

Improving Artificial Neural Network Forecasts with Kalman Filtering

J Cheeneebash*

*Department of Mathematics
Faculty of Science
University of Mauritius
jayrani@uom.ac.mu*

G Galanis

*Section of Mathematics,
Naval Academy of Greece,
Hatzikyriakion 185 39, Piraeus, Greece
ggalanis@math.uoa.gr*

A Gopaul

*Department of Mathematics
Faculty of Science
University of Mauritius
a.gopaul@uom.ac.mu*

M Bhuruth

*Department of Mathematics
Faculty of Science
University of Mauritius
mbhuruth@uom.ac.mu*

Paper Accepted on 14 January 2008

Abstract

In this paper, we examine the use of the artificial neural network method as a forecasting technique in financial time series and the application of a Kalman filter algorithm to improve the accuracy of the model. Forecasting accuracy criteria are used to compare the two models over different set of data from different companies over a period of 750 trading days. In all the cases we find that the Kalman filter algorithm significantly adds value to the forecasting process.

Keywords: Artificial Neural Networks, Kalman filter, Stock prices, Forecasting, Back propagation

*** To whom correspondence should be addressed**

1. INTRODUCTION

In this paper, we show that the post-processing of forecasts produced by an artificial neural networks (ANN) model using a Kalman filter leads to significant improvements in forecasting accuracy. Various attempts have been made in using ANN models for forecasting financial time series. The growing interest in this nonlinear method can be explained by the poor performance of classical methods for financial forecasting. In fact, the uncertainty associated with the pricing of financial derivatives was first dealt by the random walk hypothesis, also known as the efficient market model (Panda et al. (2003)). Later linear regression models such as ARIMA and VAR models were developed but their out-sample results were not better than the random walk model (Panda et al. (2003)). Since the linear models could not be used to accurately fit the financial market models, more research work was directed towards the development of nonlinear models such as the threshold model, the bilinear model and the artificial neural network model (Franses et al. (1998)). Kuan et al. (1995) showed that ANN model gave better predictions than the random walk model.

The ANN model is often classified as a nonlinear, non parametric and data driven modeling method. The input data is fully used and it determines the structure and parameters of the model. Neural networks are universal approximators capable of simulating any continuous function and they usually require a few a priori assumptions about the model under study. Other applications of ANN have been discussed for airline modeling and river flow forecasting (Atiya et al. (1999), Faraway et al. (1998)).

The performance of the ANN model depends upon the type of data, the choice a suitable model and the numerical method used for fitting the model and computing the forecasts. Faraway et al. (1998) showed that linear models give better forecasts than ANN for linear data. Other drawbacks of ANN are excessive training times, danger of over-fitting and the large number of parameters required for training. Hence the determination of an appropriate network is often a trial and error issue.

Our aim here is to add value to the performance of ANN by using a post-conditioner. Some recent works, (Galanis et al. (2006), Emmanouil et al. (2006), Louka et al. (2005)) have shown that the use of the Kalman filter produced high improvement in the accuracy of meteorological and wind power predictions. The novelty of this paper is the application of the Kalman filter on the predictions obtained by the ANN model for forecasting stock prices. The outline of our work is as follows. In sections 2 and 3, we briefly describe the ANN method and the Kalman filter algorithm. In section 4, we explain our methodology and finally, based on our results, we draw some concluding remarks.

2. ARTIFICIAL NEURAL NETWORKS

The ANN method imitates the way by which the brain processes information. Given an input vector $p = (p_1, p_2, \dots, p_n)^T$, the network produces an output vector $r = (\hat{r}_1, \hat{r}_2, \dots, \hat{r}_m)^T$ where n indicates the number of inputs and m the number of output units. A neural network is typically organized into several layers of nodes. The first layer is the input layer, the number of nodes corresponding to the number of variables, and the last layer is the output. The input and output layer can be separated by one or more hidden layers. The nodes in adjacent layers are fully connected. Each neuron receives information from the preceding layer and transmits to the following layer only. The neuron then performs a weighted summation of its inputs; if the sum passes a threshold, the neuron transmits, otherwise it

remains inactive. An example of a fully connected ANN model with one hidden layer is shown in Fig.1,

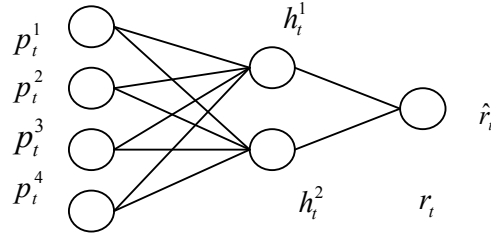


Fig. 1: An artificial neural network model.

where $p_t^i, i=1,2,3,4$, are the inputs at time t , $h_t^j, j=1,2$, are the hidden outputs. The variables r_t and \hat{r}_t are the actual and ANN model output respectively. The vector p represents the input to the ANN model where p_t^i is the level of activity at the i^{th} input. Associated with the vector is a series of weight vectors $W_j = (w_{1j}, w_{2j}, \dots, w_{nj})$ so that w_{ij} represents the strength of the connection between the input p_t^i and the unit b_j . There may also be the input bias \mathcal{G}_j modulated with the weight w_{0j} associated with the inputs. The total input of the node b_j is the dot product between vectors p and w_j less the weighted bias. It is then passed through a nonlinear activation function to produce the output value of processing unit b_j , which is defined as

$$b_j = f\left(\sum_{i=1}^n x^i w_{ij} - w_{0j} \mathcal{G}_j\right) = f(X_j).$$

The activation function introduces a degree of nonlinearity to the model and prevents the output from reaching very large values that can paralyze ANN models and inhibit training. In this paper we choose $f(x) = (e^x - e^{-x}) / (e^x + e^{-x})$ as the activation function as these have been extensively and successfully used in previous studies. The modeling process begins by assigning random values to the weights. The output value of the processing unit is passed on to the output layer. If the output is optimal, the process is halted. Else the weights are adjusted by using an appropriate algorithm, which we choose as the back propagation algorithm for our work. The process continues until an optimal solution is found. The output error, which is the difference between the actual value and the ANN model output, is also known as the optimisation error.

3. KALMAN FILTER

In this section the Kalman filter algorithm used for the improvement of neural network forecasts is described. The main idea is the estimation of the bias of the neural network model as a polynomial function of the model direct output. More precisely, if r_t denotes the direct output of the model at time t and y_t the bias of this forecast, we realize y_t by means of r_t as a 3-rd order polynomial (Galanis et al. (2006)):

$$y_t = x_{0,t} + x_{1,t}r_t + x_{2,t}r_t^2 + x_{3,t}r_t^3 + v_t.$$

The coefficients $(x_{i,t})_{i=0,\dots,3}$ are the parameters that have to be estimated by the filter and v_t the Gaussian non-systematic error in the previous procedure.

In this way, the state vector of the procedure is formed by the coefficients $(x_{i,t})$, namely $x_t = [x_{0,t} \ x_{1,t} \ x_{2,t} \ x_{3,t}]^T$, the scalar observation parameter is the bias y_t and the observation matrix $H_t = [1 \ r_t \ r_t^2 \ r_t^3]$. We choose the identity matrix as the system matrix. Therefore, the system and observation equations take the following form:

$$x_t = x_{t-1} + w_t, \quad y_t = H_t \cdot x_t + v_t.$$

Kalman filter gives a method for the recursive estimation of the unknown state x_t based on all observation values y up to time t . The evolution of the filter from time $t-1$ to t is described by the following equations:

$$\begin{aligned} x_t &= x_{t-1} + K_t \cdot (y_t - H_t \cdot x_{t-1}), \\ K_t &= (P_{t-1} + W_t) \cdot H_t^T \cdot (H_t \cdot (P_{t-1} + W_t) \cdot H_t^T + V_t)^{-1}, \\ P_t &= (I - K_t \cdot H_t) \cdot (P_{t-1} + W_t). \end{aligned}$$

In the above algorithm, K_t is the Kalman gain that arranges how easily the filter adjusts to possible new conditions or alternations of the type of data. On the other hand, P_t denotes the covariance matrix of the state vector x_t .

Kalman filters of the above type have been used in several different frameworks including meteorological and wind power applications with excellent results (Galanis et al. (2006), Emmanouil et al. (2006), Louka et al. (2005)). The proposed algorithm permits the estimation of non-linear time series since it encapsulates in the classical Kalman filter procedure non-linear polynomial functions.

4. METHODOLOGY

We consider the stock prices of four leading companies operating in Mauritius, which are namely, Mauritius Commercial Bank Ltd (MCB), Ireland Blyth Ltd (IBL), Rogers and Co. Ltd (ROGERS) and the State Bank of Mauritius Ltd (SBM). These daily stock prices, which have been collected during the period September 17, 2001 to September 16, 2004, amount to a total of 750 observations as shown in Fig. 2.

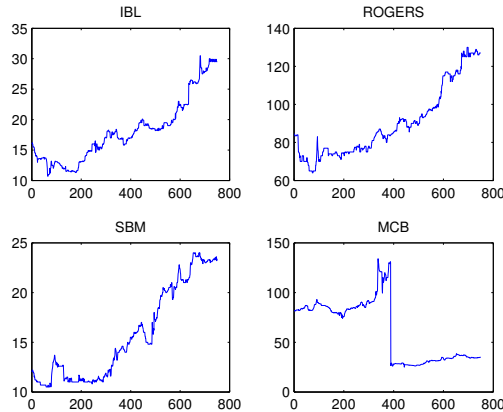


Fig. 2 Plots of the data set for IBL, ROGERS, SBM and MCB.

Denoting sp_t to be the stock price at time t , we define the price log differences as $s_t = \log sp_t - \log sp_{t-1}$. We consider the first 500 observations which are used for the training set as the in sample set and the remaining 250 observations as the out sample set. The latter is used for testing the accuracy of the forecasted prices. We present the statistics for our data set in Table 1.

Table 1: Summary statistics for data set: Log first difference.

	MCB	IBL	ROGERS	SBM
Mean	-0.0011	8.2E-4	5.4E-4	8.9E-4
Standard Deviation	0.0601	0.0155	0.0107	0.0128
Skewness	-24.6931	0.0646	-0.3955	2.1205
Kurtosis	653.8821	23.7965	14.4222	30.3580
Maximum	0.1372	0.1369	0.0651	0.1386
Minimum	-1.5904	-0.1223	-0.0621	-0.0780

The parameters used for our neural network are a single hidden layer with three nodes and a linear transfer function for the output and the number of iterations and epochs are chosen as 2000 and 5000 respectively. After carrying out the simulations with the Neural Network toolbox in MATLAB, we have based the selection of the best training sets for each company on the smallest root mean sum square error (RMSE) criteria. These errors are minimized through the adjustments of weights. The training set is used to develop the model and the test set measures how well the model interpolates over the training set. Training normally stops when the test set error reaches its lowest point. This stopping rule reduces the likelihood of over fitting. The same model is then applied to the test set. Using the forecasted values obtained by the ANN model, we then apply the Kalman filter algorithm to these values. We measure the accuracy of the forecasted values obtained from ANN and from our new method through RMSE, MAE (Mean absolute error), and MAPE (Mean absolute percentage error). Their formulas are given below

- RMSE: $\sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}$.
- MAE: $\frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$.
- MAPE: $\frac{1}{N} \sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{|y_i|}$.

5. OUT-OF SAMPLE FORECASTING ACCURACY RESULTS

The forecasting accuracy statistics are shown in Table 2 for the four different companies. A general observation is that the ANN model gives a satisfactory forecast but the application of the Kalman Filter to these predicted values improves the result significantly in all the cases. This can be seen from the graphs in Fig. 3 – Fig. 6 where the Kalman filtered values are shown to be closer to the actual values.

Table 2: Bias for the forecasted out-sample stock prices by using the ANN model and with Kalman Filter.

	MCB		IBL	
	ANN	Kalman Filter	ANN	Kalman Filter
RMSE	7.91	0.91	1.87	0.81
MAE	7.45	0.56	1.43	0.44
MAPE	0.22	0.02	0.05	0.02
	Rogers		SBM	
	ANN	Kalman Filter	ANN	Kalman Filter
RMSE	7.80	2.34	46.81	0.4
MAE	6.66	1.59	33.61	0.26
MAPE	0.05	0.01	1.47	0.01

From Table 2, we find that for the four companies, the RMSE, MAE and MAPE are much smaller for the filtered forecasted values. We also note that in the case of SBM, even if we do have a good set of predicted prices, we can obtain a good forecast by using the Kalman filter algorithm as shown in Fig. 5. The RMSE for SBM falls from 46.81 to 0.4.

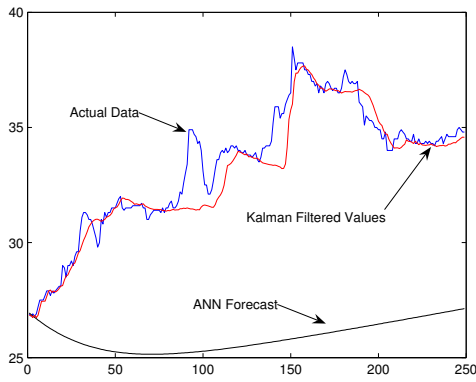


Fig. 3: Forecasted values for MCB out-sample set.

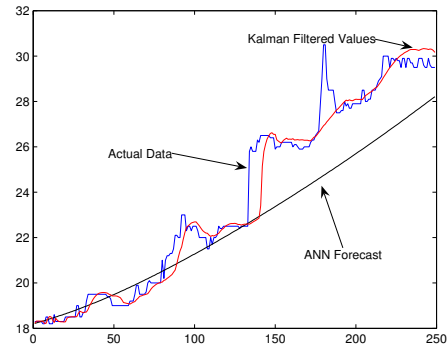


Fig. 4: Forecasted values for IBL out-sample set.

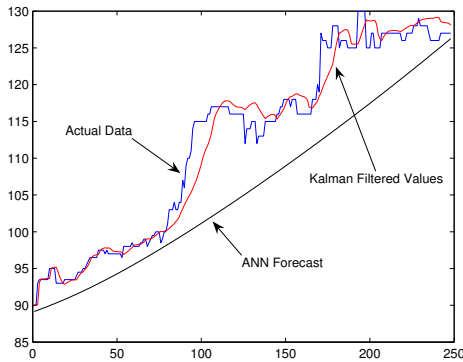


Fig. 5: Forecasted values for ROGERS out-sample set.

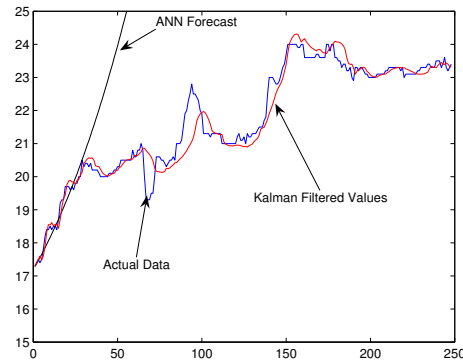


Fig. 6: Forecasted values for SBM out-sample set.

6. CONCLUSION

In this paper we have forecasted the stock prices of four companies based in Mauritius by using the ANN model. We have then applied the Kalman filter algorithm to these predicted prices and we find that this process has significantly improved their accuracy as illustrated by our simulations.

7. REFERENCES

Atiya, AF, El-Shoura, SM, Shaheen, SI, & El-Sherif, MS 1999, 'A comparison between neural-network forecasting techniques – case study: river flow forecasting', *IEEE Transactions on Neural Networks*, Vol. 10, No. 2, pp. 402-409.

Galanis, G, Louka, P, Katsafados, P, Kallos, G, & Pytharoulis, I 2006, 'Applications of non-linear Kalman filters to numerical weather predictions', *Annales Geophysicae*, Vol. 24, pp. 2451-2460.

Emmanouil, G, Galanis, G, & Kallos, G 2006, 'Statistical methods for the prediction of night time cooling and minimum temperature', *Meteorol. Appl.* Vol. 13, pp. 1–11.

Faraway J. & Chatfield C 1998, 'Time series forecasting with neural networks: a comparative study using the airline data', *Appl. Statist.* Vol. 47, Part 2, pp. 231-250.

Franses, PH & van Griensven K 1998, 'Forecasting exchange rates using neural networks for technical trading rules', *Studies in nonlinear dynamics and econometrics*, Vol. 2, No. 4, pp. 109-114.

Kuan CM & Liu T 1995, 'Forecasting exchange rates using feedforward and recurrent neural networks', *Journal of Applied Econometrics*, Vol. 10, pp. 347-364.

Louka P, Galanis G, Sibert, N & Kariniotakis G 2005, 'Improvements in wind speed forecasts for wind power prediction purposes using Kalman filtering', *Proceedings of the 2005 Conference on Computational and Mathematical Methods on Science and Engineering*.

Panda C & Narasimhan V 2003, 'Forecasting daily foreign exchange rate in India with artificial neural network', *The Singapore Economic Review*, Vol. 48, No. 2, pp. 181-199.