

where $\psi(f)$is the Fourier transform and a function of frequency f, of $\psi(t)$.

Input: Candidate input features.

Output: The features that are strongly influence the dependent variable (***Y***).

BEGIN

1. Load input features ($X_1, X_2, \ldots, X_n$)
2. Determine the correlation coefficient (***r***) of each feature and ***Y***
3. Derive a correlation matrix
4. **Repeat**
5. **FOR** (***i***=1 to ***n***) DO {

   Rank each input feature according to its absolute value of ***r*** that is larger than 0.4 from the correlation matrix (suppose X_i has the largest value of $|r|$ in the current stage)

   Check the significance of the influence of this variable on the output data; i.e., derive a regression model with the form $Y = f(X_i)$.

   Apply the ***p***-value to consider the significance of each input variable}.
6. Select the statistically significant variables from Step 5 for further verification and assume that they are ($X_1, X_2, \ldots, X_k$).
7. **FOR** (***i***=1 to ***k***) DO {

   Calculate the partial ***F*** value for those statistically significant variables, as shown in equation (8).

   If the partial ***F*** value of a number (which is assumed to have a value of X_j) is below a user defined threshold, it is removed from the model because X_j is not statistically significant for the output.}
8. If the partial ***F*** value of every input number is greater than the user defined threshold, stop.

END

Fig. 2. Algorithm 1—SRCS.

The wavelet transform (WT) is a mathematical tool that can be applied to numerous applications, such as image analysis and signal processing. It was introduced to solve problems associated with the Fourier transform as they occur. This occurrence can take place when dealing with non-stationary signals, or when dealing with signals that are localized in time, space, or frequency. Depending on the normalization rules, there are two types of wavelets within a given function/family. Father wavelets describe the smooth and low-frequency parts of a signal, and mother wavelets describe the detailed and high-frequency components. In the following equations, (2a) represents the father wavelet and (2b) represents the mother wavelet, with $j = 1, \ldots, J$ in the J-level wavelet decomposition: [20]

$$\phi_{j,k} = 2^{-j/2}\phi(t - 2^j k/2^j) \tag{2a}$$

$$\psi_{j,k} = 2^{-j/2}\psi(t - 2^j k/2^j) \tag{2b}$$

where J denotes the maximum scale sustainable by the number of data points and the two types of wavelets stated above, namely father wavelets and mother wavelets, and satisfies:

$$\int \phi(t)dt = 1 \text{ and } \int \psi(t)dt = 0 \tag{3}$$

Time series data, i.e., function $f(t)$, is an input represented by wavelet analysis, and can be built up as a sequence of projections onto father and mother wavelets indexed by both $\{k\}$, $k = \{0, 1, 2, \ldots\}$ and by $\{s\} = 2^j$, $\{j = 1, 2, 3, \ldots J\}$. Analyzing real discretely sampled data requires creating a lattice for making calculations. Mathematically, it is convenient to use a dyadic expansion, as shown in equation (3). The expansion coefficients are given by the

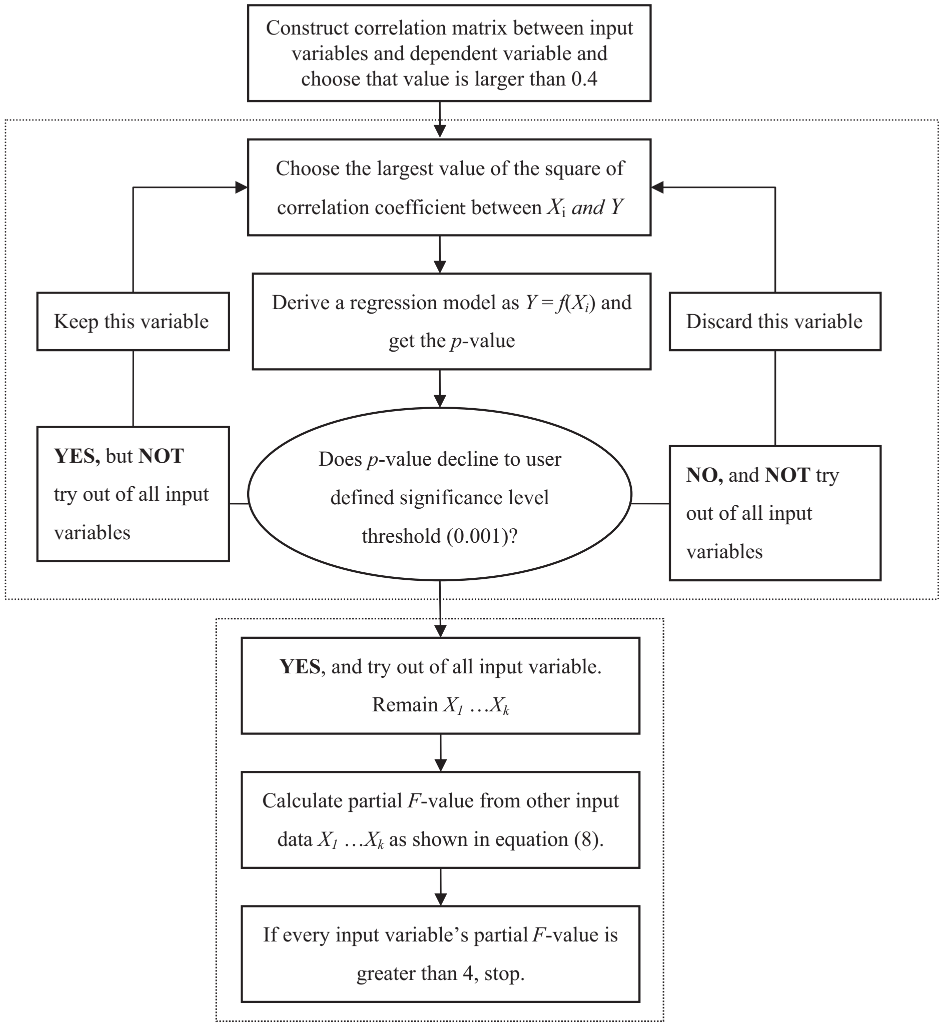

Fig. 3. Flow diagram of SRCS.

projections:

$$s_{J,k} = \int \phi_{J,k} f(t)dt$$
$$d_{j,k} = \int \psi_{j,k} f(t)dt \, (j = 1, 2, \ldots, J) \tag{4}$$

The orthogonal wavelet series approximation to $f(t)$ is defined by:

$$f(t) = \sum_k s_{J,k}\phi_{J,k}(t) + \sum_k d_{J,k}\psi_{J,k}(t) + \sum_k d_{J-1,k}\psi_{J-1,k}(t) + \cdots + \sum_k d_{1,k}\psi_{1,k}(t) \tag{5}$$

Another brief form can also be represented:

$$f(t) = S_J(t) + D_J(t) + D_{J-1}(t) + \cdots + D_1(t)$$
$$S_J(t) = \sum_k s_{J,k}\phi_{J,k}(t) \tag{6}$$
$$D_J(t) = \sum_k d_{J,k}\psi_{J,k}(t)$$

The WT is used to calculate the coefficient of the wavelet series approximation in Eq. (5) for a discrete signal $f_1, f_2, \ldots, f_n$ with finite extent. The WT maps the vector $f = (f_1, f_2, \ldots, f_n)$ to a vector of n wavelet coefficients $w = (w_1, w_2, \ldots, w_n)$, which contains both the smoothing coefficient $s_{J,k}$ and the detail coefficients $d_{j,k}$, $j = 1,2,\ldots,J$. The symbol $s_{j,k}$ describes the underlying smooth behavior of the sig-

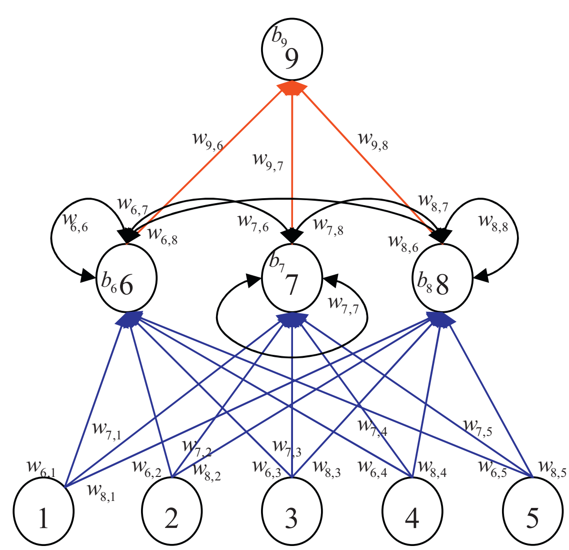

Fig. 4. RNN architecture with parameter space design.

Input: Training data set.

Output: The parameter set of the RNN.

BEGIN

1. Load training samples
2. Generate the initial population X_h randomly, h=1, 2,…, SN.
3. Evaluate the fitness (fit_h) , h=1, 2,…, SN.
4. Cycle=1
5. **Repeat**
6. **FOR** (The employed phase){
   Produce a new solution V_i by (6)
   Calculate fit_i for V_i
   Apply greedy selection process}
7. Calculate the probability P_h for X_h by (7)
8. **FOR** (The onlooker phase){
   Apply Roulette Wheel to select a solution X_h depending on P_h
   Produce a new solution V_i by (6)
   Calculate fit_i for V_i
   Apply greedy selection process}
9. **IF** (The scout phase)
   There is an abandoned solution for the scout depending on (11)
   THEN
10. Repeat it with a new one which will by randomly produced by
   (12)Memorize the best solution so far
11. Cycle=Cycle+1
12. Stop if Cycle=MCN

END

Fig. 5. Algorithm 2—ABC.

nal at coarse scale 2^J, while $d_{j,k}$ describes the coarse scale deviations from the smooth behavior, and $d_{j-1,k}$,. . .,$d_{1,k}$ provides progressively finer scale deviations from the smooth behavior [21].

When n is divisible by 2^J, $d_{1,k}$ contains $n/2$ observations at the finest scale $2^1 = 2$, and $n/4$ observations in $d_{2,k}$ at the second finest scale, $2^1 = 2$. Likewise, each of $d_{j,k}$ and $s_{j,k}$ contain $n/2^J$observations, where

$$n = n/2 + n/4 + \cdots + n/2^{J-1} + n/2^J \tag{7}$$

Let $f(t)$ denote the original data, S_1, represents an approximation signal, and D_1 is a detailed signal. This study defines the multi-resolution decomposition of a signal by specifying: S_J is the coarsest scale and $S_{J-1} = S_J + D_J$. Generally, $S_{j-1} = S_j + D_j$ where {S_J, S_{J-1},. . .,S_1} is a sequence of multi-resolution approximations of the function $f(t)$, with ever increasing levels of refinement. The corresponding multi-resolution decomposition of $f(t)$ is given by {S_J, D_J, D_{J-1},. . .D_j,. . .,D_1}.

The sequence of terms S_J, D_J, D_{J-1},. . .D_j,. . .,D_1 represents a set of orthogonal signal components that represent the signal at resolutions 1 to J. Each D_{J-k} provides the orthogonal increment to the representation of the function $f(t)$ at the scale (or resolution) 2^{J-k}.

When the data pattern is very rough, the wavelet process is repeatedly applied. The aim of preprocessing is to minimize the Root Mean Squared Error (RMSE) between the signal before and after transformation. The noise in the original data can thus be removed. Importantly, the adaptive noise in the training pattern may reduce the risk of overfitting in training phase [33]. Thus, we adopt WT twice for the preprocessing of training data in this study.

Table 2
Selected input variables for stocks.

ID	TAIEX	DJIA	Nikkei	FTSE
1	Moving average (6) (MA (6))	Moving average (6) (MA (6))	Moving average (6) (MA (6))	Moving average (6) (MA (6))
2	Demand index (DI)	Demand index (DI)	Exponential moving average (12) (EMA (12))	Demand index (DI)
3	Exponential moving average (12) (EMA (12))	Exponential moving average (12) (EMA (12))	Exponential moving average (26) (EMA (26))	Exponential moving average (12) (EMA (12))
4.	Exponential moving average (26) (EMA (26))	Exponential moving average (26) (EMA (26))	K (9)	Exponential moving average (26) (EMA (26))
5	The close price one day ago (C)	MACD	D (9)	MACD
6	The open price one day ago (O)	The close price one day ago (C)	The close price one day ago (C)	PSY (13)
7	The highest price one day ago (H)	The open price one day ago (O)	The open price one day ago (O)	The close price one day ago (C)
8	The lowest price one day ago (L)	The highest price one day ago (H)	The highest price one day ago (H)	The open price one day ago (O)
9	–	The lowest price one day ago (L)	The lowest price one day ago (L)	The highest price one day ago (H)
10	–	–	–	The lowest price one day ago (L)

Table 3
RNN architecture setting.

Architecture		TAIEX
Input layer	8	The ahead closed price, the ahead open price, the ahead highest price, the ahead lowest price, Demand Index, Moving average (6), Exponential moving average (12), Exponential moving average (26)
Hidden layer	3 (by try and error)	
Output layer	1 (close price)	
Architecture		DJIA
Input layer	9	The ahead closed price, the ahead open price, the ahead highest price, the ahead lowest price, Demand Index moving average (6), Exponential moving average (12), Exponential moving average (26), MACD
Hidden layer	3 (by try and error)	
Output layer	1 (close price)	
Architecture		Nikkei
Input layer	9	The ahead closed price, the ahead open price, the ahead highest price, the ahead lowest price, Moving Average (6), Exponential moving average (12), Exponential moving average (26), *K*, *D* values
Hidden layer	4 (by try and error)	
Output layer	1 (close price)	
Architecture		FTSE
Input layer	10	The ahead closed price, the ahead open price, the ahead highest price, the ahead lowest price and Demand Index, Moving average (6), Exponential moving average (12), Exponential moving average (26), MACD, PSY (13)
Hidden layer	4 (by try and error)	
Output layer	1 (close price)	
Transformation function	Input neurons to hidden neurons: sigmoid function	
Hidden neurons recur to hidden neurons: linear function	Hidden neurons to output neurons: linear function	

2.2. Input selection using SRCS

Table 1 lists a set of important input variables, including technical indexes and fundamental factors. This study further selects these important input variables via Stepwise Regression-Correlation Selection (SRCS).

The method of SRCS is applied to determine the set of independent variables that most strongly influences the dependent variable. This is accomplished by the repetition of a variable selection. The SRCS procedure (called algorithm 1) is detailed in Fig. 2.

$$F_j = \frac{\mathrm{MSR}(X_j \left| X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_k \right.)}{\mathrm{MSE}(X_1, \ldots, X_k)} \tag{8}$$

$$F_j^* = \underset{1 \le j \le k}{\mathrm{Max}}(F_j) \tag{9}$$

Fig. 3 shows the flow diagram of SRCS. All candidate input features are considered at the outset. The correlation coefficient (r) of each feature is determined, and also Y. We subsequently derive a correlation matrix.

There are two procedures involved in applying SRCS. Each input feature must first be ranked according to its absolute value of r that is larger than 0.4 from the correlation matrix. (Suppose X_i has the largest value of $|r|$ in the current stage.) Then, the following steps are required: (1) Check the significance of the influence of this variable on the output data. This means deriving a regression model with the form $Y = f(X_i)$; (2) Apply the p-value to consider the significance of each input variable; (3) Try out all input features.

And then, select the statistically significant variables from the first procedure for further verification (assume they are X_1, $X_2, \ldots, X_k$). Three further steps follow, which are included in the second procedure. (1) Calculate the partial F value for the statistically significant variables, as shown in Eq. (8); (2) If the partial F value of a number (which is assumed to have a value of X_j) is below the user defined threshold, remove it from the model, as it is not statistically significant for the output; (3) If the partial F value of every input number is greater than the user defined threshold, stop SRCS.

According to the outcome of SRCS, every input value should significantly influence the output value. This study sets the significance level threshold to 0.001. If the p-value of a specific variable is below the user defined threshold, that variable is added to the

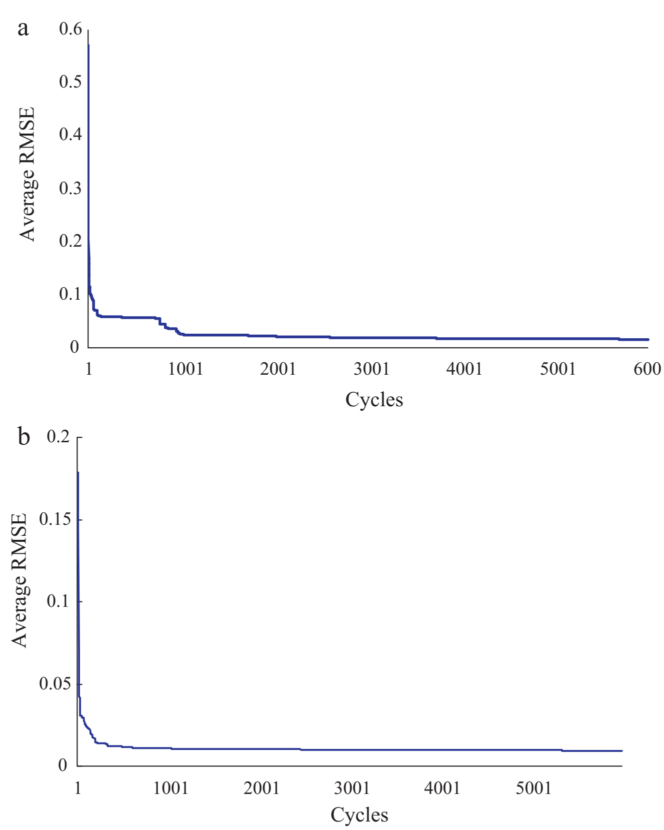

Fig. 6. Average RMSE for ABC-RNN training process: (a) 1997–2003; (b) 2002–2008.

Table 4.1
Convergence training value for ABC-RNN in period 1997–2003.

	1997	1998	1999	2000	2001	2002	2003	Average
RMSE	0.017654	0.016329	0.014195	0.015052	0.016757	0.012934	0.007022	0.014277

Table 4.2
Convergence training value for ABC-RNN in period 2002–2008.

	2002	2003	2004	2005	2006	2007	2008	Average
RMSE	0.00907	0.008361	0.010372	0.005094	0.008350	0.007173	0.014953	0.009053

model as a significant factor. If the *p*-value of a specific variable exceeds the user defined threshold, that variable is removed from the model. Following previous studies [8,22], this study always sets the partial *F* value threshold to 4. If the *F* value of a specific variable exceeds the user defined threshold, that variable is added to the model as a significant factor. If the *F* value of a specific variable is below a user defined threshold, that variable is removed from the model. The statistical software Statistical Package for Social Sciences (SPSS) version 15.0 for Windows was used to support SRCS in this study.

2.3. Artificial Bee Colony algorithm (ABC) for optimization

This work presents an enhanced ANN architecture, RNN, which is a variant of Elman's network [23]. The reason for adopting RNN is that it can form more complex computations that static feedford networks. Furthermore, some RNNs are capable of learning temporal pattern sequences that are context or time dependent, for example, time series. The presented RNN comprises an input layer, hidden layer, collection layer, and output layer, as shown in Fig. 4. Each hidden neuron is connected to itself and also to all the other hidden neurons. The ABC optimization process is further explained below. The main process involved in the RNN construction design is explained in the next subsection.

The reason for selecting the ABC algorithm as the optimized tool is that it possesses the ability to find optimal solutions with relatively modest computational requirements. In previous studies [24–26], the ABC algorithm has been used for optimizing multi-dimensional numeric problems, and the results have been compared to several famous heuristic algorithms, including the Genetic Algorithm (GA), Particle Swarm Algorithm (PSO), evolutionary algorithm (EA), and Particle Swarm Inspired Evolutionary Algorithm (PS-EA). The results show that ABC outperforms the other algorithms. Fig. 4 presents the application of ABC for RNN optimization. The pseudo-code of the ABC algorithm (called algorithm 2) is described in Fig. 5. The following statement presents the application of ABC for RNN optimization.

The ABC algorithm is a new population-based meta-heuristic approach proposed by Basturk and Karaboga [10] and further developed by Karaboga et al. [24–26]. The ABC algorithm is inspired by the intelligent foraging behavior of honeybee swarms. The foraging bees are classified into three categories—employed, onlookers and scouts. All bees currently exploiting a food source are classified as "employed". The employed bees bring loads of nectar from the food source to the hive and may share information on the food source with onlooker bees. "Onlookers" wait in the hive for employed bees to share information about their food sources, while "scouts" search for new food sources near the hive. Employed bees share information regarding food sources by dancing in a common area of the hive called the dance area. The duration of a dance is proportional to the nectar content of the identified food source. Onlooker bees, which watch numerous dances before choosing a food source, tend to choose a source that appears to be of high quality; thus good food sources attract more bees than bad ones. Whenever a bee, whether a scout or an onlooker, finds a food source, it becomes employed. Moreover, once a food source has been fully exploited, the associated employed bees abandon it and may return to being scouts or onlookers. Scout bees perform exploration, whereas employed and onlooker bees perform exploitation.

2.3.1. Solution representation

In the ABC algorithm, each food source represents a possible solution (i.e., the weight space and the corresponding biases for RNN optimization in this study) to the considered problem and the size of a food source represents the quality of the solution.

The RNN architecture presented in this study is shown in Fig. 4, where the element $w_{j,i}$ represents the weight connected from neuron i to neuron j, and the element b_k represents the bias in neuron k. When the ABC algorithm is utilized in the training process of the neural network, the parameters (linked weights and biases) between input, hidden, and output layers are represented by two parameter matrices, $s_1 = [w_{j,i}, w_{j,j}, w_{k,j}]$ and $s_2 = [b_j, b_k]$ of size $p[n+p+q]$and $(p+q)$, respectively. Element $w_{j,i}$ is the weight of the connection from input neuron i to hidden neuron j, $w_{j,j}$ is the weight that hidden j connects itself, and $w_{k,j}$ is the weight of the connection from hidden neuron j to output neuron k. Elements b_j and b_k are the biases for the hidden neuron j and output neuron k, respectively ($i = 1,2,\ldots,n$; $j = 1,2,\ldots,p$; $k = 1,2,.,q$. n, p and q are the number of input, hidden, and output neurons, respectively). Thus, the current solution for each food source will be represented by $s_t(t) = [s_1, s_2]$, where $s_t(t)$ updates with a better food source (solution).

2.3.2. Solution design

This algorithm represents a colony of artificial bees (referred to here simply as bees) comprising three types—employed, onlookers and scouts. The first half of the bee colony comprises employed bees, while the second half comprises onlookers. The ABC algorithm assumes only one employed bee per food source—namely, the number of food sources equals the number of employed bees. When a food source is abandoned, employed bees working on that food source become scouts until they find a new food source, at which point they once again become employed.

Initially, the ABC generates a randomly distributed initial population of SN solutions (food source positions), where SN denotes the number of employed or onlooker bees. Each solution X_h ($h = 1,2,\ldots,\mathrm{SN}$) is a d-dimensional vector. Here, d represents the number of optimization parameters. The population of the positions (solutions) is subject to repeated cycles, C = 1,2. . .,Maximum Cycle Number (MCN), of the search processes of the employed bees, onlooker bees and scout bees.

The ABC algorithm is an iterative algorithm, and first associates all employed bees with randomly generated food sources (solutions). Next, during iteration, every employed bee determines a food source near its currently associated food source and evaluates its nectar amount (fitness). If the nectar amount of a food source is larger than the food source it is currently working with the bee abandons the current food source in favor of the new ones; otherwise it remains at the original food sources. Once all employed

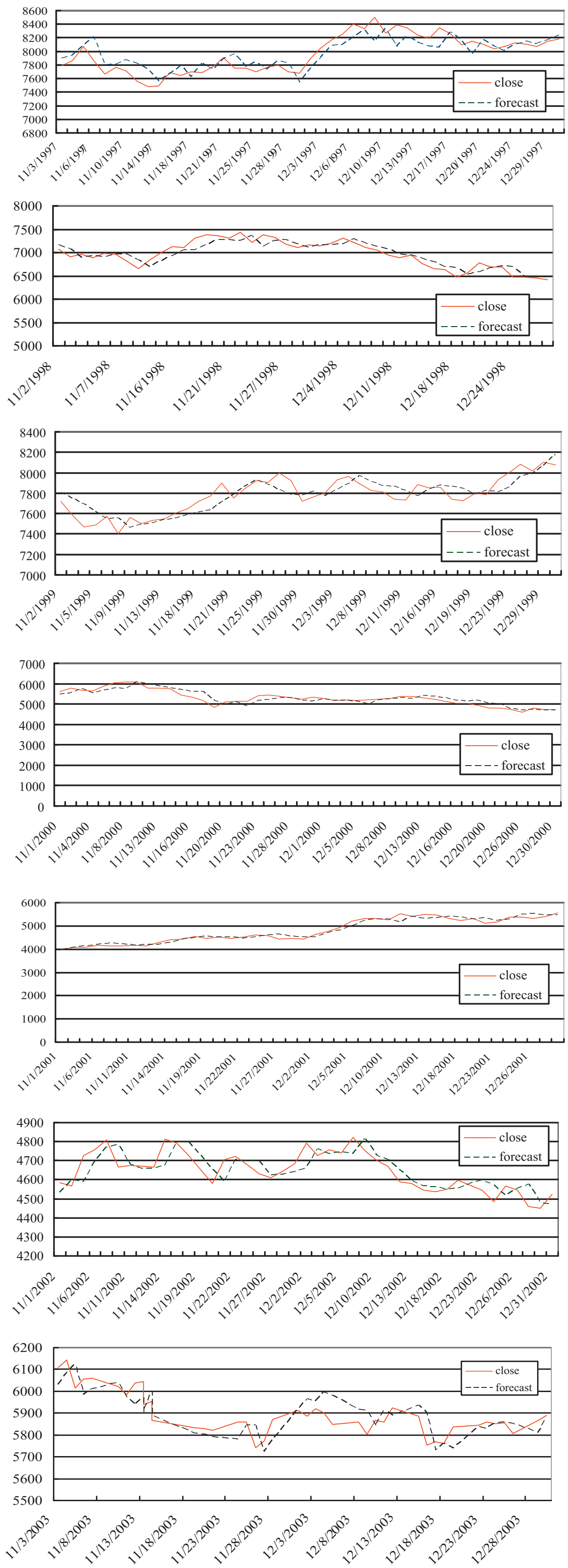

Fig. 7. Testing results of ABC-RNN for TAIEX for period 1997–2003.

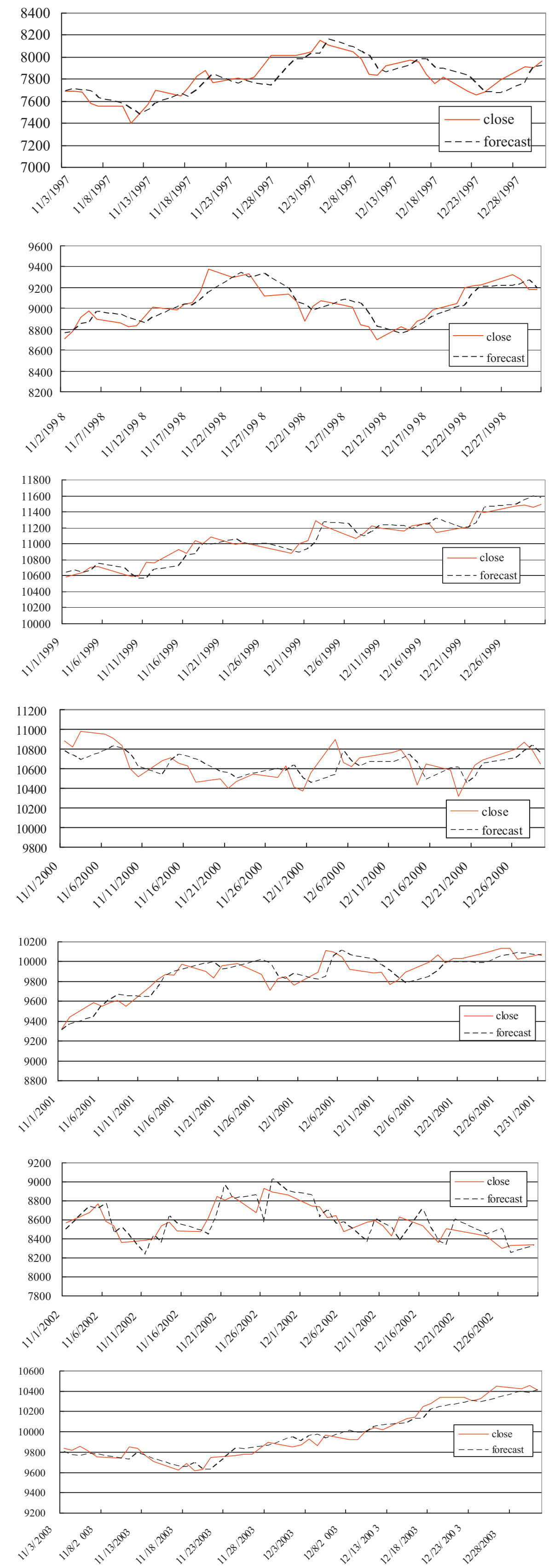

Fig. 8. Testing results of ABC-RNN for DJIA for period 1997–2003.

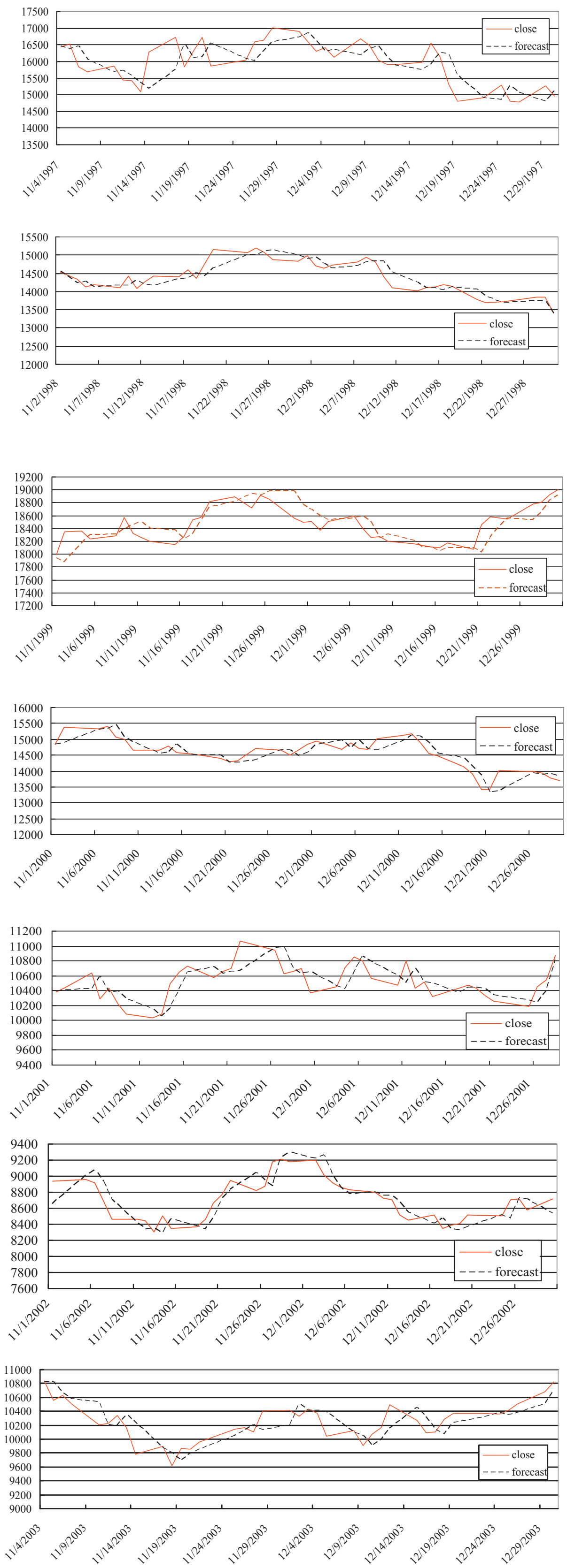

Fig. 9. Testing results of ABC-RNN for Nikkei for period 1997–2003.

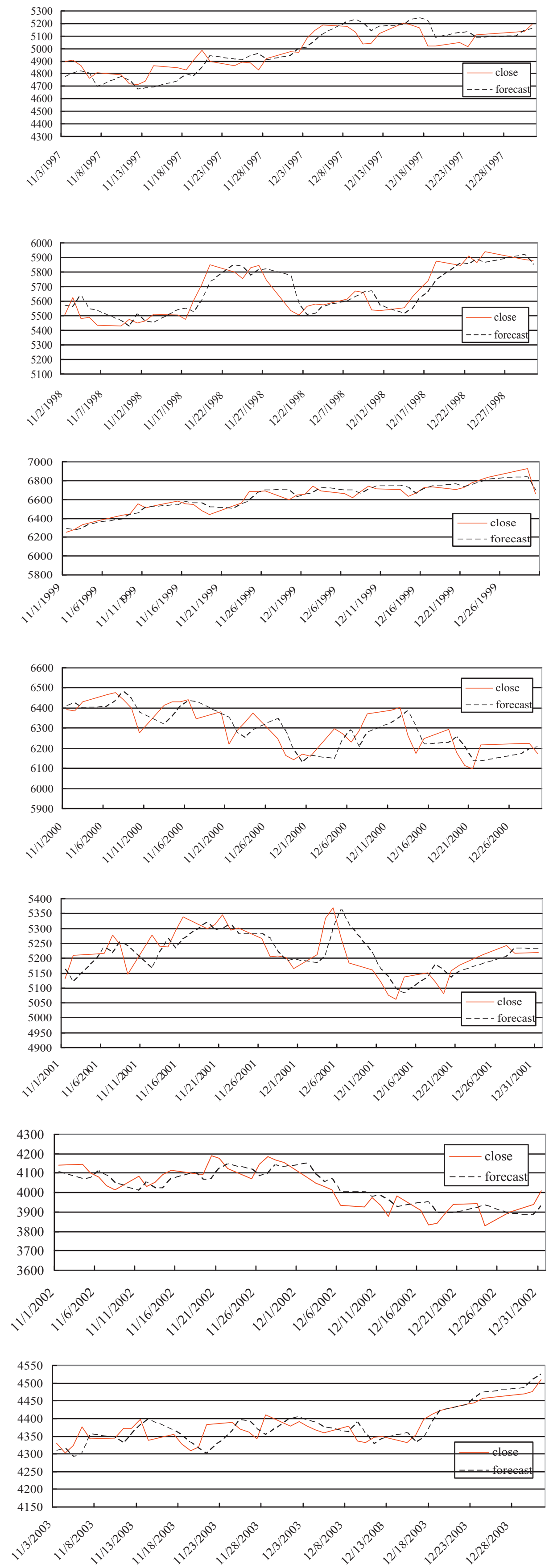

Fig. 10. Testing results of ABC-RNN for FTSE for period 1997–2003.

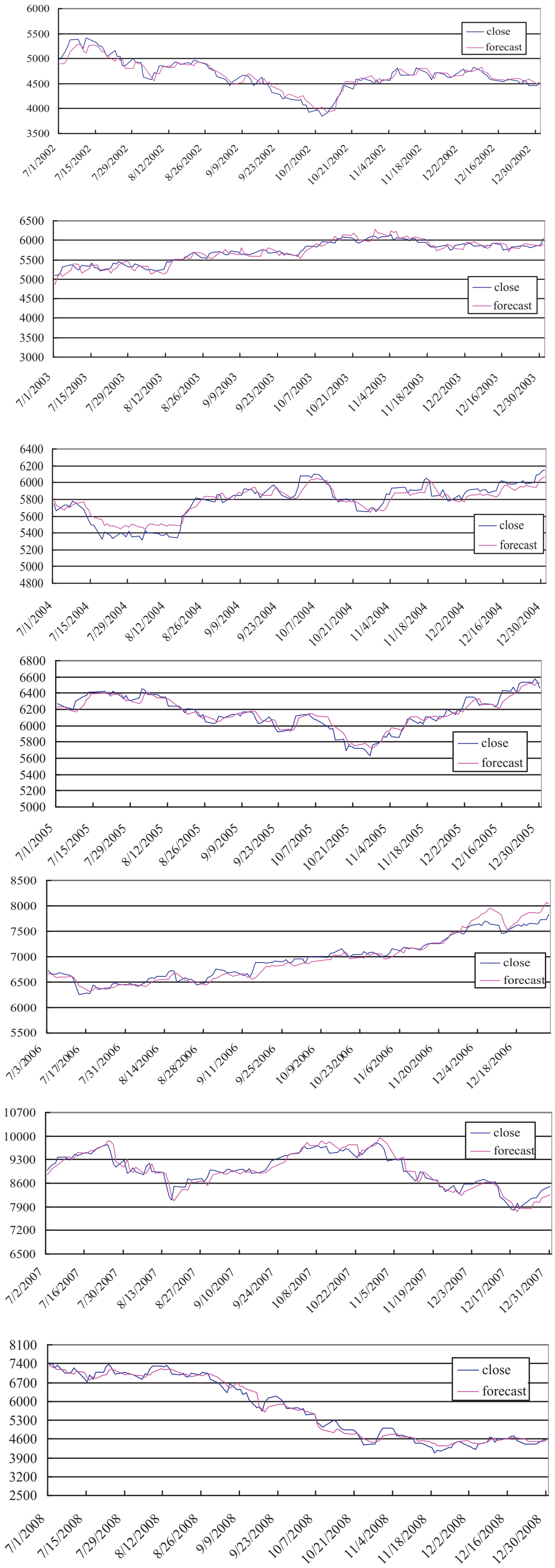

Fig. 11. Testing results of ABC-RNN for TAIEX for period 2002–2008.

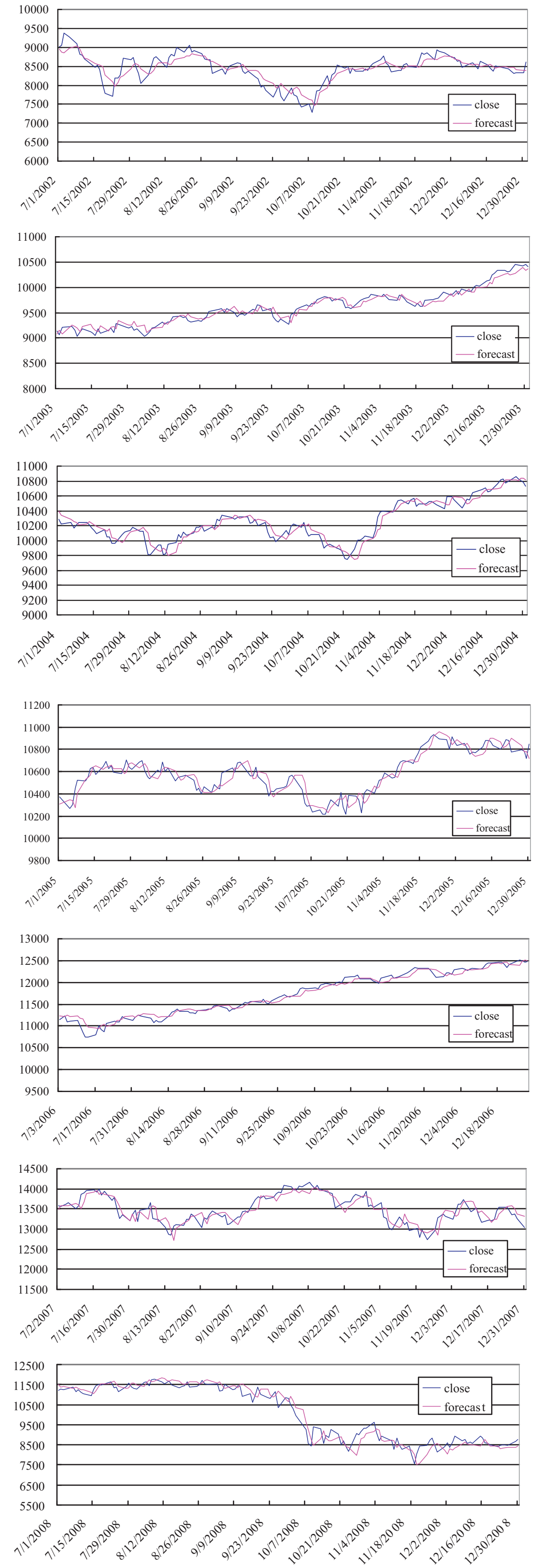

Fig. 12. Testing results of ABC-RNN for DJIA for period 2002–2008.

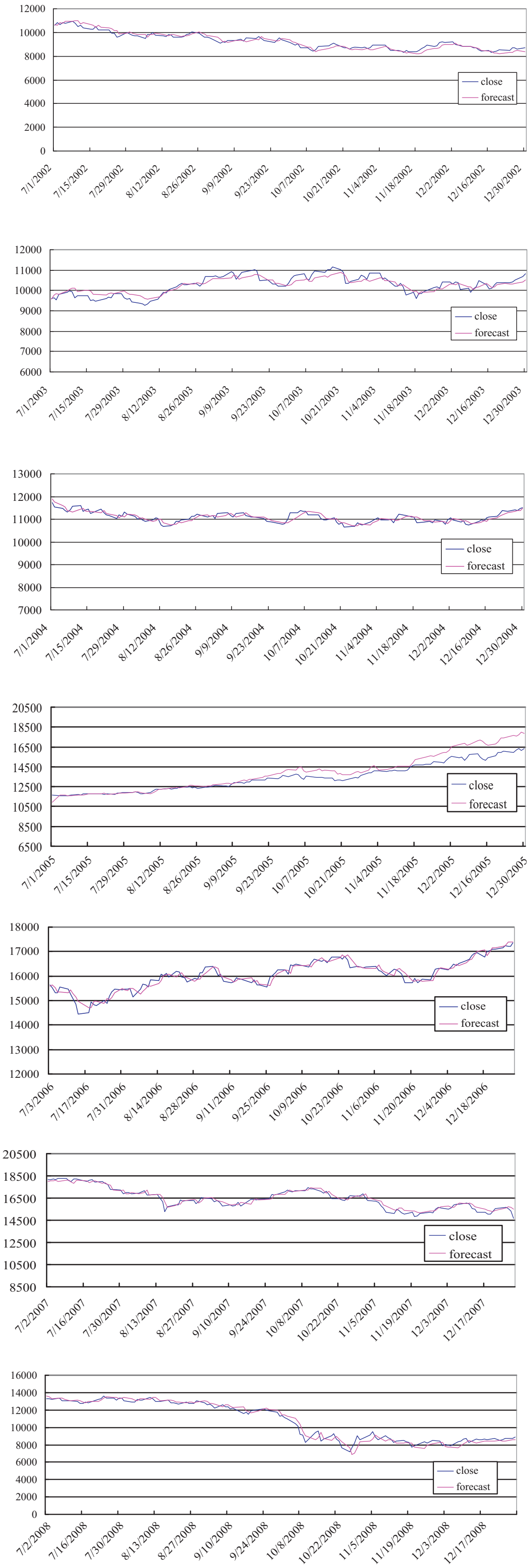

Fig. 13. Testing results of ABC-RNN for Nikkei for period 2002–2008.

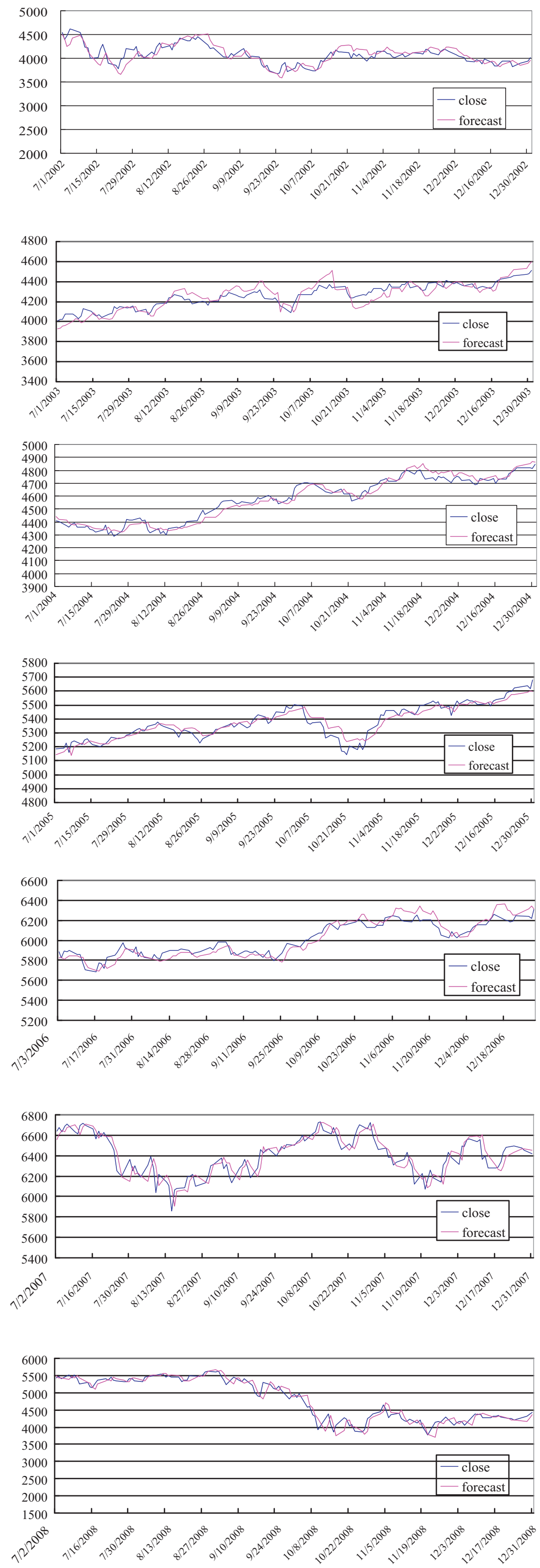

Fig. 14. Testing results of ABC-RNN for FTSE for period 2002–2008.

bees have completed this process, they share the information on the size of the food sources with the onlookers, each of whom then selects a food source, with the probability of them selecting any individual source being proportional to the amount of nectar contained in that food source; that is, the probabilities established by Eq. (10) and the probability P_h of selecting a food source (solution) X_h is determined.

$$P_h = \frac{\text{fit}_h}{\sum_{h=1}^{\text{SN}} \text{fit}_h} \tag{10}$$

where fit_h denotes the fitness of the solution represented by food source h and SN represents the total number of food sources. Clearly, this scheme leads to good food sources attracting more onlookers than bad ones. After all onlookers have selected their food sources, each onlooker determines a food source near his chosen food source and calculates its fitness. The best among the neighboring food sources as determined by the onlookers associated with a particular food source h, is selected as the next location for food source h.

2.3.3. Local search for improvement in solution

After a solution is generated, that solution is improved via a local search process called greedy selection, carried out by onlooker and employed bees; according to this process, if size (fitness) of the candidate source is better than the present source, the bee abandons the present source in favor of the candidate one. This is achieved by adding to the current value of the selected parameter the product of a uniform variable in [−1,1] and the difference in values of this parameter for this food source and some other randomly selected food source.

Formally, suppose each solution comprises d parameters and let $X_h = (X_{h1}, X_{h2},\ldots, X_{hd})$ denote a solution with parameter values X_{h1}, $X_{h2},\ldots,X_{hd}$. To determine a solution V_h near X_h, a solution parameter j and another solution $X_k = (X_{k1}, X_{k2},\ldots, X_{kd})$ are selected randomly. Except for the value of the selected parameter j, all other parameter values of V_h are the same as X_h, namely, $V_h = (X_{h1}, X_{h2},\ldots, X_{h(j-1)}, V_{hj}, X_{h(j+1)},\ldots, X_{hd})$. The value V_{hj} of the selected parameter j in V_h is determined using the following formula:

$$V_{hj} = X_{hj} + u(A_{hj} + X_{kj}) \tag{11}$$

where u denotes an uniform variable in [−1,1]. If the resulting value falls outside the acceptable range for parameter j, it is set to the corresponding extreme value in that range.

2.3.4. Solution intensity update

If a solution represented by a particular food source does not improve for a predetermined number of iterations, that food source is abandoned by its associated employed bee and the employed bee becomes a scout; namely, the bee randomly searches for a new food source. This arrangement is tantamount to assigning a randomly generated food source (solution) to the scout and again changing its status from scout to employed. After determining the new location of each food source, another iteration of the ABC algorithm begins.

The process is repeated until the termination condition is satisfied. In ABC algorithm, if a position cannot be further improved through a predetermined number of cycles, the food source is assumed to be abandoned. The value of the predetermined number of cycles is an important control parameter of the ABC algorithm, and is called the "*limit*" for abandonment and defined using Eq. (12). Assuming the abandoned source is X_h, and $j \in \{1,2,\ldots,d\}$, then the scout discovers a new food source to be replaced with X_h, and the operation is defined as Eq. (13)

$$\text{limit} = \text{SN} \times d \tag{12}$$

$$X_h^j = X_{\min}^j + \text{rand}\,[0, 1](X_{\max}^j - X_{\min}^j) \tag{13}$$

3. Illustration results

In this section of the study, we evaluate the model accuracy and compare it with other models. We also perform profit evaluations and comparisons. To evaluate the forecasting quality and performance of the ABC-RNN model, our work is applied to four stock markets. To ensure that the application is sufficiently robust, we have chosen the Dow Jones Industrial Average Index (DJIA), London FTSE-100 Index (FTSE), Tokyo Nikkei-225 Index (Nikkei), and Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX) for two periods, 1997–2003 and 2002–2008 (7 sub-datasets for each period). For the purpose of variety, we have ensured that each period possesses a different proportion of training/test data. For the 1997–2003 period, the sub-datasets for the first ten-month period are used for training (83%), while those from November to December are selected for testing; for 2002–2008, the first six-month period (50%) is used for training and the next six-month period for testing.

3.1. RNN

Before training the RNN, a wavelet transformation is applied for data preprocessing. The RMSE is thus used to perform two-level wavelet preprocessing. This process can remove the noise in the original data.

According to algorithm 1 detailed in Section 2.2, more important input variables are finally selected as inputs for predicting stock prices. Table 2 lists the selection results, where the correlation of all input variables exceeds 0.4, reaching at least a medium correlation. Furthermore, all the p-values are below 0.001.

Table 3 lists the RNN architecture setting for algorithm 2, which is implemented in *C* programming language on an Intel Pentium IV, 2.8 GHz PC with 512 MB memory. The explanation presented in Section 2.3 specifies that algorithm 2 includes three control parameters, including the number of food sources, which equals the number of employed or onlooker bees (SN), the value of *limit*, and the maximum cycle number (MCN). The values of these parameters are set as follows: number of food sources = 100, MCN = 6000. The number of limit is specified in Eq. (12). The average convergence diagram and the RMSE values of the seven training sub-datasets for each of the two periods using algorithm 2 are shown in Fig. 6 and Table 4, respectively. The MSE of the forecasting model gradually converges to around 0.0002 and 0.00008 in average for the first period and the second period, respectively. The satisfactory training results can be approved in testing data, and the forecasting figures are presented in Figs. 7–14. They are also presented numerically using criteria found in Tables 6 and 7.

Statistics about the mean and standard deviation of residual errors, i.e., R^2 and Jarque-Bera, could be helpful to check the goodness of fit and the fluctuation of the predicted results, respectively. Table 5 reveal some interesting information. The R^2 is almost 80% above for all stock markets and periods, thus, the ABC-RNN might be an adaptive model for stock market prediction, despite the marginally unsatisfactory results for Nikkei in the period 1997–2003. Jarque-Bera reveals that the residuals under the 0.05 confidence level almost confirm to normal distribution, with most fluctuations and a few large variations centering around zero.

3.2. Forecasting performance

In this section of the study, we compare the performances of the integrated system, ABC-RNN, with the conventional back propagation ANN (BP-ANN), the conventional ANN optimized by the ABC algorithm (called BNN for short) [27], and the conventional fuzzy time-series model of Chen [28] and Yu [29]. Furthermore, to examine whether the ABC-RNN surpasses the latest time series model,

Table 5.1
Residual test for model fitness and fluctuation of the predicted results in period 1997–2003.

	1997	1998	1999	2000	2001	2002	2003	Average
	DJIA							
R^2	0.818750903	0.851544324	0.873366359	0.804428666	0.808705321	0.85875372	0.940875328	0.850917803
Jarque-Bera	0.737008	0.396817	4.322374	0.116939	0.269504	2.360649	0.747284	
	NIKKEI							
R^2	0.878783264	0.795306627	0.812997726	0.88071331	0.806156058	0.796600217	0.87598882	0.83522086
Jarque-Bera	0.513177	0.055465	0.731174	1.687246	0.499714	0.75503	2.001095	
	TAIEX							
R^2	0.877964629	0.815346285	0.889835518	0.88427118	0.95170098	0.891851189	0.850667198	0.880233854
Jarque-Bera	1.477803	1.300384	1.08014	0.850524	3.08012	1.423721	2.342792	
	FTSE							
R^2	0.958694205	0.842927546	0.897444581	0.928405662	0.953063238	0.890902158	0.870148573	0.905940852
Jarque-Bera	0.429243	3.136172	0.103648	1.77944	0.098763	1.404064	4.138295	

Table 5.2
Residual test for model fitness and fluctuation of the predicted results in period 2002–2008.

	2002	2003	2004	2005	2006	2007	2008	average
	DJIA							
R^2	0.86706460	0.941876108	0.898008891	0.86742077	0.96882726	0.851805619	0.93602594	0.9044327
Jarque-Bera	0.889075	1.540462	0.694557	1.36684	36.64152	0.911256	3.752844	
	NIKKEI							
R^2	0.90034167	0.803766751	0.935529888	0.85649055	0.927779066	0.937324729	0.96423915	0.9036388
Jarque-Bera	0.614899	2.739275	1.325992	16.2705	3.228279	9.971908	10.17862	
	TAIEX							
R^2	0.91229723	0.870389456	0.867092373	0.92214703	0.925190477	0.878941135	0.97836235	0.9077742
Jarque-Bera	6.369567*	2.684154	2.424592	0.294907	25.11419	11.88336	0.537918	
	FTSE							
R^2	0.92238005	0.879822393	0.944973105	0.88027223	0.843765956	0.874952445	0.93776644	0.8977046
Jarque-Bera	13.79405	1.274335	0.358074	16.39358	5.489475*	0.875331	15.99833	

* Significant under 0.05.

Table 6
The performance comparisons of different models—TAIEX [30].

Methods	RMSE/MAE/MAPE/Theil U Year 1997	1998	1999	2000	2001	2002	2003
BP-ANN	172/152/0.0190/0.0162	186/166/0.0185/0.01729	138/111/0.0154/0.00824	187/167/0.0213/0.01771	129/105/0.0125/0.007978	133/108/0.0120/0.008006	77/69/0.0102/0.00755
BNN [27]	146/112/0.0168/0.00964	121/102/0.0127/0.00774	122/102/0.0124/0.007918	149/114/0.0164/0.01036	111/98/0.0117/0.007826	80/70/0.0100/0.007719	64/52/0.0101/0.0070028
Chen [28]	154/129/0.0171/0.0118	134/109/0.0137/0.00816	120/102/0.0121/0.007903	176/153/0.0172/0.01712	148/113/0.0133/0.010442	101/92/0.0120/0.007884	74/67/0.0105/0.007609
Yu [29]	165/137/0.0183/0.0127	164/137/0.0182/0.0125	145/112/0.0159/0.00883	145/112/0.0160/0.008791	191/172/0.0181/0.01829	75/68/0.0113/0.00746	66/54/0.0098/0.00698
Cheng et al. [30]	130/107/0.0129/0.00798	113/99/0.0120/0.00783	103/94/0.0117/0.00776	138/111/0.0156/0.00831	118/100/0.0124/0.007841	67/54/0.0104/0.007017	50*/31*/0.0057*/0.006803*
ABC-RNN	126*/105*/0.0131*/0.007922*	83*/72*/0.0115*/0.007715*	86*/78*/0.0089*/0.007148*	135*/116*/0.0164*/0.0082*	93*/87*/0.0124*/0.007772*	62*/48*/0.0095*/0.006942*	59/37/0.0071/0.006810

* The best performance among methods.

Table 7.1a
ABC-RNN for DJIA, Nikkei and FTSE in period 1997–2003—RMSE.

RMSE	1997	1998	1999	2000	2001	2002	2003
DJIA	92	90	94	137	104	132	89
Nikkei	246	203	187	230	181	141	177
FTSE	103	79	48	79	56	57	31

Table 7.1b
ABC-RNN for DJIA, Nikkei and FTSE in period 1997–2003—MAE.

MAE	1997	1998	1999	2000	2001	2002	2003
DJIA	75	70	72	116	85	109	49
Nikkei	214	157	144	174	142	117	143
FTSE	69	65	37	63	45	49	25

Table 7.1c
ABC-RNN for DJIA, Nikkei and FTSE in period 1997–2003—MAPE.

MAPE	1997	1998	1999	2000	2001	2002	2003
DJIA	0.0096	0.0078	0.0065	0.00903	0.0086	0.012	0.0051
Nikkei	0.0107	0.00912	0.00782	0.0115	0.0108	0.0099	0.0074
FTSE	0.01108	0.01013	0.0056	0.00991	0.0087	0.0061	0.0057

the performance of the Cheng et al. model [30] is compared to that of our model. The forecasting performances of TAIEX (1997–2003) are listed in Table 6.

Numerous measures of prediction accuracy are used to compare forecasting methods out of sample. This study focuses on out-of-sample performance, since the precise purpose of stock prediction is to simultaneously achieve profits and accurately predict future prices. Evaluated methods are used to measure performance such as root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and Theil's inequality coefficient (Theil U). The calculations for the above-mentioned criteria are shown in Eqs. (14)–(17).

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(X_t - F_t)^2} \tag{14}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{t=1}^{N}\left|X_t - F_t\right| \tag{15}$$

$$\mathrm{MAPE} = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{X_t - F_t}{X_t}\right| \tag{16}$$

$$\text{Theil}\,U = \frac{\sqrt{\frac{1}{N}\sum_{t=1}^{N}(X_t - F_t)^2}}{\sqrt{\frac{1}{N}\sum_{t=1}^{N}(X_t)^2} + \sqrt{\frac{1}{N}\sum_{t=1}^{N}(F_t)^2}} \tag{17}$$

Table 7.2a
ABC-RNN for TAIEX, DJIA, Nikkei and FTSE in period 2002–2008—RMSE.

RMSE	2002	2003	2004	2005	2006	2007	2008
TAIEX	95	92	81	60	116	108	91
DJIA	107	84	91	68	84	112	93
Nikkei	211	205	116	124	161	242	203
FTSE	119	71	40	42	41	88	114

Table 7.2b
ABC-RNN for TAIEX, DJIA, Nikkei and FTSE in period 2002–2008—MAE.

MAE	2002	2003	2004	2005	2006	2007	2008
TAIEX	72	78	66	45	85	92	79
DJIA	82	68	69	55	65	83	76
Nikkei	171	165	95	97	127	182	168
FTSE	85	53	30	32	37	70	91

Table 7.2c
ABC-RNN for TAIEX, DJIA, Nikkei and FTSE in period 2002–2008—MAPE.

MAPE	2002	2003	2004	2005	2006	2007	2008
TAIEX	0.0156	0.0138	0.0116	0.0075	0.012	0.0146	0.0142
DJIA	0.0083	0.0071	0.0068	0.0052	0.0056	0.0098	0.01091
Nikkei	0.0081	0.01088	0.0085	0.00772	0.00705	0.00949	0.00827
FTSE	0.00899	0.00907	0.0066	0.00614	0.00841	0.00873	0.00910

Table 8
The profits comparisons of different models (no transaction costs)—TAIEX [30].

Year	α	Methods					
		BP-ANN	BNN [27]	Chen [28]	Yu [29]	Cheng et al. [30]	ABC-RNN
1997	0.02	12	263	−127	−107	935	1491*
1998	0.03	−729	866	661	−864	878	1244*
1999	0.03	−97	109	−702	−756	124	168*
2000	0.01	−54	228	−106	−200	−154	246*
2001	0.02	204	135	−340	450	91	1753*
2002	0.005	−85	−15	−17	−96	466*	40
2003	0.005	−195	201	−238	−190	481*	260
Cumulated profits		−944	1787	−869	−1763	2821	5202*

* The best profits among methods.

where X_t denotes the true value and F_t represents the predicted value at time t. Table 6 lists the comparison results of the above criteria, and clearly shows that our model is superior to the listing methods.

Furthermore, we also challenge our approach to other stock markets in two predicted periods: 1997–2003 and 2002–2008. These stock markets include the Dow Jones Industrial Average Index (DJIA), London FTSE-100 Index (FTSE), and Tokyo Nikkei-225 Index (Nikkei). Table 7 and Tables 7.2a–7.2c show the results for the criteria RMSE, MAE and MAPE. ABC-RNN is also applicable to DIJA, Nikkei and FTSE.

3.3. *Profit evaluation*

Cheng et al. [30] identify two trading rules for calculating profits on simulated trades, where profits are calculated in units of one. Eq. (18) thus defines the profit formula, as follows:

Rule 1: Sell
If $\frac{|F_t - X_t|}{X_t} \le \alpha$ and $F_{t+1} - X_t > 0$ then sell.
Rule 2: Buy
If $\frac{|F_t - X_t|}{X_t} \le \alpha$ and $F_{t+1} - X_t < 0$ then buy.
Where α denotes the threshold parameter ($0 < \alpha \le 0.07$, and the threshold parameter depends on daily fluctuation of TAIEX). Next, profit is defined, as follows:

$$\text{Profit} = \sum_{t_s=1}^{p} (X_{t+1} - X_t) + \sum_{t_b=1}^{q} (X_t - X_{t+1}) \tag{18}$$

where p denotes the total number of days for selling, q represents the total number of days for buying, t_s represents the t-th day for selling and t_b represents the t-th day for buying.

The optimal threshold parameter α is obtained when the forecasting performance maximizes the profits in the training dataset [30]. The optimal threshold parameter α and Eq. (18) are used to calculate the profits for different models, and Table 8 lists the calculation results. Table 8 shows that the ABC-RNN achieves higher profits than the listing models (excluding 2002 and 2003) for TAIEX. More importantly, it accumulates the most profits.

For further discussion of profit evaluation, we consider the influence of transaction costs on profit. The transaction cost in the Taiwan Stock Exchange (TSE) is 0.1425% for buying and selling, which is a fixed commission fee. Selling a stock involves an additional transaction tax of 0.3%. Thus, the transaction cost is 0.1425% for buying and 0.4425% for selling, and Eq. (18) could be written as Eq. (19). Since the investor may use day trading strategies, the transaction cost will likely have a big impact on profits. Table 9 reports the performance in Table 8 when transaction costs are taken into consideration.

$$\text{Profit} = \sum_{t_s=1}^{p} [(X_{t+1} - X_t) - |X_{t+1} - X_t| \times 0.1425\%] + \sum_{t_b=1}^{q} [(X_t - X_{t+1}) - |X_t - X_{t+1}| \times 0.4425\%] \tag{19}$$

When transaction costs are taken into account, all methods have reduced profits compared to those shown in Table 8. This is due

Table 9
The profits comparisons of different models (with transaction costs)—TAIEX.

Year	α	Methods					
		BP-ANN	BNN [27]	Chen [28]	Yu [29]	Cheng et al. [30]	ABC-RNN
1997	0.02	3	208	−318	−384	791	1217*
1998	0.03	−922	695	520	−1369	577	983*
1999	0.03	−131	92	−967	−992	58	132*
2000	0.01	−93	182	−304	−367	−497	197*
2001	0.02	190	113	−492	352	61	1592*
2002	0.005	−127	−49	−46	−299	387*	15
2003	0.005	−253	183	−533	−501	341*	163
Cumulated profits		−1333	1424	−2140	−3560	1718	4299*

* The best profits among methods.

to heavy daily trading: the buy and sell strategies suffer from the heavy transaction costs. Table 9 shows that there will be more loss than the negative profits shown in Table 8. Furthermore, Table 9 shows that the ABC-RNN model also achieves higher profits than the listing models, and gets closer to [30] in 2002 and 2003. It also accumulates the most profits.

4. Conclusions and discussion

This paper proposes an integrated system, ABC-RNN, for stock price prediction, and one that combines several statistical methods and soft computing techniques. The system involves three stages: (1) data preprocessing using the Wavelet Transform (WT), which is applied to decompose the stock price time series to eliminate noise; (2) application of the Recurrent Neural Network (RNN), which has a simpler architecture than the traditional ANN and uses fundamental and technical indicators to construct the input features chosen via Stepwise Regression-Correlation Selection (SRCS); and (3) the use of the Artificial Bee Colony Algorithm (ABC) to optimize the RNN weights and biases under a parameter space design.

The proposed approach is compared to other methods presented in the literature [27–30] for stock price prediction. Simulation results show that the proposed model with wavelet-based preprocessing greatly outperforms the other methods in TAIEX. For creating further comparative benchmark problems and to show that the ABC-RNN is sufficiently robust, we have also applied our approach to other stock markets, including the Dow Jones Industrial Average Index (DJIA), London FTSE-100 Index (FTSE), and Tokyo Nikkei-225 Index (Nikkei), and we extend the predicted period from 2002 to 2008. The results for the criteria RMSE, MAE and MAPE show that the ABC-RNN model is also applicable to the DIJA, Nikkei and FTSE without respect to the training set proportions (50% or 83%) or the predicted periods (1997–2003 or 2002–2008).

Although the proposed integrated system has a satisfactory predictive performance, it still has some insufficiencies, which means the system might be enhanced. For example, it would be better to simplify the system organization, possibly by finding a scheme that includes functions of feature selection and an information supply of addictive parameters. Therefore, in the future, a different intelligent integrated method, such as a nonlinear model using SVMs as a consequence, might be applied to a more complex time series problem. In addition, a more advanced pattern selection scheme might be embedded in the system to retrieve significant patterns from the data.

Acknowledgement

The authors wish to thank the editor and three anonymous referees for their constructive comments and recommendations, which have significantly improved the presentation of this paper.

References

[1] Y.S. Abu-Mostafa, A.F. Atiya, Introduction to financial forecasting, Applied Intelligence (1996) 205–213.
[2] A.N. Refenes, A. Zapranis, G. Francis, Stock performance modeling using neural networks: a comparative study with regression models, Neural Networks (1994) 375–388.
[3] Yoo, Y. n, G. Swales, T.M. Margavio, A comparison of discriminate analysis versus artificial neural networks, Journal of the Operations Research Society (1993) 51–60.
[4] Y.Q. Zhang, S. Akkaladevi, G. Vachtsevanos, T.Y. Lin, Granular neural Web agents for stock prediction, Soft Computing Journal (2002) 406–413.
[5] P.C. Chang, Y.W. Wang, W.N. Yang, An investigation of the hybrid forecasting models for stock price variation in Taiwan, Journal of the Chinese Institute of Industrial Engineers (2004) 358–368.
[6] A.S. Chen, M.T. Leung, H. Daouk, Application of neural networks to an emerging financial market: forecasting and trading the Taiwan stock index, Computers and Operations Research (2003) 901–923.
[7] K. Parasuraman, A. Elshorbagy, Wavelet networks: an alternative to classical neural networks, IEEE International Joint Conference on Neural Networks (2005) 2674–2679.
[8] L.A. Zadeh, The role of fuzzy logic in modeling, identification and control, Modeling Identification and Control (1994) 191–203.
[9] V Marmer, Nonlinearity, nonstationarity, and spurious forecasts, Journal of Econometrics (2008) 1–27.
[10] B. Basturk, D. Karaboga, An artificial bee colony (abc) algorithm for numeric function optimization, in: IEEE Swarm Intelligence Symposium, Indianapolis, Indiana, USA, May, 2006.
[11] A. Cohen, I. Daubechies, P. Vial, Wavelets on the interval and fast wavelet transform, Applied and Computational Harmonic (1993) 54–81.
[12] J.B. Ramsey, Z. Zhang, The analysis of foreign exchange data using waveform dictionaries, Journal of Empirical Finance (1997) 341–372.
[13] A. Popoola, K. Ahmad, Testing the suitability of wavelet preprocessing for TSK fuzzy models, in: Proceeding of FUZZ-IEEE: International Conference Fuzzy System Networks, 2006, pp. 1305–1309.
[14] R. Gençay, F. Selcuk, B. Whitcher, Differentiating intraday seasonalities through wavelet multi-scaling, Physica A (2001) 543–556.
[15] J.B. Ramsey, The contribution of wavelets to the analysis of economic and financial data, Philosophical Transactions of the Royal Society of London Series A-Mathematical Physical and Engineering Sciences (1999) 2593–2606.
[16] K. Papagiannaki, N. Taft, Z.-L. Zhang, C. Diot, Long-term forecasting of internet backbone traffic, IEEE Transactions on Neural Networks (2005) 1110–1124.
[17] Y. Kim, W.N. Street, An intelligent system for customer targeting: a data mining approach, Decision Support System (2004) 215–228.
[18] F. Abramovich, P. Besbeas, T. Sapatinas, Empirical, Bayes approach to block wavelet function estimation, Computational Statistics and Data Analysis (2002) 435–451.
[19] R. Gencay, F. Selcuk, B. Whitcher, An Introduction to Wavelets and Other Filtering Methods in Finance and Economics, Academic Press, New York, 2002.
[20] J.B. Ramsey, C. Lampart, The decomposition of economic relationships by time scale using wavelets: expenditure and income, Nonlinear Dynamics and Econometrics (1998) 23–42.
[21] S. Adel, C. Martin, J.R. Heather, A.M. Jose, The reaction of stock markets to crashes and events: a comparison study between emerging and mature markets using wavelet transforms, Physica A (2006) 511–521.
[22] P.C. Chang, C.H. Liu, Y.W. Wang, A hybrid model by clustering and evolving fuzzy rules for sale forecasting in printed circuit board industry, Decision Support System (2006) 1715–1729.
[23] J.L. Elman, Finding structure in time, Cognitive Science 14 (1990) 179–211.
[24] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm, Journal of Global Optimization (2007) 459–471.
[25] D. Karaboga, B. Basturk, On the performance of artificial bee colony (abc) algorithm, Applied Soft Computing (2008) 687–697.
[26] D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm, Applied Mathematics and Computation (2009) 108–132.
[27] D. Karaboga, C. Ozturk, Neural networks training by artificial bee colony algorithm on pattern classification, Neural Network World (2009) 279–292.
[28] S.M. Chen, Forecasting enrollments based on fuzzy time-series, Fuzzy Sets Systems (1996) 11–319.
[29] H.K. Yu, Weighted fuzzy time-series models for TAIEX forecasting, Physica A (2005) 609–624.
[30] C.H. Cheng, L.Y. Wei, Y.S. Chen, Fusion ANFIS models based on multi-stock volatility causality for TAIEX forecasting, Neurocomputing (2009) 3462–3468.
[31] S.H. Chen, Genetic Algorithms and Genetic Programming in Computational Finance, Kluwer Academic Publishers, Dordrecht, 2002.
[32] R. Engle, GARCH 101: the use of ARCH/GARCH models in applied econometrics, The Journal of Economic Perspectives (2001) 157–168.
[33] D.W. Patterson, Artificial Neural Networks: Theory and Applications, Prentice Hall, 1996.