Global Journal of Computer Science and Technology
Network, Web & Security
Volume 13 Issue 2 Version 1.0 Year 2013
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Convergence of Actual and Predicted Share Prices – An ADALINE Neural Network Approach

By Ravindran Ramasamy & Tan Chee Siang
University Tun Abdul Razak

Abstract - Accurate forecasting of share prices is needed for fund managers and institutional investors for hedging decisions. Robust forecasting results will not only increase the effectiveness of hedging and reduce the hedging costs but also provide benchmarks for controlling and decision making. Existing traditional models for forecasting share prices rarely produce fair results. In this paper we have applied neural net work ADALINE approach to forecast the share prices listed in the Malaysian stock exchange. Adaptive linear neural net uses a moving window approach in updating its weights while training and this improves the accuracy of forecasting. We applied this technique on four share prices at four learning rates and the results nicely converge with the actual prices at higher learning rates. Our findings will increase the confidence in forecasting and will be helpful for stakeholders immensely.

Keywords : adaline, learning rate, neuron, neural network, share return, synapse.

GJCST-E Classification : I.2.6

CONVERGENCE OF ACTUAL AND PREDICTED SHARE PRICES AN ADALINE NEURAL NETWORK APPROACH

Strictly as per the compliance and regulations of:

Convergence of Actual and Predicted Share Prices – An ADALINE Neural Network Approach

Ravindran Ramasamy [α] & Tan Chee Siang [σ]

Abstract - Accurate forecasting of share prices is needed for fund managers and institutional investors for hedging decisions. Robust forecasting results will not only increase the effectiveness of hedging and reduce the hedging costs but also provide benchmarks for controlling and decision making. Existing traditional models for forecasting share prices rarely produce fair results. In this paper we have applied neural net work ADALINE approach to forecast the share prices listed in the Malaysian stock exchange. Adaptive linear neural net uses a moving window approach in updating its weights while training and this improves the accuracy of forecasting. We applied this technique on four share prices at four learning rates and the results nicely converge with the actual prices at higher learning rates. Our findings will increase the confidence in forecasting and will be helpful for stakeholders immensely.

Keywords : adaline, learning rate, neuron, neural network, share return, synapse.

I. Introduction

Forecasting is an important task the fund managers perform for decision making and controlling especially very important for those who are managing other people's money like fund managers. With the uncertain future, the manager needs to have a set of guidelines and tools in assisting him to predict the future movement of financial time series like share prices (Yoon and Swales, 1991; Thomaidis and Dounias, 2007).

Investments are made with the objective of maximizing the return and simultaneously reducing the risk (Banz, 1981; Hirt and Block, 1996). The Sharpe ratio gives the investors how much they earn for every unit of risk they face (Jones, 2007). The mutual fund managers' objective is to maximize the return, minimize the risk and in addition they have to guarantee the safety of the funds invested by hedging. Several hedging tools are available for a fund manager presently and he has to select the best tool with minimum cost and fewer complexities to manage. All these require a well balanced efficiently forecasted share prices. The forecasted prices not only serve the purpose of hedging but also they help in controlling and decision making (Mitchell and Pavur, 2002) whether to buy or hold or sell.

The objective of this paper is to apply ADALINE neural network technique to forecast the share prices. Though several traditional techniques are available like moving averages, Bollinger bands, and chartist approach (Janssen, Langager and Murphy, 2011) they depend too much on the past data and they predict the future prices for a long period ahead with the same base data. The traditional linear regression technique (Grønholdt and Martensen, 2005) takes fundamental economic variables as independent and share price as dependent variable, fail to achieve good convergence because the independent variables are macro economic variables which slowly change but the share prices are dynamic and changes daily. This mismatch results in poor forecasting.

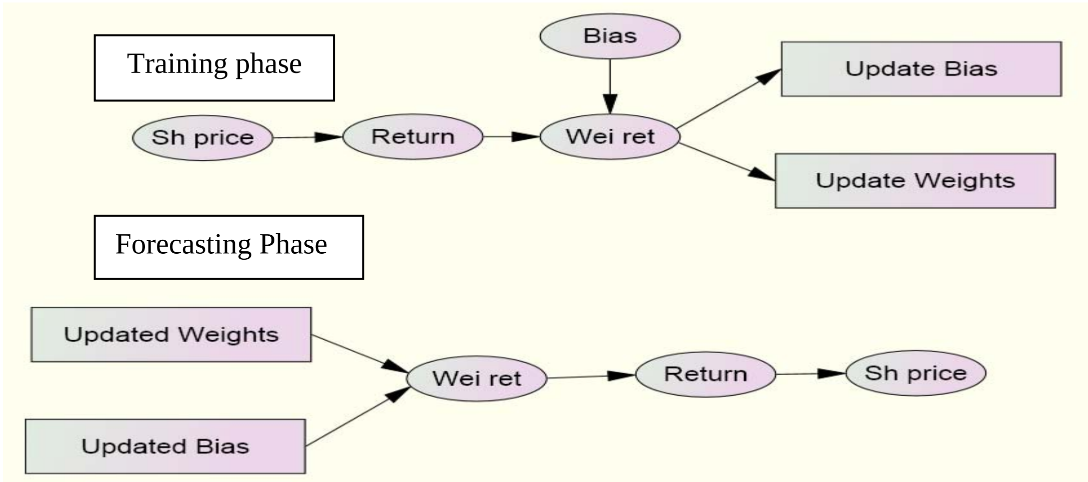
There are several plus points in applying neural networks to forecast the share prices. The first major merit is that it does not consider fundamental assumptions like normality of data (Aleksander and Morton, 1995), extreme data etc. All traditional statistical assumptions are absent here. In addition the neural nets always go for iterations which update the weights several times repeatedly with a learning rate which controls the weights of the neurons (Hecht-Nielsen 1989; Govindarajan and Chandrasekaran, 2007). Yet another advantage of neural net is the data memory issue. The old data becomes obsolete as the data has life cycle. The recent data is more useful than the oldest data. To capture this moving window technique is adopted in networks which ignore the oldest data and adds the new data for training and forecasting. This gives the required efficiency in forecasting.

II. Adaline Neural Network

Adaptive Linear Neuron known as ADALINE is a single layer neural network which is useful in predicting time series like share prices (Lin and Yeh 2009; Matilla-Garcia, and Arguello, 2005; Remus and O'Connor, 2001; Rude 2010). ADALINE is adopted with the assumption that the relationship between historical daily returns and the forecasted daily returns are linear and each of it carried different weight. The weight is not constant but ever changing when a new data arrives (Kaastra and Boyd, 1996).

Author α : Graduate School of Business University Tun Abdul Razak. E-mails : ravindran@unirazak.edu.my, klcp109005@pgs.unirazak.edu.my

ADALINE Architecture



Wei ret = weighted return

Figure 1 : Training and forecasting flow algorithm

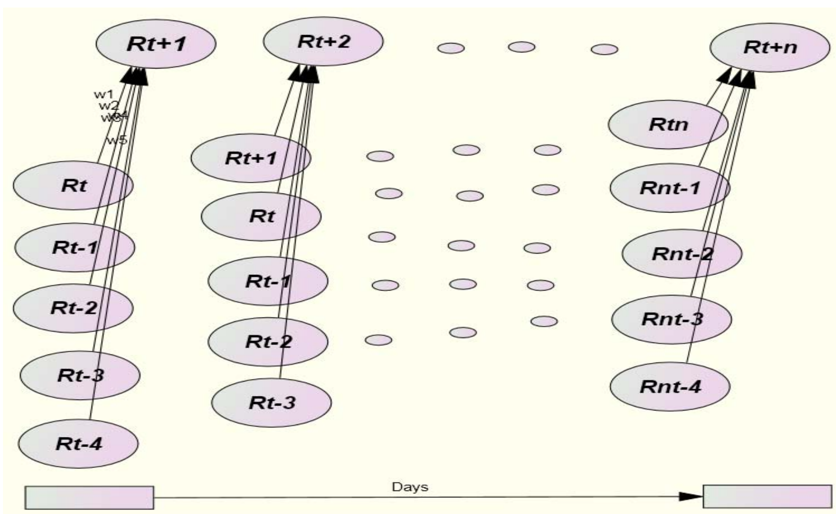


Figure 2 : Neuron and synapse connection in forecasting

ADALINE ALGORITHM

Given: *Share prices*

Initialise: *Small random weight for each neuron, a bias and a learning rate*

Convert: *Share prices to returns* $r = p_t / p_{t-1}$

Where r = *daily return*

p_t = *price today*

p_{t-1} = *previous day's price*

Iterate: *Until a condition is satisfied (say, 100 times)*

Compute the net input and keep it in y_i $y = \sum_{i=1}^{n} w_i * x_i + b$

where y = *forecasted daily return*

b = *bias*

w = *weight*

x = *historical daily return*

n = *number of synapse*

Update weights $w_{i(new)} = w_{i(old)} + alpha * (t-y)$

where $alpha$ = *Learning rate*

t = *target return* (6^{th} *day return)*

Update bias $b_{(new)} = b_{(old)} + alpha * (t\text{-}y)$

End iteration after 100 times

Forecasting: *Take the above updated weights and bias*

Iterate: *for 252 days (∵ stock market works for 252 days approximately)*

Compute the return $y = \sum_{i=1}^{n} w_i * x_i + b$

Convert: *Returns to Share price* $P = r * p_{t-1}$

The main reason to convert the daily share price to daily return is to avoid non-stationary nature of share price. Moreover the daily share price does not indicate whether the price is moving up or down. The positive sign or negative sign of the daily return will be useful in finding the hit rate. There are three stages in this study, i.e., initialisation phase, training phase and forecasting phase.

III. Initialisation Phase

At this phase the learning rate, number of neurons. Synapses, weights and bias are decided and given to the net for starting the computation process. The random weights and bias will change at every time we start the program. To avoid this we have set the random state as 10. This will make sure the random numbers are identical whenever or wherever the program is executed.

IV. Training Phase

The training of the network is performed through a windowing technique (Sapena, Botti, and Argente, 2003). The window will move as time progresses. The net will compute the activation value by multiplying the random weights and window of five returns and the result will be added to bias. This will be treated as the forecasted return. This value will be compared with the target the sixth day return to find variance. It is stored as error. This error with learning rate and the original sixth day's return and old weight all determine the new weight. This process will be repeated by dropping the oldest data and taking the newest data in updating weights till the end of the training set. Similar approach has been used by Buscema & Sacco (2000) in attempting to predict the stock market index returns. The same procedure is adopted by Refenes and Francis (1993) in predicting the currency exchange rate and (De Faria *et al*, 2009) in predicting the Brazilian stock market index.

V. Testing or Forecasting Phase

The training process will be carried out in forecasting phase also. First five returns will be taken from the December 2010 data and January 2011 first return will be computed and stored. Then as in training the error will be computed comparing final return with the predicted return. Then this error, final return of 2010 and the updated weight in the training phase all will decide the first new weight and bias for 2011. This procedure will be repeated for 252 days. Later all 252 returns will be converted to predicted share prices.

VI. Measurement of Effectiveness

The difference in actual and predicted prices is recorded for the purpose of performance evaluation. The root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE) and hit rate are computed and recorded as follows.

The RMSE is computed based on following formula

$$\varepsilon = \sqrt{\frac{\sum_{t=1}^{n} (x_t - y_t)^2}{n}}$$

Where ε = root mean squared error
n = total number of days
t = days
x_t = t[th] actual share price
y_t = t[th] forecasted share price

The MAE is computed based on the following formula

$$MAE = \frac{\sum_{t=1}^{n} |x_t - y_t|}{n}$$

The MAPE is computed based on the following formula

$$MAPE = \frac{MAE}{\left(\frac{\sum_{t=1}^{n} x_t}{n}\right)} x\ 100\%$$

VII. Hit Rate

The hit rate is one if the actual and predicted returns have the same sign. This shows the direction of prediction.

$$hr = \begin{cases} 1, & x \text{ and } y;\ same\ sign \\ 0, & x \text{ and } y;\ opposite\ sign \end{cases}$$

Average hit rate is computed as follows

$$hr = \frac{1}{n} \sum_{t=1}^{n} hr\,(x_t, y_t)$$

VIII. Sample, Analysis and Interpretation

With the above ADALINE architecture and methodology a MATLAB program was written to test the efficiency of neural networks forecasting time series, the share prices. Share prices of four companies listed in Malaysian stock exchange was selected for two years from yahoo finance and the star newspaper websites for 2010 an 2011. The share price of 2010 was considered as training sample and 2011 was retained for validation. To overcome the non-stationary problem, 2010 share prices were converted to returns. The returns forecasted for 2011 were reconverted to share prices and graphs were prepared to visually observe the convergence or divergence of price lines.

IX. AMMB

AMMB is a listed company in Malaysian stock exchange. The share price prediction results are given in the following table. The actual mean price for 2011 is RM 6.28. The average predicted prices at various learning rates are very close to the actual mean price except at the learning rate of 15%. At this level the forecasted mean price deviates more showing RM 7.02. The standard deviation gives the volatility or variation of share prices. The actual standard deviation is 0.34 and the forecasted volatilities are around 0.40 except at 15% learning rate, which is 0.43. The median price and range also move in tandem with mean price.

Table 1 : Descriptive statistics of actual and forecasted share prices of AMMB, 2011

	Actual Share price		Learning Rates Predicted share prices		
Parameters		0.15	0.35	0.55	0.75
Mean	6.28	7.02	6.59	6.53	6.56
Std Deviation	0.34	0.43	0.40	0.40	0.40
Median	6.36	7.05	6.64	6.57	6.60
Range	1.77	2.45	2.19	2.09	2.22
Maximum	7.15	8.28	7.63	7.55	7.59
Minimum	5.38	5.83	5.44	5.46	5.37
Correlation coefficients	--	0.55	0.73	0.77	0.78

The correlation coefficients show the relationship between the actual price and predicted prices at various learning rates. A high correlation indicates that the actual and predicted prices move in tandem and vice versa. At the learning rate of 75% the correlation is 78%. The correlation coefficients increase continuously as the learning rate increases. This implies at lower learning rates the actual prices and forecasted prices do not converge and more gaps is existing between them.

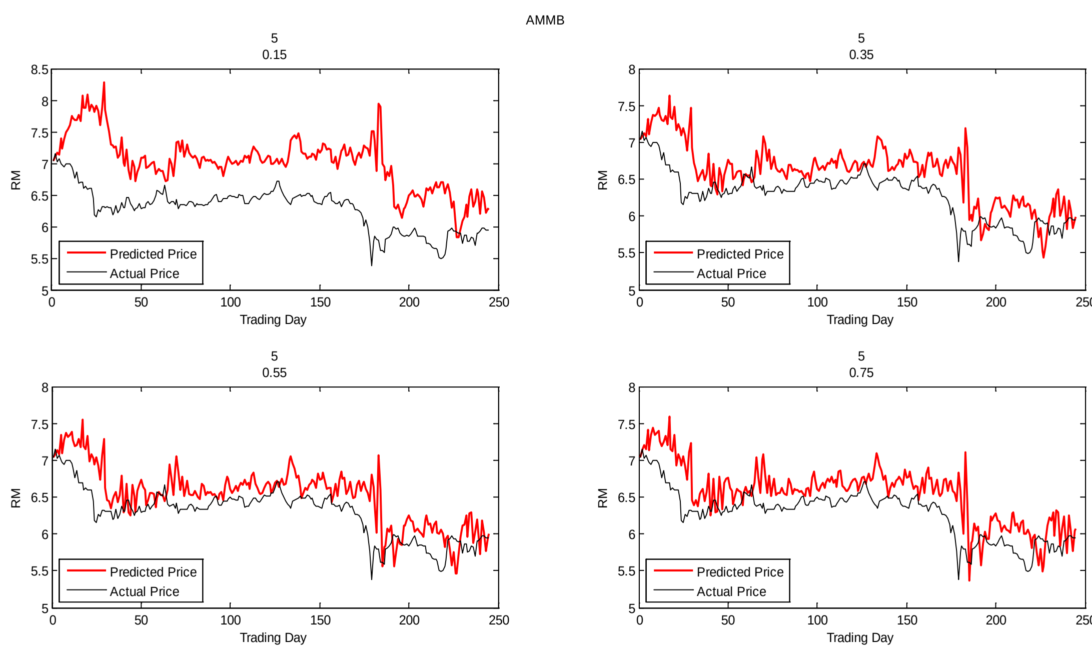


Figure 1 : Convergence of actual and forecasted share prices of AMMB

A close observation of the figures reveal at 75% of learning rate the forecasting is efficient because the forecasted price and the actual price converge almost perfectly. In January and February the forecasted line and the actual lines show a larger gap and later predicted line follows the actual price line closely. At lower levels of learning there is an appreciable gap between the lines which results in larger error.

Table 2 : Different types of errors produced at various learning rates - AMMB

Learning rates	RMSE	MAE	MAPE (%)	Hit Rate (%)
0.15	0.83	0.74	11.82	42.39
0.35	0.42	0.34	5.44	39.09
0.55	0.36	0.28	4.53	37.45
0.75	0.37	0.31	4.89	35.80

When the learning rate increases the error levels fall steeply. At 15% learning level the RMSE was 0.83 but in 75% learning level it decreased to 37%. The same trend is visible in MAE and MAPE. Hit rate also reduces but not as steep as other error measures. These results imply at higher learning rates the ADALINE neural net predicts the share prices more precisely.

X. Axiata

Axiata share prices are forecasted at different learning rates ranging from 15% to 75% for 2011 and the results are as follows.

Table 3 : Descriptive statistics of actual and forecasted share prices of Axiata, 2011

	Actual share price		Learning Rates Predicted share prices		
Parameters		0.15	0.35	0.55	0.75
Mean	4.89	4.95	4.91	4.89	4.88
Std Deviation	0.13	0.18	0.17	0.16	0.17
Median	4.89	4.95	4.92	4.90	4.88
Range	0.57	1.04	0.94	0.92	0.90
Maximum	5.14	5.54	5.37	5.36	5.35
Minimum	4.57	4.50	4.44	4.44	4.45
Correlation coefficients	--	0.05	0.25	0.30	0.33

The actual mean price for 2011 is RM 4.89 and the forecasted mean prices are very close to this price except at the learning rate of 15% which is RM 4.95. When the learning rate increases the mean prices are decreasing and come closer to actual mean price which implies at the higher learning rates the net learns better and forecasts better. The volatility is 0.13 for actual prices but for forecasted prices the volatility is slightly more. Median prices show similar trends as mean prices. The range also decreases when the learning rate increases. The correlation coefficients increase from 5% to 33% when the learning rates increase from 15% to 75%. These results imply that the net forecasts well in higher learning rates and the movements are also closer to actual prices.

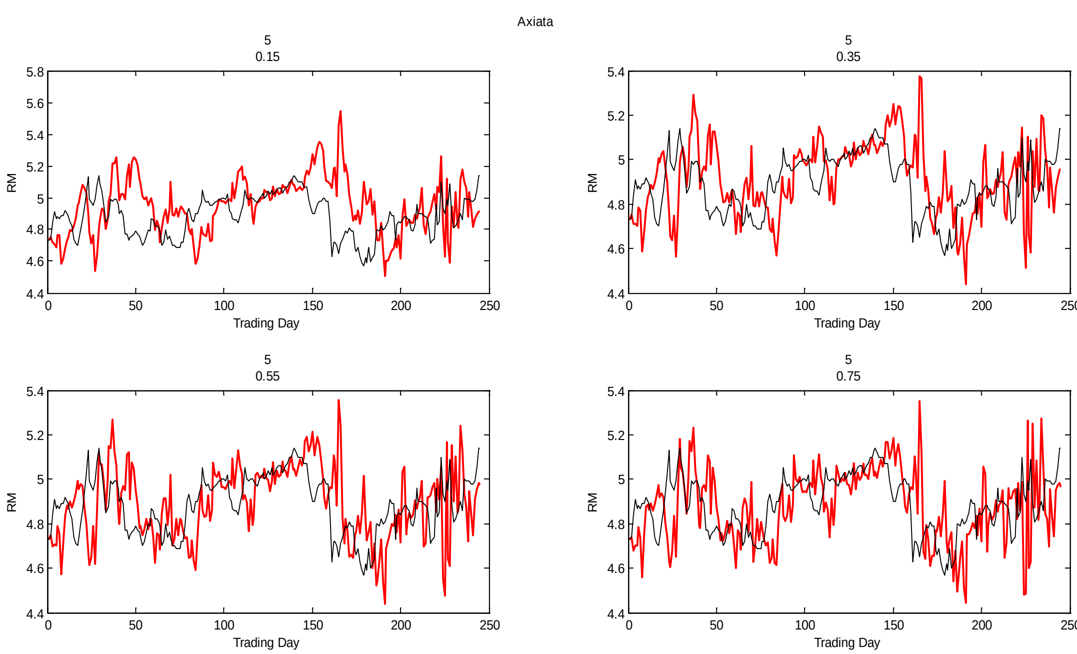


Figure 2 : Convergence of actual and forecasted share prices of Axiata

The thin black line shows the actual price and the thick line shows the predicted prices. It could be observed that both lines are moving in tandem capturing the same trend. However the thick line is more volatile and oscillates up and down more compared to the actual line. At 15% learning rate the lines diverge more than at 75% learning rate where the convergence is better.

Table 4 : Different types of errors produced at various learning rates - Axiata

Learning rates	RMSE	MAE	MAPE (%)	Hit Rate (%)
0.15	0.22	0.17	3.52	50.21
0.35	0.19	0.14	2.93	45.27
0.55	0.18	0.13	2.75	43.21
0.75	0.17	0.13	2.70	43.21

At the learning rate of 15% all errors are very high including the hit rate. When the learning rate increases the errors decline gradually but the hit rate falls steeply. The results indicate the net performs well in higher learning rates.

XI. HLB

HLB is another listed company in the Malaysian stock exchange. By applying the same procedure the share prices are predicted by the ADALINE net after training by the 2010 return data. The actual mean price is RM 11.07 for HLB in 2011 and at various levels of learning rates the forecasted mean prices are very close in the range of RM 11.03 to 11.09. The average price predicted at the learning rate of 15% is very low at RM 10.88. The standard deviation is also very high for this company price when compared to all other companies' standard devotions. Like mean prices the median prices also increase when the learning rate increases. The range is higher for the predicted prices than the actual price. The correlation coefficients between the actual and forecasted prices are strong around 85% to 89% except at 15% learning rate which indicates the actual and the forecasted prices move very closely in tandem. All these reveal that the net is producing robust results at higher learning rates.

Table 5 : Descriptive statistics of actual and forecasted share prices of HLB, 2011

	Actual Share price		Learning Rates Predicted share prices		
Parameters		0.15	0.35	0.55	0.75
Mean	11.07	10.88	11.03	11.08	11.09
Std Deviation	1.45	1.63	1.56	1.55	1.54
Median	10.58	10.46	10.58	10.64	10.68
Range	4.55	6.29	5.45	5.27	5.18
Maximum	13.76	14.81	14.35	14.31	14.23
Minimum	9.21	8.52	8.91	9.05	9.05
Correlation coefficients	--	0.71	0.89	0.88	0.85

The following figure shows the convergence of actual and predicted share prices for HLB. The first graph which is predicted at 15% learning rate shows wider gap between the actual and predicted prices. This gap reduces gradually with the same trend when the learning rate increases progressively. The convergence is excellent at 75% learning rate.

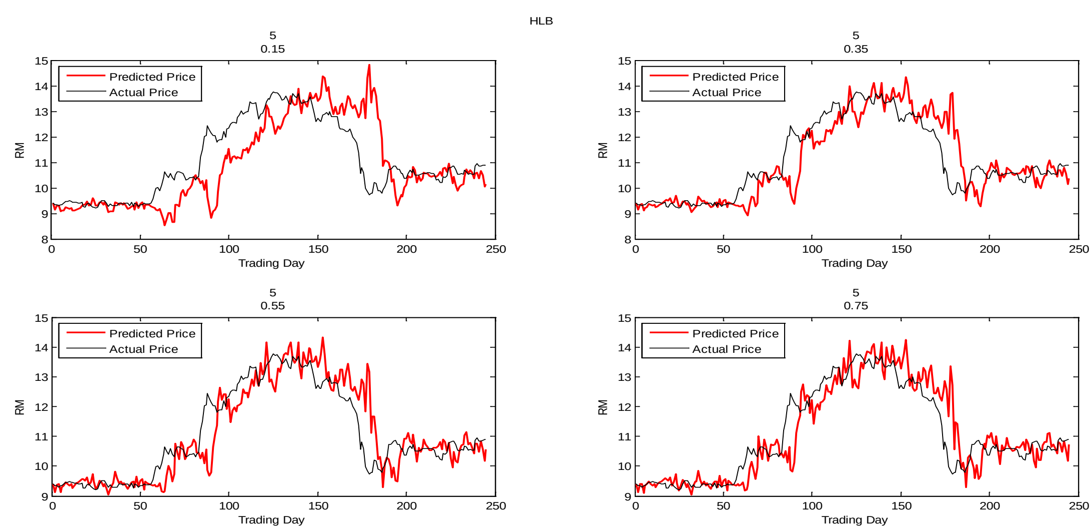


Figure 3 : Convergence of actual and forecasted share prices of HLB

Table 6 : Different types of errors produced at various learning rates - HLB

Learning rates	RMSE	MAE	MAPE (%)	Hit Rate (%)
0.15	1.19	0.79	7.17	41.56
0.35	0.84	0.54	4.88	45.27
0.55	0.74	0.49	4.39	45.68
0.75	0.69	0.46	4.18	45.27

The RMSE, MAE and MAPE all decease steeply when the learning rate increase from 15% to 75%. The RMSE declines from 1.19 to 0.69 almost a drop of 42%. The MAE and MAPE also fall by the same percentage. The hit rate behaves in an opposite way. When learning rate increases from 15% to 75% the hit rate also increases by 8% approximately. It seems at higher learning rates the net forecasts well.

XII. KLK

KLK is another listed company in Malaysian stock market. The mean actual price for this company in 2011 is RM 21.45 and the median price is RM 21.34, with a volatility of 0.67. The predicted mean prices are not very close to the actual mean price they differ substantially. At the learning rate of 15% the mean and median prices go up to RM 24.05 and RM 23.92 respectively. The range values are also higher. The correlation coefficient is 44% at the learning rate of 75% but registered a poor correlation coefficient of 9% at the learning rate of 15%.

Table 7 : Descriptive statistics of actual and forecasted share prices of KLK, 2011

	Actual Share price		Learning Rates Predicted share prices		
Parameters		0.15	0.35	0.55	0.75
Mean	21.45	24.05	23.42	23.01	22.78
Std Deviation	0.67	0.95	0.89	0.88	0.89
Median	21.34	23.92	23.31	22.91	22.66
Range	3.54	5.56	4.70	4.59	5.68
Maximum	23.10	27.27	26.26	25.41	25.56
Minimum	19.56	21.70	21.56	20.81	19.88
Correlation coefficients	--	0.09	0.35	0.42	0.44

The following figures also reveal the poor convergence of actual and predicted prices of KLK for 2011. The gap is substantial in 15% learning rate. When the learning rate goes up the gap between the actual and predicted price reduces a bit but not to the expected levels as in the other companies. The volatility is also very steep in the predicted lines.

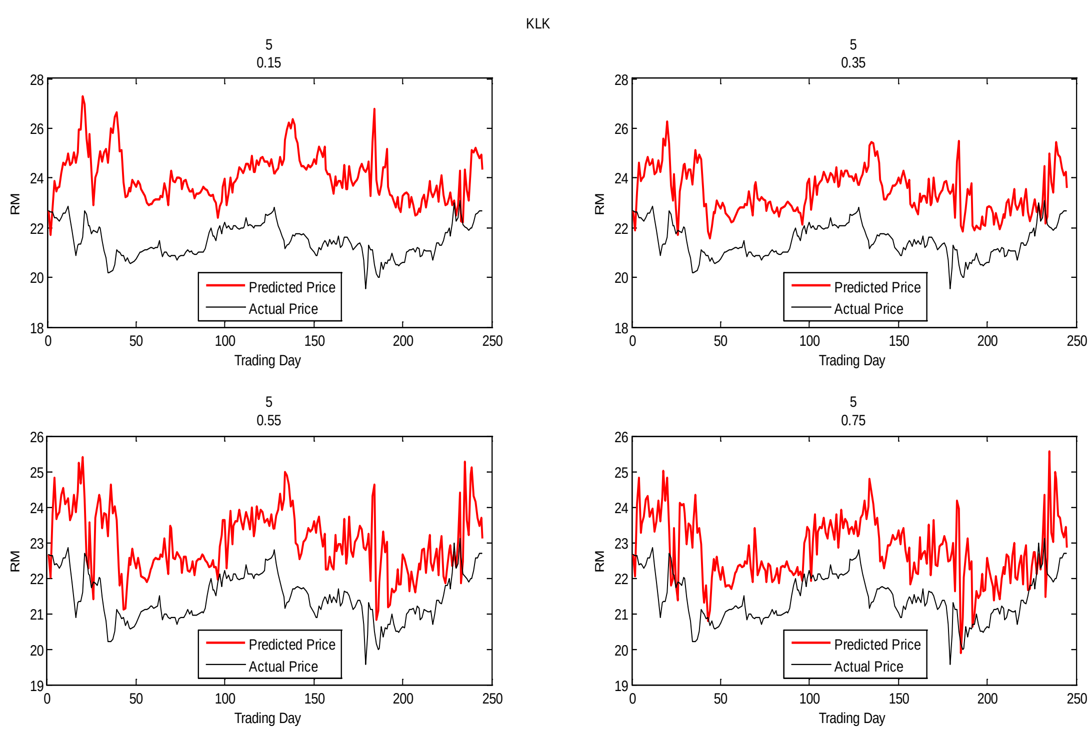


Figure 4 : Convergence of actual and forecasted share prices of KLK

Table 8 : Different types of errors produced at various learning rates - KLK

Learning rates	RMSE	MAE	MAPE (%)	Hit Rate (%)
0.15	2.82	2.60	12.14	48.97
0.35	2.16	1.98	9.21	48.15
0.55	1.78	1.58	7.35	46.50
0.75	1.57	1.36	6.32	47.74

The RMSE declines when the learning rate increases from 15% to 75% by 44% approximately. Similarly the MAE and MAPE decline by 47.69% and 47.94% respectively. The hit rate declines when the learning rate increases by 2.51%. In absolute terms it declines from 48.97% to 47.74%. These higher error levels reveal the poor convergence of actual and predicted share prices of KLK.

XIII. Conclusion

In this article we applied ADALINE neural network to predict the selected share prices of companies listed in Malaysian stock market. The ADALINE neural network predicts the trends well for all the four companies. The convergence of actual and predicted prices is excellent at higher learning rates in three companies. KLK company's graph shows poor fitting. The predicted prices closely converge with the actual prices with negligible gap at the higher learning rates. At lower learning rates the convergence is poor for all four companies. Our finding will be useful for fund managers to predict the share prices which will facilitate not only in decision making, controlling, and hedging but also in selection of shares for constructing share portfolios.

References Références Referencias

1. Aleksander, I., & Morton, H. (1995), An Introduction to Neural Computing, International Thompson computer press, London.
2. Banz, R. W. (1981), The Relationship between Return and Market Value of Common Stocks, Journal of Financial Economics, 9 (1), 3-18
3. Buscema, M., & Sacco, P., (2000), feed forward networks in financial predictions: The future that modifies the present. Expert Systems, 17, 149-170
4. De Faria E. L., Albuquerque, M. P., Gonzalez, J. L., Cavalcante, J. T., & Albuquerque, M. P. (2009), Predicting the Brazilian stock market through neural networks and adaptive exponential smoothing methods, Expert Systems with Applications.
5. Govindarajan, M., & Chandrasekaran, R. M., (2007), Classifier Based Text Mining for Neural Network. World Academy of Science, Engineering and Technology , 27, 200-205
6. Grønholdt, L., & Martensen, A., (2005), Analysing Customer Satisfaction Data: A Comparison of Regression and Artificial Neural Networks, International Journal of Marketing Research, 47, 121-130
7. Hecht-Nielsen, R. (1989). Theory of the Back propagation Neural Network, International Joint Conference IEEE 1, pp. 593-605 Washington, DC, USA: IEEE.
8. Hirt, G. A., & Block, S. B. (1996), Fundamentals of Investment Management Irwin, New York: McGraw-Hill.
9. Janssen, C., Langager, C., & Murphy, C. (2011), Technical Analysis: The Basic Assumptions.
10. Jones, C. P. (2007). Investments: Analysis and Management. Hoboken, N.J.: John Wiley & Sons, Inc.
11. Kaastra, I., & Boyd, M. (1996), Designing a neural network for forecasting financial and economic time series Neuro computing, 10, 215-236
12. Lin, C. T., & Yeh, H. Y. (2009), Empirical of the Taiwan Stock Index Option Price Forecasting Model - Applied Artificial Neural Network, Applied Economics , 41
13. Matilla-Garcia, M., & Arguello, C. (2005) A Hybrid Approach Based on Neural Networks and Genetic Algorithms to the Study of Profitability in the Spanish Stock Market. Applied Economics Letters, 12, 303-308
14. Mitchell, D., & Pavur, R. (2002) Using modular neural networks for business decisions Management Decision, 40 (1), 58-63
15. Refenes, A. N., & Francis, G. (1993), Currency exchange rate prediction and neural network design strategies. Neural Computing and Applications Journal, 1, 46-58
16. Remus, W., & O'Connor, M. (2001), Neural networks for time-series forecasting - A handbook for researchers and practitioners, (pp. 245-256). Norwell, MA: Kluwer Academic Publishers.
17. Rude, A (2010), Using Artificial Neural Networks to Forecast Financial Time Series Norwegian University of Science and Technology.
18. Sapena, O., Botti, V., & Argente, E., (2003), Application of neural Networks to Stock Prediction in "Pool" Companies Applied Artificial Intelligence, 17, 661-673
19. Thomaidis, N. S., Tzastoudis, V. S., & Dounias, G. D. (2007), A Comparison of Neural Network Model Selection Strategies for the Pricing of S&P 500 Stock Index Options International Journal on Artificial Intelligence Tools, 16, 1093-1113
20. Yoon, Y., & Swales, G. (1991), Predicting stock price performance: A neural network approach. Proc. 24th Annual International Conference of Systems Sciences, (pp. 156-162). Chicago.

MATLAB Program

```
close all
clear all
clc

load data4                      %  Load data
for m = 1:4                     %  Data from 1 to 4
   p=5;                         % Number of neurons
   alp=[.15 .35 .55 .75];       % Learning Rates

   for kk=1:4
      x=data(22-p:end,m);          % Take the share price
      xr=price2ret(x);             % Convert share price to geometric returns
      xr10=xr(1:ceil(length(xr)/2),:);        % Take returns of 2010
      xr11=xr(ceil(length(xr)/2)+1:end,:);    % Take returns of 2011
      t1=xr10(1:end);                         % 2010 target returns
      t2=xr11;                                % 2011 target returns
      X=convmtx(xr,p);                        % Convolution matrix for taking a few returns
      X=X(p:(end-p+1),:);                     % Take only the rows with full data
      rand('state',10)                        % Fix the random numbers
      w=2*(rand(1,p)-0.5);                    % Generate 5 random numbers for weights
      b=rand(1,1);                            % Generate a random number for bias

   %% Training algorithm

   for i= 1:length(t1);
      y(i) = b+w*X(i,:)';                  % Find the weighted total
      err(i) = t1(i) -y(i);                % Find deviation from the target
      w=w+alp(kk)*err(i)*X(i,:);           % Update the old weight
      b=b+alp(kk)*err(i);                  % Update the bias
      ww(i,:) = w;                         % Store the new weights
   end

    %% Testing or prediction
   k=1;                                    % Counter initialisation
   nn=length(xr10);                        % 2010 Number of days
   spa=x(nn+1);                            % Take share prices of 2010
   spaa=x(nn+1+p);                         % Take share prices of 2010 plus another 5 prices

   for j=nn+1:length(X)
      y1=b+w*X(j,:)';                      % Compute a predicted return
      yy(k)=y1;                            % Store it in yy
      err=t2(k)-y1;                        % Find the deviation the predicted and actual price
      w=w+alp(kk)*err*X(j,:);              % Update the weight
      b=b+alp (kk)*err;                    % Update the bias
      k=k+1;
   end
   sp=ret2price(yy,spaa);                  % Predicted share price

   p11=x(length(t1)+1+p:end);              % Actual share price
   e=p11;
   f=sp(1:end-1)';
   p12=(sp(:,end-1))';
   err1=(mean((p12-p11).^2)^0.5);
   q=(e - f);                              % Errors
   qq=(e - f).^2 ;                         % Squared Error
   qqq=mean((e - f).^2)  ;                 % Mean Squared Error
   rmse = sqrt(mean((e - f).^2));          % Root Mean Squared Error
```

```
%% Graph codes

        subplot(2,2,kk);
        bh=plot(sp,'r', 'linewidth', 1.5);
        hold on
        bg=plot(p11,'k');
        tt=num2str(alp(kk));
        title([textdata(m), 'Learning Rate',tt]);
        xlabel ('Trading Day')
        ylabel ('RM')
        legend('Predicted Price','Actual Price',3)
        axis([0 250 -inf inf])
    end
    figure
end
```