

Stock prices forecasting based on wavelet neural networks with PSO

Kai-Cheng Wang^{1,a}, Chi-I Yang¹ and Kuei-Fang Chang²

¹*Ph.D. Program in Mechanical and Aeronautical Engineering, Feng-Chia University, Taichung 40724, Taiwan*

²*Department of Applied Mathematics, Feng-Chia University, Taichung 40724, Taiwan*

Abstract. This research examines the forecasting performance of wavelet neural network (WNN) model using published stock data obtained from Financial Times Stock Exchange (FTSE) Taiwan Stock Exchange (TWSE) 50 index, also known as Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX), hereinafter referred to as Taiwan 50. Our WNN model uses particle swarm optimization (PSO) to choose the appropriate initial network values for different companies. The findings come with two advantages. First, the network initial values are automatically selected instead of being a constant. Second, threshold and training data percentage become constant values, because PSO assists with self-adjustment. We can achieve a success rate over 73% without the necessity to manually adjust parameter or create another math model.

1 Introduction

Wavelet neural network (WNN) was first proposed by Zhang and Benveniste in 1992 [1]. It is a special feed-forward network supported by the wavelet theory. Wavelet neural network is not just confined to the analysis of the stock market; it is also widely used in financial derivatives market, gold market and so on [2, 3]. Zhang Qian and Gao Liqun (2002) used wavelet noise filtering technique and least square method to predict price of Denghai Seed industry (stock code: 002041) shares in China [4].

In Taiwan, most investors prefer holding mutual funds over other assets in their investment portfolio. Since the majority of Taiwan's mutual funds are stock-based, there is value in developing a model using opening and closing stock prices. We believe this model should possess qualities such as universality and ease of use.

The reason why neural networks had been widely adapted is because its distinctive characteristic of simulating the human brain thought process, having the ability to learn and solve problems. Through artificial neural network, parallel computing of the biological neural system can be achieved, allowing high volume data analysis within a short period of time. Now, used across multiple disciplines, stock price prediction is one among the many it's used in. Universality is the goal for models. In our earlier researches [5], we combined second-order autoregressive mode (AR(2)) with

^a Corresponding author : gtotony98@gmail.com

wavelet analysis and its relative techniques. Unfortunately, this method was extremely limited by concavity and concavity distribution. However, WNN's network architecture is constructed with many wavelons through combination of weighted wavelons. Therefore, it is naturally unaffected by concavity.

The WNN consists of three parts: parameter; network architecture; and forecasting method. Generally, WNN has five parameters: weight; translation; dilation; threshold; and adjustment. We constructed a model which combines PSO. This method has two advantages. First, network initial values are automatically selected instead of a constant. Second, the threshold and training data percentage become constant values, because after applying PSO, the remaining three of five parameters can self-adjust to the algorithm. Thus, the model presented in this paper has strong operational ease. In 2000, Y. Oussar's research indicated that weight, translation, and dilation parameters shall be decided by the stock price's minimum and maximum values. Experiment results proved, when the initial value choice-making methodology suggested by Y. Oussar was used, not only does the calculation fall into an infinite loop often, the forecasted stock price value also exceeded the government enforced limit-up and limit-down policy, which is non-coherent with Taiwanese laws.

2 Methodology

We collected data of 41 stocks from the Taiwan 50 (Table 1) between 2010/07/15~2010/12/16, 100 trading business days. The daily stock closing price was chosen for input because the closing price reflects all the activities of the index on that day [6]. This means, for each company, a total of 100 stock prices, one from each trading day, is inputted for calculation. During experimentation, 10% of the data points from each stock price data were adopted as the training set. The remaining 90% were used for testing. This 10% was set as the training data percentage parameter. The threshold parameter was 0.01, constant without change. Figure 1 shows the main procedures of our research. The inputs and outputs of each block are further detailed in the material that follows

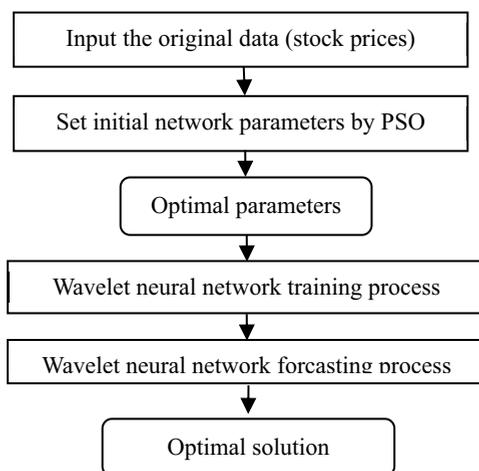


Figure 1. WNN with PSO simple flow chart.

Table 1. 100 times results from 41 Stocks of Taiwan 50.

2010/07/15~2010/12/16	Training data percentage=10% Threshold= 10^{-2}		
Stocks (symbol)	PSO(B=C=2) < WNN's RMSE [The number of Success]	WNN's RMSE	PSO(B=C=1.8) < WNN's RMSE [The number of Success]
Yuanta Financial Holdings (2885)	100	1.6988	65

Ctbc Financial Holding Co., Ltd.(2891)	100	2.4007	100
China Steel Corporation(2002)	100	0.9933	100
Compal Electronics, Inc.(2324)	100	1.3612	100
Advanced Semiconductor Engineering,Inc.(2311)	98	2.1026	100
AU Optronics Corp.(2409)	100	0.7067	100
Formosa Chemicals & Fibre Corporation(1326)	0	4.6481	0
Taiwan Cement Corp.(1101)	100	0.6244	100
Taiwan Fertilizer Co., Ltd.(1722)	75	2.9549	1
Formosa Plastics Corporation(1301)	99	3.7825	100
Formosa Petrochemical Corp(6505)	100	4.4747	100
Delta Electronics, Inc.(2308)	100	2.1597	100
Taiwan Semiconductor Manufacturing Co., Ltd.(2330)	0	3.7634	99
Taiwan Mobile Co., Ltd.(3045)	0	0.5353	0
Cheng Shin Rubber Ind., Co., Ltd.(2105)	58	1.4546	98
Sinopac Financial Holdings Company Limited(2890)	100	2.2117	100
Mega Financial Holding Company Ltd.(2886)	100	2.3221	100
Lite-On Technology Corp.(2301)	99	2.1014	100
Acer Incorporated(2353)	98	4.4261	100
HTC Corporation(2498)	100	42.1494	100
Asia Cement Corporation(1102)	100	1.9381	100
Siliconware Precision Industries Co., Ltd.(2325)	0	1.0993	0
EVA Airways Corporation(2618)	100	0.6122	100
Cathay Financial Holding Co., Ltd.(2882)	0	0.9139	82
Uni-President Enterprises Corp.(1216)	0	0.5644	0
President Chain Store Corporation(2912)	0	4.4315	0
Fubon Financial Holding Co., Ltd.(2881)	100	1.7556	100
Inotera Memories, Inc.(3474)	0	0.4160	99

Hua Nan Financial Holdings Co., Ltd.(2880)	100	0.9318	100
Asustek Computer Inc.(2357)	100	24.0938	100
China Development Financial Holding Corp.(2883)	100	1.9064	100
Shin Kong Financial Holding Co.,Ltd.(2888)	100	2.2873	100
Far Eas Tone Telecommunications Co., Ltd.(4904)	100	3.1666	100
Quanta Computer Inc.(2382)	98	2.3954	100
Wistron Corporation(3231)	100	1.7876	3
Synnex Technology International Corp.(2347)	0	2.5592	0
MediaTek Inc.(2454)	98	26.6667	2
United Microelectronics Corp.(2303)	1	0.3137	99
Hon Hai Precision Ind. Co., Ltd.(2317)	0	2.3107	0
Foxconn Technology Co., Ltd(2354)	0	6.7135	0
Pou Chen Corporation(9904)	90	0.3341	100
The number of success stocks	28		32
41 stock's success rate	68%		73%

2.1 Practical swarm optimization

In 1995, J. Kennedy and R. C. Eberhart were the first to introduce PSO[7]. Originally, it was for establishing an algorithm simulating social behaviors based on the practical example of simulating behaviors of foraging birds. Later on, it was used for solving optimization problems. The PSO calculation process and parameters used in this research are as follows.

Assuming there is D number of stock prices, each price is a point in the D-dimensional space. Location of the i -th point can be presented as:

$$X_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{iD})$$

The best location that the i -th price had passed can be recorded as:

$$P_i = (p_{i1}, p_{i2}, p_{i3}, \dots, p_{iD})$$

The component closest to the target can be found by analysing p_1 through p_D , changing the component to P_{gd} .

The i -th price's velocity is recorded as:

$$V_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{iD})$$

The two formulas below simulate the behaviour pattern of the bird flock:

$$V_{id} = \omega \times V_{id} + c_1 \times \text{rand}() \times (P_{id} - X_{id}) + c_2 \times \text{Rand}() \times (P_{gd} - X_{id}) \quad (1)$$

$$X_{id} = X_{id} + V_{id} \quad (2)$$

Table 2. PSO's parameters.

Parameters	Description
V_{id}	Velocity of particle at ith
ω	Weight. $\omega = 1$.
C_1 and C_2	Acceleration factor. $c_1 = c_2 = 2$
$\text{rand}()$ \ ` $\text{Rand}()$	Random number between 0 and 1
P_{id}	i -th best position of particle
X_{id}	i -th current position of particle
P_{gd}	Best position of swarm

Based on the data above, (1) was used to update particle velocity. Then, iterated in (2) for the new location.

2.2 Using PSO to optimize the initial values of WNN

Five network parameters were assigned for this WNN: weight parameter w ; translation parameter t ; dilation parameter d ; adjustment parameter \bar{y} ; and threshold value. The first three parameters, weight parameter w , translation parameter t , and dilation parameter d , were the optimized parameters. The method of implementation was:

Step 1: The initial values of w , t , and d were assigned based on the parameters developed by Y. Oussar [8], shown below:

$$w: \text{Set as } 0, \quad t: (M+m)/2, \quad d: (M-m)/5$$

When m is the original data's minimum value, M is the maximum value. There were 100 initial values for each parameter.

Step 2: The PSO parameters chosen were based on Table 2 [9, 10].

Step 3: A range must be specified when using PSO to optimize the three initial value parameters w , t , and d .

1. Range of weight parameter: $[-m/A, m/A]$.
2. Range of translation parameter: $[m, Bt]$.
3. Range of dilation parameter: $[-Cd, Cd]$.

Various statistics were used to test how "small" the weight needed to be. The experimental results were best when $A=200$. We have found that the same stands true for the translation parameter and dilation parameter ranges. This was the most significant discovery in this research, the usage of smaller values for t and d after much experimenting, performing favorably in the forecast of stock prices. In this experiment, setting $B=1.8, C=1.8$ produced better experimental results compared to our previous works ($B=2, C=2$) [11]. However, both B and C have low bounds. By setting $B=1$ or $C=1$, WNN fell into infinite loops. We also found that the translation parameter values must all be positive for best performance. The range of PSO parameters cannot be

too much beyond the original range, thus the minimum value of stock price data were selected as the left end point, and the right end point assigned was double or less the original t . The range of dilation parameter can be slightly greater than the original parameter. When B and C were assigned the values of 1 and 2.5, the results were extremely poor. When substituted with 1.8 and 2, the results were much more favorable. With the above mentioned, we suggest assigning B and C with values between 1.5 and 2.

Step 4: To obtain the optimized value of w , t , and d , we ran the PSO program loop using the statistics gathered from step 1 through step 3. Objective function is $1/(w_i \psi(x_i - t_i/d_i) + \bar{y} - x_i)$. The 100th generation is set to be the stopping generation. Since 100 stock prices were inputted, a corresponding 100 sets of w , t , and d were generated as well, vastly different from Y. Ousar's experiment, which had identical w , t , and d values. The optimized WNN parameters are $\{w_i\}$, $\{t_i\}$, and $\{d_i\}$.

2.3 Wavelet neural networks

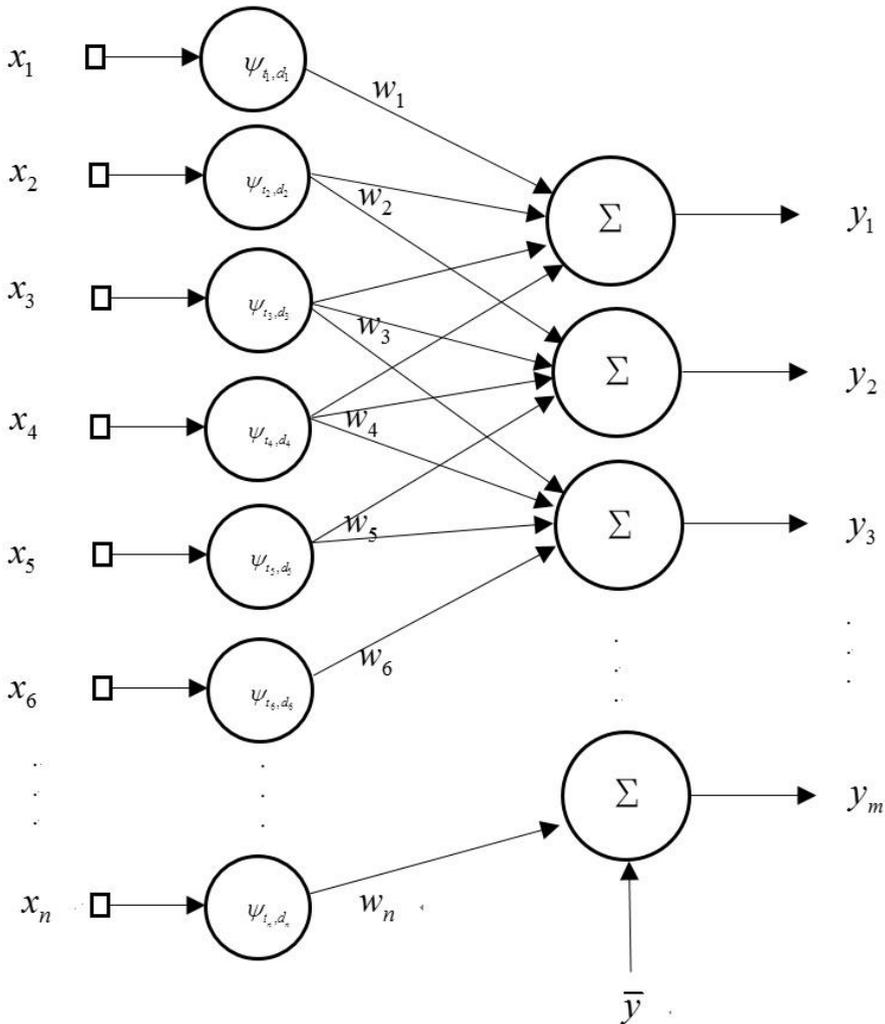


Figure 2. Wavelet Neural Network Structure.

There are many types of artificial neural networks (ANNs). We say that an ANN is a wavelet neural network (WNN) where the activation function is a wavelet [12]. In order to construct a WNN, the number of hidden layers as well as the number of neurons in these hidden layers must be provided. The advantage of the WNN structure proposed in this paper is that neither of these parameters is required (Figure 2). This model uses the Morlet wavelet as the activation function, and there are two main reasons.

First, using wavelet is simpler than using Fourier basis. To understand the first reason, one must first understand the differences between sine/cosine waves and wavelets. Based on the fact that reconstructing signal f from dilations of a fixed sinusoidal function $x \rightarrow e^{2\pi i x}$ by $f(x) = \int_{\mathbb{R}} e^{2\pi i \xi x} f(\xi) d\xi$, Fourier transform of f can be treated as the amount of oscillation $x \rightarrow e^{2\pi i x}$ while presenting signal f . However, sine/cosine waves are localized in the frequency domain, but not in time domain. This means that Fourier bases suffer from nonlocality; and nonlocality will interfere with the behavior of Fourier transform. Comparatively, there are more signal representation interference present than in wavelet algorithm. Two issues arise:

Can the use of Fourier transform provide the information needed? The behavior of signal in a time interval (no matter how small it is) influences the “global” behavior of Fourier transform. This means $f(t), f(\xi)$ cannot both be localized in their respective spaces (Heisenberg’s uncertainty principle).

Since sine/cosine waves are infinitely supported in time domain, Fourier bases generally cause high frequency oscillation near the jump point and the endpoints of the interval. The high oscillation causes more time complexity here, than it would in wavelet algorithm, especially, when the signal is stocks.

Luckily, wavelets are localized in both frequency domain and time domain. They can be designed to have desired properties such as near optimal time-frequency localization, compact support, and regularity. Stock signal normally exists as high localized oscillation in time domain. Since, wavelets can be designed to have finitely supported localized oscillation in time domain, using wavelets to represent “bad” signals such as stocks is more advantageous than using Fourier bases. Those are the main reasons why the previous algorithm based on Fourier bases was abandoned.

Secondly, the Morlet wavelet is very useful for the detection of system nonlinearities because of its good support in both frequency and time domain. In practice, the Morlet wavelet is the product of a complex exponential wave and a Gaussian envelope. Based on Figure 3, one can see that the Morlet wavelet enjoys not only advantages of Fourier bases, but also localization in both time and frequency domain, since it is a wavelet.

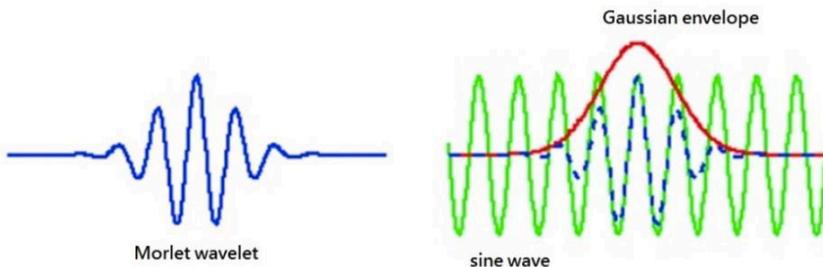


Figure 3. Morlet wavelet, sine wave and Gaussian envelope[16].

Wavelons are basic units of the WNN. The input data was x , with t as the translation parameter and d as the dilation parameter. The Morlet wavelet was used to obtain wavelons.

$$\psi_{t,d}(x) = \psi\left(\frac{x-t}{d}\right).$$

$$\psi(x_i) = \sqrt{e} \cdot k_i \cdot e^{-k_i^2/2} \quad (3)$$

where $k_i = (x_i - t_i) / d_i$. The following process describes how a set of data can be used via WNN to generate a forecast. When a set of data $X = (x_1, x_2, x_3, \dots, x_n)$ is inputted, it would be translated, using $T = (t_1, t_2, t_3, \dots, t_n)$, and dilated, using $D = (d_1, d_2, d_3, \dots, d_n)$, according to the parameters.

$$\frac{x_i - t_i}{d_i}, i = 1, 2, 3, \dots, n. \quad (4)$$

Each calculated data from (4) would be entered into (3). Linear combinations would be performed with weight vector $W = (w_1, w_2, w_3, \dots, w_n)$, and values would be calculated by (5) to obtain (6).

$$\psi_{t_i, d_i}(x_i) = \psi\left(\frac{x_i - t_i}{d_i}\right), i = 1, 2, 3, \dots, n \quad (5)$$

$$\sum_{i=1}^n w_i \psi\left(\frac{x_i - t_i}{d_i}\right) \quad (6)$$

Use (6) and adjustment parameter \bar{y} to obtain the output data y , shown in (7). Adding \bar{y} is required for (7) because of $\int_{-\infty}^{\infty} \psi = 0$. To avoid low convergence rate, adjustment parameter \bar{y} was added.

$$y = \sum_{i=1}^n w_i \psi\left(\frac{x_i - t_i}{d_i}\right) + \bar{y} \quad (7)$$

2.4 WNN training process

The WNN architecture is composed of weighted linear combinations of many wavelons. Supervised learning method was used to train our WNN, and backpropagation algorithm was used to correct network parameters [6, 13, 14].

The WNN training process is as follows:

Step 1: Input data to the WNN.

Step 2: Five settings were assigned to the WNN, namely the w , t , and d after PSO optimization, as well as the adjustment parameter \bar{y} , and the threshold value.

Step 3: Input all data and network parameters and use WNN to calculate the network output values.

Step 4: Calculate the error between the network output value and x using error function E .

$$E = \frac{1}{2} \sum_k (x_k - y_k)^2. \quad (8)$$

Step 5: Determine if the error is within the given threshold value:

1. if yes, proceed to the next step.

2. if not, use the L-M Algorithm to adjust the network parameters and return to Step 3.

Repeat the network output value calculation until the error falls within the given threshold value.

Step 6: Verify whether data still requires training:

1. if yes, return to step 3.
2. if not, the training is complete.

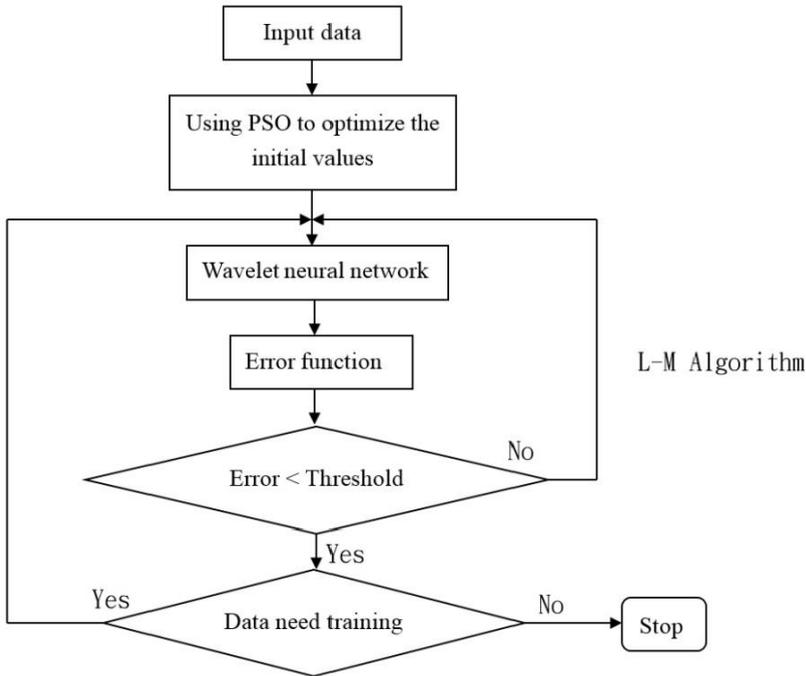


Figure 4. WNN Training Flowchart.

2.5 WNN forecasting process

Our model uses each stock’s 100 closing prices to forecast the following 10 days’ stock prices. This research used 10% of the training result data for calculation. The chosen 10% were the last 10% of the calculated w , t , and d training results produced. Therefore, the parameters used for calculation were $(w_{90}, w_{91}, \dots, w_{99}) \cdot (t_{90}, t_{91}, \dots, t_{99}) \cdot (d_{90}, d_{91}, \dots, d_{99})$ and \bar{y} . Applying the parameters listed in (7), the below derived:

$$y = \sum_{90}^{99} w_i \psi\left(\frac{x_i - t_i}{d_i}\right) + \bar{y} \tag{9}$$

This is our WNN forecasting function (9).

$$x_{101} = \sum_{90}^{99} w_i \psi\left(\frac{x_{i+1} - t_i}{d_i}\right) + \bar{y} \tag{10}$$

$$x_{102} = \sum_{90}^{99} w_i \psi\left(\frac{x_{i+2} - t_i}{d_i}\right) + \bar{y} \tag{11}$$

$$x_{110} = \sum_{90}^{99} w_i \psi\left(\frac{x_{i+10} - t_i}{d_i}\right) + \bar{y} \tag{12}$$

Using $x_{101}, x_{102}, \dots, x_{110}$, stock price of the following 10 days were forecasted.

3 Results and discussions

In Y. Oussar's research, although the initial parameters change along with stock prices taken from different time intervals, but in fact, the parameters w , t , and d remain the same for each data point in the same set of training data. This is not very logical, because it is impossible for all data points to have the same initial value. One disadvantage is the infinite loop that occurs when using the backpropagation algorithm during training. Therefore, the assumption is, if more suitable initial parameters can be found for each data point, the problem should be resolved. The resolution we found was the usage of PSO. Before the initial parameters are assigned for the WNN training process, PSO can analyze each data point, optimizing the corresponding three initial parameters. When this is done, not only does the speed for training increases, the RMSE becomes smaller and the forecast accuracy increases.

In Table 1, we compared our algorithm, which uses PSO optimization, to Y. Oussar's algorithm. The comparison method was running the program 100 times, recording each instance where the RMSE is less than the original RMSE. When the RMSE was less more than 65 times out of 100, we considered using WNN with PSO as better performance than using Y. Oussar's algorithm. For example, when we used China Steel Corporation (2002) to test forecast, the RMSE was consistently less than 0.9933, so it was considered better performing. After running the program using $B=C=2$, we found 28 companies that had less RMSE more than 65 times. Then, an even smaller range of parameters for were used, $B=C=1.8$, presenting 30 companies, successfully raising the success rate from 68% to 73%, shown in the last row of Table 1.

This algorithm can not only be used to forecast Taiwanese stocks, but can also be used to forecast other nations' stocks, commodities, and other financial instruments. For testing, we randomly selected three gold index companies: Barrick Gold Corp (ABX); Agnico-Eagle Mines Ltd (AEU); and Anglogold Ashanti Ltd (AU). Data collected were during time interval chosen was March 6, 2014 through August 11, 2014. In Table 3, it shows a success rate of 67%. Then, three large American corporations were randomly chosen: IBM; Facebook; and Apple. Data collected were during time interval February 25, 2014 through July 31, 2014 (Table 4). For both algorithms, there were no problems running data of IBM and Facebook.

Table 3. 100 times results from Three Gold Companies.

Company Name	Training data percentage= 10% threshold= 10^{-2}	
	WNN with PSO < WNN RMSE [The number of Success]	WNN RMSE
Barrick Gold Corp	100	0.8403
Agnico-Eagle Mines Ltd	0	2.5956
Anglogold Ashanti Ltd	100	1.7455

Table 4. 100 times results from Three American Companies.

Company Stock	Training data percentage= 10% threshold= 10^{-2}	
	WNN with PSO < WNN RMSE [The number of Success]	WNN RMSE
IBM	100	3.5754
Facebook	100	7.5791
Apple	One of RMSE = 56.7879	N/A

However, while running the program for Apple, the data produced from this specific experiment was what led us to affirmatively believe that this algorithm is better than Y. Oussar's. When data for Apple was calculated, Y. Oussar's algorithm fell into infinite loops. Looking into the incident further,

we found it was because Apple had a stock split of 1:7 on June 9, 2014. The stock price changed from \$645.57 to \$92.25. Our algorithm, using WNN with PSO, accommodated this drastic change. This is a great advantage. This indicates many future possibilities for this algorithm, such as the possibility of creating an extra step in the model or revising the existing error function to compensate for a stock split. Although, we do point out, in this calculation, there is a dramatic difference in the RMSE value, showing the test results were invalid.

We also realized two major limitations of the algorithm. First, due to the fact that this model was made to have higher adaptability, the accuracy rate is lower. When compared to Ayodele Ariyo Adebisi's model[15], created a model specific designed to forecast Dell stock, during a ten day interval, our model was not as accurate. In ANN vs. WNN, our model had five lower error values. In ARIMA vs. WNN, our model had only four lower error values. However, if Adebisi's model was used to forecast the stock price of any other company, the algorithm or the model itself may need complete redesign. Since their model was targeted while our model was not, this error value difference was acceptable and expected (Table 5).

Table 5. Forecasting Value and Error of Three Methods.

Original chart without WNN columns from Adebisi's paper

DELL (2010/03/01-2010/03/12)							
Sample period	Actual value	Forecasting value			Forecasting error		
		ANN	ARIMA	WNN	ANN	ARIMA	WNN
03/01	13.57	13.16	13.35	13.46	0.03021	0.016212	-0.0081
03/02	13.68	13.55	13.46	13.35	0.0095	0.016082	-0.0239
03/03	13.71	13.7	13.56	13.55	0.00073	0.010941	-0.0116
03/04	13.67	13.55	13.67	13.8	0.00878	0	0.0096
03/05	13.88	13.53	13.78	14.08	0.02522	0.007205	0.0146
03/08	14.01	13.89	13.88	14.06	0.00857	0.009279	0.0035
03/09	14.18	13.92	13.99	14.18	0.01834	0.013399	-0.0002
03/10	14.31	13.85	14.09	14.51	0.03215	0.015374	0.0139
03/11	14.21	14.18	14.2	14.63	0.00211	0.000704	0.0295
03/12	14.26	14.31	14.3	14.93	-0.0035	-0.00281	0.047

The other difficulty is that the range for the PSO parameter is difficult to choose. When the range is too small, the optimal solution may be excluded. When the range is too large, more time is wasted in search of the optimal solution. Therefore, finding a better method of choosing a proper range is an area that needs to be studied.

The limitation of research we had was choosing the time interval for the training data. Dates where significant events that would affect the stock or stock market dramatically had occurred must be avoided. Natural disasters, stock market crashes, or any other significant events such as the Apple stock split will cause the model to fail. We don't have enough manpower to search for all possible events during the 100 days in order to avoid certain time intervals.

4 Conclusion

Our experiment results indicated using smaller range of numbers for the translation and dilation parameters increased performance during forecast. Among some older methods for WNN operations [10,11,13], the network threshold parameters were set as constants, and proper forecast was obtained

by adjusting threshold and training data percentage. Our WNN adds PSO and resolves this slightly illogical method. The network initial values of every training data point can be configured by PSO in coordination with WNN, which means the network initial values are “automatically” selected instead of being a constant, and prevents infinite loops during calculation. Threshold and training data percentage become constant values, because PSO provides self-adjustment, which means it is not a true constant during calculation.

From the two advantages mentioned above, our model has higher ease of operation in comparison to the older WNN models, because the end users do not have to set any parameters. The necessary threshold and training data parameters would be chosen for them. If an end user wants to know the following 10 days’ stock price, the only data that needs input are the company name and symbol, and the forecast prices will be presented for consideration. In the future, we hope to improve our results by revising the existing training methods or adding additional steps.

Disclosure

The authors declare that there is no conflict of interests regarding the publication of this article.

References

1. Z. Qinghua, IEEE Trans. on Neural Network, **3**, 6 (1992)
2. H. Dongdong and Z. Wenhong, BIFE, 32-36 (2011)
3. H. Zhang and H. Lin, J. Southeast Univ. Nat. Sci., **37** (4), 716-720 (2007)
4. Y. Zhao, Y. Zhang, and C. Qi, PACIA, **1**, 31-36 (2008)
5. H.R. Ke, K.C. Wang, C.I. Yang, and K.F. Chang, IJAPM, **4** (1), 9-14 (2014)
6. C.J. Lin and H.M. Tsai, Math. Comput. Model., **47** (9-10), 982-996 (2008)
7. J. Kennedy, *Encyclopedia of Machine Learning*, (Online, 2011)
8. Y. Oussar and G. Dreyfus, Neurocomputing, **34** (1), 131-143 (2000)
9. Y. Shi and R.C. Eberhart, IEEE WCCI, 69-73 (1998)
10. Y. Shi and R.C. Eberhart, EP98, 591-600 (1998)
11. C.I. Yang, K.C. Wang, and K.F. Chang, ICDIP, 91590V (2014)
12. J. Zhang, G. Walter, Y. Miao, and W.N.W. Lee, IEEE Trans. Signal Process., **43** (6), 1485-1497 (1995)
13. Y. Wang and X. Cheng, EMEIT, **6**, 3109-3112 (2011)
14. F. Zhang, H. Dai, and D. Tang, J. Appl. Math., 2014 (2014)
15. A.A. Adebisi, A.O. Adewumi, and C.K. Ayo, J. Appl. Math., 2014 (2014)
16. <http://paos.colorado.edu/research/wavelets/wavelet2.html>