PERFORMANCE STATISTICS FOR RECURRENT NEURAL NETWORKS ON STOCK PRICES FORECASTING

Suwat Pattamavorakun*[a] and Suwarin Pattamavorakun[b]

[a]Business Administration Faculty, [b]Science Faculty
Rajamangala University of Technology, Thanyaburi
Klong Six, Pathumthani, 12110 Thailand

ABSTRACT

Forecasting the behavior of complex systems has been a broad application domain for neural networks. In particular the recurrent neural networks has been widely studied. to the extend this study with specific emphasis on the topic of stock prices forecasting. Since how to accurately forecast stock prices is become more important and challenging the marketers are competition increase. In order to evaluate the performance of recurrent neural networks. the three performance statistics are discussed. In terms of Efficiency Index. Mean Absolute Deviation and Maximum Relative Error. an experiments carried out by means of the use of stock prices and volumes for the fully recurrent neural network and two partially recurrent neural networks training with an online algorithm. The result found that mostly Efficiency Index and Mean Absolute Deviation are suitable used to indicate the overall recurrent neural network performance.

KEYWORDS: forecasting. recurrent neural networks. performance statistics. Efficiency Index. Mean Absolute Deviation. Maximum Relative Error

1. INTRODUCTION

Predicting the future has always been one of humanity's desires. Time series measurements are the means for us to characterize and understand a system and to predict its future behavior. If there are underlying deterministic equations. the system's behavior can be determined by solving the equations with the given initial condition. In time series prediction these equations and the initial conditions are either unknown or only partially known. so one needs to find out the rules that given the system evolution and the actual current state of the system. The governing rules may be inferred from the regularities in past time series measurements.

Artificial neural networks (ANNs) have been shown to be successful as a forecasting tool in a variety of ways: forecasting that some event will or will not occur. forecasting the time at which an event will occur. or forecasting the level of some event outcome. To predict with an acceptable level of accuracy. an ANN has to be trained with a sizable set of examples of past patterns together with known future outcomes and it will be able to generalize and extrapolate from new patterns to predict future outcomes. Generally. the training set comes from historical data that has been collected over a given time period. Many applications have used ANNs for forecasting including financial time series forecasting. foreign exchange trading. stock selection. portfolio management. loan management. sensor data fusion and fault prediction. and weather forecasting.

Backpropagation (BP) networks [1] are a type of ANN closely related to statistical methods. They are suitable for forecasting applications [2. 3]. While ANNs provide a great deal of promise. they also embody much uncertainty (recurrent neural networks (RNNs) are also in this case). Researchers to date are still not certain about the effect of key factors on forecasting performance of ANNs. Note that every RNNs model has limits on accuracy it can achieve for real problems. Although there can be many performance measures for an ANN forecaster like the modeling time and training time. the

* Corresponding author :Tel: 02-549-4828 Fax: 02-549-3242. Tel 02-549-4194 Fax: 02-549-4195

E-mail: suwat@rmut.ac.th. suwarin@rmut.ac.th

ultimate and the most important measure of performance is the prediction accuracy it can achieve beyond the training data.

This paper examines the performance measures of RNNs, therefore the comparison will be made empirically using actual data, financial forecasting time series. So, this work describes selected training algorithms for popular RNN structures and determines their performance in terms of accuracy (indicated by three performance statistis) and speed for the stock prices of important companies listed on the Stock Exchange of Thailand. For the financial markets differ a lot from physical systems. The stock market is an important institution serving as a channel that transforms saving into real capital formation. It will simulate economic growth and also increases the Gross National Product (GNP).

2. SELECTED ARTIFICIAL NEURAL NETWORK

There are two major types of ANNs, namely feedforward neural networks and recurrent neural networks. The feedforward or strictly one-way flows an information through this network, meaning from input to output. Of course the other models are more complicated, with all sorts of recurrent connections producing feedback, and capable of discriminating temporal patterns. But with their characteristic, feedforward neural networks are static mapping and difficult to represent a nonlinear dynamic system.

2.1 Recurrent Neural Networks

Recurrent neural networks (RNNs) are those neural networks in which the output from the output layer are fed back to a set of input units. This is in contrast to feed-forward networks, where the outputs are connected only to the inputs of units in subsequent layers. RNNs are able to store information about time and have a dynamic system possessing state. Therefore they are particularly suitable for forecasting applications. They have been used with considerable success for predicting several types of time series. In recent years, the use of recurrent neural networks for time series prediction has gained popularity and can now compete with among known methods. Nowadays, RNNs are very interesting for problems involving a single time series with related information. Related information supplements the time series data, and may be necessary to ensure correct predictions. In training, a recurrent neural network acts as a filter, compressing historical information to best represent the past history of a time series. This is most effective when the important historical information is not in the too distant past.

- Fully Recurrent Neural Networks (FRNNs)

FRNNs can take an in-determinate amount of time. FRNNs are used primarily for optimization problems and as associative memories. The connections in the FRNNs are mainly feedforward but include a set of feedback connections in which any node in output layer may be connected to any other nodes as illustrated in Figure 1. FRNNs can be viewed as nonlinear dynamical systems because their connections can represent the dynamical complexity such as these three main dynamical behaviors (properties): their behavior in the limit reaches a steady state (fixed point), an oscillation (limit cycle), and a periodic instability (choas). These behaviors can be exploited for a useful purpose in neural networks.

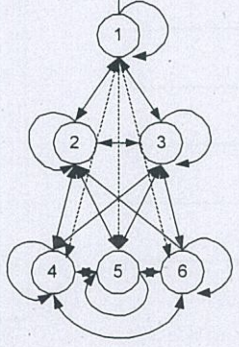


Figure 1. A fully connected recurrent network.

- Narendra & Parthasarathy (N&P) and Frasconi-Gori-Soda (FGS) networks

Narendra & Parthasarathy (N&P) and Frasconi-Gori-Soda (FGS) locally recurrent have been applied to classification tasks. Especially, classification of natural language sentences as grammatical or ungrammatical. N&P network is a partially connection recurrent network with feedback connections linked from each node of output layer to all hidden nodes as illustrated in Figure 2. In this network, output values are feedback to the network as inputs and act as short-term memory [4]. FGS locally recurrent network is a multilayer perception (MLPs) with local feedback (self-loop) around each hidden node as illustrated in Figure 3. A potential advantage of this network is that it can adapt its internal representation [5].

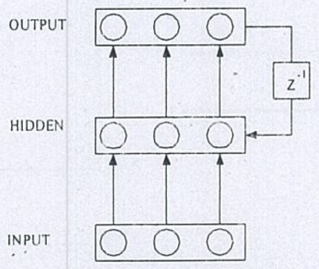


Figure 2. A Narendra & Parthasarathy recurrent network.

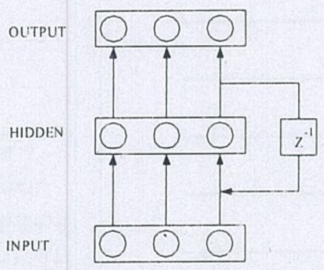


Figure 3. A Frasconi-Gori-Soda locally recurrent network

2.2 Training Algorithm

Learning is a fundamental aspect of neural networks and a major feature that makes the natural approach so attractive for applications and have from the beginning been an elusive goal for artificial intelligence [6]. Due to algorithms which based on the steepest descent optimization, the speed of convergence is relatively low [7], therefore in this study the non-gradient method was used instead of exact-gradient.

- The Error Back Propagation and Exponentially Weighted Least Squares [8] which is considered an online algorithm was used. This algorithm is derived using an algebraic method instead of a gradient approach. Since the function *f* is monotonic and is continuously differentiable, there is an inverse function of the activation function *f*, and instead of the gradient descent technique, the following criterion is minimized to derive a learning algorithm.

$$e_i(k) = \begin{cases} d_i(k) - y_i(k) & i \in T(k) \\ 0 & otherwise \end{cases} \tag{1}$$

$$\begin{aligned} e_i(k) &= f^{-1}(d_i) - f^{-1}(y_i) \\ &= f^{-1}(d_i) - \mathbf{W}^T y(k) \end{aligned} \tag{2}$$

The minimization of Eq.(2) can be done by appropriately choosing **W**. And it also requires to find the fictitious target signals for both output and hidden nodes, d_k and y_{hj}, respectively which can be determined by Error Back Propagation (EBP) algorithm. This is in contrast to the Back Propagation (BP) method, where the BP is essentially the sensitivity of the error with respect to the weight parameters. Figure 4. represents the fully recurrent network architecture. For $y_i(k) \in O$, it is considered as $y_o \in$ output nodes, $y_h \in$ hidden nodes, and $u \in$ external input. For this algorithm, this is defined $\mathbf{W_o}$ is the connecting weight to k+1 from k, and $\mathbf{W_H}$ is the connecting weight to k from k-1 time step.

From $\quad y_i(k+1) = f_i(s_i(k))$

$$s_o(k) = W_{oo}^T y_o(k) + W_{oh}^T y_h(k) + W_{oi}^T u(k)$$

$$y_o(k+1) = f_o(s_o(k)) \qquad (3)$$

$$s_h(k) = W_{ho}^T y_o(k) + W_{hh}^T y_h(k) + W_{hi}^T u(k) \qquad y_h(k+1) = f_h(s_h(k)) \qquad (4a)$$

where o, h, i are the output nodes, hidden nodes, and external input nodes.
Eq.(4a) is also represented by

$$s_h(k-1) = W_{ho}^T y_o(k-1) + W_{hh}^T y_h(k-1) + W_{hi}^T u(k-1) \qquad y_h(k) = f_h(s_h(k-1)) \qquad (4b)$$

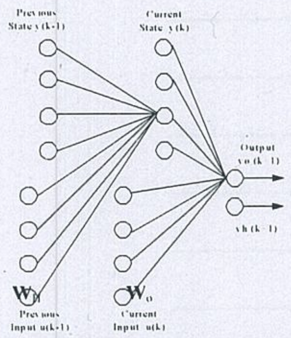


Figure 4. A standard fully connected recurrent neural network computational scheme.

An EBP method for obtaining a fictitious target signal of output of the hidden nodes is as follows. First, assume

$$d(k+1) = y(k+1) \qquad (5)$$

We define,

$$f^{-1}(d(k+1)) = W^T(y(k) + \Delta y(k)) \qquad (6)$$

or equivalently

$$\mathbf{W}^T \Delta y(k) = f^{-1}(d(k+1)) - \mathbf{W}^T y(k) \qquad (7)$$

From Eq.(2), therefore

$$e(k) = \mathbf{W}^T \Delta y(k) \qquad (8)$$

where

$$y(k) = [y_o(k),\ y_h(k),\ u(k)]^T \qquad (9)$$

$$\Delta y(k) = [\Delta y_o(k),\ \Delta y_h(k),\ \Delta u(k)]^T \qquad (10)$$

For finding a fictitious target signal y_{ht}, we first calculate Eq.(8) and substitute $W^T \equiv W_o = \left(w_{oo}, w_{oh}, w_{oi}\right)^T$. A desired variation of the output of hidden nodes, denoted by $\Delta y_h(k)$, is determined as

$$\Delta y(k) = \mathbf{W_o}\left(\mathbf{W_o^T W_o}\right)^{-1} e(k) \tag{11}$$

In this method, the variations $\Delta y_o(k)$ and $\Delta u(k)$ are ignored. Then a fictitious target signal y_{ht} is obtained as

$$y'_{ht}(k) = y_h(k) + \Delta y_h(k) \tag{12}$$

$$y_{ht}(k) = \mathbf{H} y'_{ht}(k) \tag{13}$$

The resulting vector y_{ht} has the role of a temporary target function for y_h. This is accomplished by multiplying an appropriate scalar for a vector y'_{ht} in order to prevent the saturation problem. The symbol scalar **H** represents the operator for a vector y'_{ht} to be included in the range space of the function. In this work, the range space H is in the interval [0.05, 0.95]. After that, we determine the weight parameters solved by an Exponentially Weighted Least Squares (EWLS) method.
An EWLS performance for the output of a hidden unit is

$$E_{ht} = \frac{1}{2}\sum_{i=0}^{k-1} \beta^i e_h(k-i)^T e_h(k-i) \tag{14}$$

where β $(0 < \beta < 1)$ is the exponential weight.

$$e_h(k) = f^{-1}(y_{ht}(k)) - \mathbf{W_H^T} y(k-1) \tag{15}$$

where $\mathbf{W_H^T} = \left(w_{ho}, w_{hh}, w_{hi}\right)^T$

$$y(k-1) = \left[y_o(k-1),\ y_h(k-1),\ u(k-1)\right]$$

The weight is updated by

$$\Delta \mathbf{W_H}(k-1) = \frac{P_h(k-1)y(k-1)}{\beta + y(k-1)^T P_h(k-1) y(k-1)} e_h(k) \tag{16}$$

$$P_h(k) = \frac{1}{\beta}\left(P_h(k-1) - \frac{P_h(k-1)y(k-1)y(k-1)^T P_h(k-1)}{\beta + y(k-1)^T P_h(k-1)y(k-1)}\right) \tag{17}$$

Then.
$$\mathbf{W_H}(k) = \mathbf{W_H}(k-1) + \Delta\mathbf{W_H}(k-1) \tag{18}$$

After modification of the hidden weight matrix, the new output value of hidden units is changed to

$$\hat{y}_h(k) = \mathbf{W_H}(k)^T y(k-1) \tag{19}$$

in order to obtain an exact the value of output node at time k+1. Therefore,

$$\hat{y}(k) = \left(y_o(k),\ \hat{y}_h(k),\ u(k)\right) \tag{20}$$

On the other hand, the output of a neural network is performed as:

$$\Delta W_o(k-1)=\frac{P_o(k-1)\hat{y}(k)}{\beta+\hat{y}(k)^T P_o(k-1)\hat{y}(k)}e(k) \tag{21}$$

And

$$P_o(k)=\frac{1}{\beta}\left(P_o(k-1)-\frac{P_o(k-1)\hat{y}(k)\hat{y}(k)^T P_o(k-1)}{\beta+\hat{y}(k)^T P_o(k-1)\hat{y}(k)}\right) \tag{22}$$

3. PERFORMANCE MEASURES

A suitable measure of accuracy for a given problem is not universally accepted by the forecasting academicians and practitioners. An accuracy measure is often defined in terms of the forecasting error which is the difference between the actual (desired) and the predicted value. There are a number of measures of accuracy in the forecasting literature and each has advantages and limitations [9]. The most frequently used are

- the mean absolute deviation (MAD) = $\frac{\sum|e_t|}{N}$;
- the sum of squared error (SSE) = $\sum(e_t)^2$;
- the mean squared error (MSE) = $\frac{\sum|e_t|^2}{N}$;
- the root mean-squared error (RMSE) = $\sqrt{\text{MSE}}$;
- the mean absolute percentage error (MAPE) = $\frac{1}{N}\sum\left|\frac{e_t}{y_t}\right|(100)$.

where e_t is the individual forecast error; y_t is the actual value; and N is the number of error terms.

In addition to the above, other accuracy measures are also found in the literature. For example, the mean error (ME) is used by Gorr et al. [10], Theil's U-statistic is tried by Hann and Steurer[11], and the median absolute percentage error (MdAPE) and the geometric mean relative absolute error (GMRAE) are used by Foster et al. [12]. Cottrell et al. [13] adopt the residual variance and Akaike information criterion and Bayesian information criterion (BIC). Because of the limitations associated with each individual measure, one may use multiple performance measures in a particular problem. However, one method judged to be the best along one dimension is not necessarily the best in terms of other dimensions.

It is important to note that the first four of the above frequently used performance measures are absolute measures and are of limited value when used to compare different time series. MSE is frequently used accuracy measure in the literature. However, the merit of using the MSE is much debated in evaluating the relative accuracy of forecasting methods across different data sets [14]. Furthermore, the MSE defined above may not be appropriate for ANN model building with training sample since it ignores the important information about the number of parameters the model has to estimate. From the point of view of statistics, as the number of estimated parameters (are weights) in the model goes up, the degrees of freedom for the overall model goes down, thus raising the possibility of overfitting in the training sample. An improved definition of MSE for the training part is the total sum of squared errors divided by the degrees of freedom, which is the number of observations minus the number of forecast values are weights and node biases in an ANN model.

4. DATA USED

When training the network by a chosen algorithm, it should be done step by step until the system error has reached an acceptable criterion. The stopping point (ε), determination of the optimal network structure, is based upon the relative difference of the sum of squared error (SE) between the previous epoch and current iteration. The smaller the number of ε, the better the neural network performs, but it may also take longer for the network to train. Therefore it was chosen to be 0.05.

$$\left|\frac{SE(t+1)-SE(t)}{SE(t)}\right| \le \varepsilon \qquad (23)$$

where ε is a constant that indicates the acceptable level of the algorithm and $SE(t)$ denotes the value of SE at iteration t. The maximum number of iterations for the training process was limited to 20,000. All parameter values obtained in the training phase cannot be changed in validation phase

In order to evaluate the performance of the recurrent neural network, the following performance statistics are used.

Efficiency Index (EI)

Nash and Sutclifee [15] proposed the efficiency index for measuring the performance of a given model:

$$EI = \frac{SR}{ST} \qquad (24),$$

$$SR = ST - SE \qquad (25)$$

$$ST = \sum_{i=1}^{M}(y_i - \bar{y})^2 \qquad (26)$$

$$SE = \sum_{i=1}^{M}(y_i - \hat{y}_i)^2 \qquad (27)$$

$$\bar{y} = \frac{1}{M}\sum_{i=1}^{M} y_i \qquad (28)$$

where	SR	= Variation explained by the model,	ST	=	Total variation,
	SE	= Sum of squared errors,	y_i	=	Actual output (observed value) at time i,
	$\bar{y}$	= Mean value of the actual output,	$\hat{y}_i$	=	Model output at time i,
	M	= Number of data points (training patterns) used.			

EI is the indicator measuring the efficiency of the network. The closer the value is to 1 (or 100 percent), the better is the model representation.

Mean Absolute Deviation (MAD)

MAD is the average of the absolute value of the difference between forecast and observation as shown by the equation:

$$MAD = \frac{1}{M}\sum_{i=1}^{M}|y_i - \hat{y}_i| \qquad (29)$$

MAD values near or equal to zero indicate perfect or near perfect forecasts. This measure is heavily weighted towards large differences in forecast.

Maximum Relative Error ($\mathbf{RE_{max}}$)

RE_{max} indicates the maximum error between forecast and observation. When the value of RE_{max} is near or equal to zero, it indicates near perfect forecasts.

$$RE_{\max} = \max\left(\left|\frac{y_i - \hat{y}_i}{y_i}\right|\right) \qquad i = 1,\ldots, M \qquad (30)$$

From the definitions of EI, SE and RMSE, it is clear that when EI is high, RMSE and SE are low, and conversely, when EI is low, RMSE and SE are high. As such, EI can fully indicate the performance of the model without the need of SE and RMSE.

To measure the performance statistics of selected networks, the data series are divided into two phases, approximately 2/3 for calibration and 1/3 for validation. Since the logistic activation function approaches 0 and 1 asymptotically when the variable approaches negative infinity and positive infinity respectively, the data are transformed to the range [0.05, 0.95] instead of [0,1]. Moreover, the data series are fed into the network in the following manner;

$$V_1(t+1)=g(V_1(t),V_1(t-1),V_1(t-2),V_2(t),V_2(t-1),V_2(t-2)) \\ +g(V_1^{H1}(k-1),V_1^{H2}(k-1),V_1^{O}(k-1)), \tag{31}$$

where t is daily data time, k is time step of the network and g denotes symbolically the "functions" of the variables shown. The network inputs can be divided into two phases. The first phase is composed of two variables; V_1 and V_2 as the external inputs. The ratio of data applied for the first phase between two variables varies depending on type of data. For instance, V_1 and V_2 are price and volume respectively, the ratio is 3:3. The second phase is composed of the net outputs of hidden nodes and output node from the previous state. It is important that characteristics of recurrent network depend on this phase since it is mainly control aspects of weight connections.

Let ***m***(***n***)-***p***-***r*** denote a recurrent network, having ***n*** units and ***m*** external inputs. The ***n*** units are composed of ***p*** hidden nodes and ***r*** output nodes. The 6(3)-2-1 network architecture is applied for FRNN, N&P and FGS networks. In other words, there are six external inputs, two hidden nodes, and one output node, respectively. All feedback connection weights equal to 1.0. The sigmoid slope was set to 1. The forecasting lead-time is equal to one day.

For financial data, *Stock Prices* in Thai Market is taken as an example. All data were obtained from The Securities Exchange of Thailand (SET), as formally enacted in 1974 and the SET began trading on April 30, 1975. Three companies were selected; Land and Houses Public Company Limited (LH) in real estate property sector, Bangkok Bank Public Company Limited (BBL) in banking sector, and Shin Corporation Public Company Limited (SHIN) in communication sector. Daily data on the stock prices and volumes in the market from 1993 to 1996 were used. The stock market is closed on weekends; therefore, consecutive data series are from Mondays to Fridays.

5. PERFORMANCE RESULTS AND DISCUSSIONS

Performance statistical results from network experiments with a selected set of data are used to compare the recurrent networks. The performance statistical results of networks depend on their intrinsic characteristics of networks in the sense that a weight update needs different information in each network. In this section, we shall focus and discuss on speed of training and network performances indicated by three performance statistics classified by the type of training algorithms and data sets.

To measure the applicability of Yamamoto-Nikiforuk or EBP-EWLS algorithm for forecasting, three network architectures, Fully Recurrent Neural Network (FRNN), Narendra & Parthasarathy (N&P) and Frasconi-Gori-Soda (FGS) were used for the comparative study. The forgetting factor at 0.99 which was the best suitable value that leads the network to fastest convergence that is used. This value of the forgetting factor is set through all the data sets. From the performance results, it can be noticed that when the validation phase has the similar patterns as calibration phase, the networks can forecast well.

Tables 1 and 2 show the performance results among selected networks for the calibration and validation phase of financial forecasts. Because of the limitations associated with each individual measure, the three performance measures were used. The EI values of almost cases close to 1, all the values of MAD are near zero but for the values of RE_{max} are not near zero. Observing the stock price performance results on RNNs, it is noted that FRNN has the larger value of EI. This is also true for all network architectures that are for LH. FRNN for LH has the highest EI and computational time overall per epoch, suggesting the highest skill in capturing predictability. In contrast, FGS for LH has the highest computational time while the lowest EI is N&P for SHIN. All of the models tend to show high skill in predicting stock price, as indicated by the EI, which are all at least 0.87. According to Table 2 (validation phase), the highest EI is FRNN for SHIN, however the value of RE_{max} is not lowest. The lowest RE_{max} is FRNN for BBL, 0.149. It can be seen from the Figure 5, at the tail of line that FGS for LH illustrated the most line distance error between forecast and observation as indicated by RE_{max}.

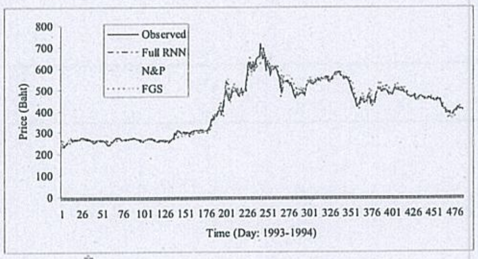
Table 1. Performance statistics among difference network architecture (Calibration)

Stock Company	Network Type	EI	MAD	RE_{max}	Time(s)
LH	FRNN	0.991	0.013	0.370	0.412
	N&P	0.982	0.020	0.691	0.359
	FGS	0.972	0.026	0.736	0.538
BBL	FRNN	0.987	0.017	0.698	0.411
	N&P	0.959	0.025	5.879	0.357
	FGS	0.953	0.028	6.066	0.284
SHIN	FRNN	0.932	0.017	1.082	0.501
	N&P	0.875	0.033	1.089	0.394
	FGS	0.882	0.036	0.936	0.320

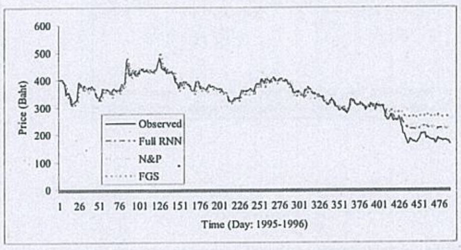
Note: Time(s) based on seconds.

Table 2. Performance statistics among difference network architecture (Validation)

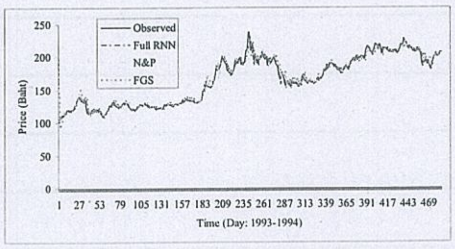
Stock Company	Network Type	EI	MAD	RE_{max}
LH	FRNN	0.951	0.018	10.087
	N&P	0.921	0.026	10.288
	FGS	0.795	0.037	26.641
BBL	FRNN	0.910	0.027	0.149
	N&P	0.839	0.039	0.209
	FGS	0.561	0.063	0.212
SHIN	FRNN	0.983	0.007	0.376
	N&P	0.791	0.031	1.029
	FGS	0.786	0.030	1.697



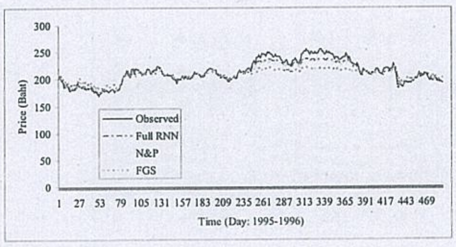
LH (a)



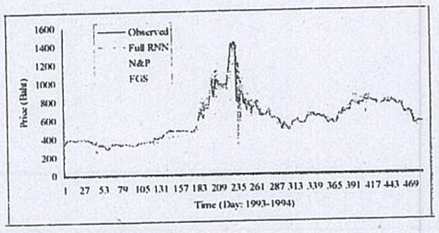
(b)



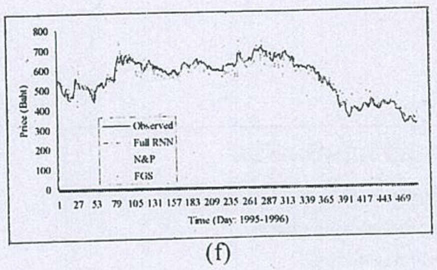
BBL (c)



(d)



SHIN (e)



(f)

Figure 5. Observed vs. forecast stock prices (Baht) using Yamamoto-Nikiforuk algorithm: (a), (c) and (e): for calibration. (b), (d) and (f): for validation.

- **Comparing based on three performance statistics**

Using the model accuracy expressed by the values of 1, 2, 3 for the EI, MAD and RE_{max}, we can say that in some cases the values of EI go along with the values MAD but some go along with the values of RE_{max}.

Stock Company	Network Types	EI	MAD	RE_{max}
LH	FRNN	1	1	1
	N&P	2	2	2
	FGS	3	3	3
BBL	FRNN	1	1	1
	N&P	2	2	2
	FGS	3	3	3
SHIN	FRNN	1	1	2
	N&P	3	2	3
	FGS	2	3	1

where 1 means "that case provides the highest accuracy"
2 means "that case provides the second best of the accuracy"
3 means "that case provides the least accuracy"

A close inspection leads to the following observations:

- Due to the values of EI related to SE and the formula of SE is the sum of squared error, while MAD is the absolute value of the errors but RE_{max} is the maximum error. So the results of EI and MAD show that their values all close to the excellent performance, but the values of RE_{max} depend on the fluctuation of the data.

The time series data used in this study correspond to the period of Thai economic would occur the Thai financial crisis. As such, all stocks are generally hard to predict due to their high volatility. So the superior performance of the RNNs in forecasting has been noticed as following

Stock Company	Network Types	EI	MAD	RE_{max}
LH	FRNN	*	*	***
	N&P	*	*	***
	FGS	*	**	***
BBL	FRNN	*	*	***
	N&P	*	**	***
	FGS	*	**	***
SHIN	FRNN	**	*	***
	N&P	**	**	***
	FGS	**	**	***

where * "means EI > 0.95, MAD or RE_{max} <0.20" (excellent)
** "means EI > 0.80, MAD or RE_{max} <0.50" (acceptable)
*** "means not in the above cases"

- **Comparing based on three network architectures**

We demonstrated how Partially Recurrent Neural Networks (PRNNs) can be used interchangeably with Fully Recurrent Neural Networks (FRNNs) by not decreasing the network performance. Although FRNNs can forecast well but the requirement of computational time is quite high compared with PRNNs. In the experimental study, the difference of computational time requirement between FRNN and PRNN is slightly different because of the use of small networks. Since it is true that a PRNN needs a number of input signals less than a FRNN, it is then possible to take advantage of the abilities of PRNNs to train the network. In this section, we shall summarize the performance results based upon the selected networks and training algorithms

- Efficiency Index

  FRNN > N&P, FGS

  Typically, N&P and FGS networks have EI less than FRNN because of less information for capturing variability (less weight connections). Among PRNNs themselves, the network performance is hardly to make a comparison since it also depends on the selected training algorithms.

- Computational Time

  FRNN > N&P, FGS

  Normally, N&P and FGS networks required lower computational time than FRNN.

6. CONCLUSIONS

The first goal of this paper has been to apply the three performance statistics : Efficiency Index (EI), Mean Absolute Deviation, Maximum Relative Error (RE_{max}) to the problem of stock prices forecasting. The other goal has been to compare between different recurrent neural networks (RNNs) : fully recurrent neural network (FRNN), Narendra & Parthasarathy (N&P) and Frasconi-Gori-Soda (FGS).

We have seen that generalization performances were very reasonable which indicating confidence in the experiment results as a way to the different RNNs for this single step ahead forecasting. The comparative study on the RNN models show that the three networks produced fairly accurate forecast. By means of the experiments (using three performance statistics which all are prediction accuracy measure and are defined in terms of error), the result show that EI and MAD are more suitable than RE_{max}.

7. ACKNOWLEDGEMENTS

The authors would like to thank Associate Professor Dr. Chanongkorn Kuntonbutr of Business Administration Faculty, Rajamangala University of Technology, Thanyaburi (RMUT) for the supporting throughout the study and providing the opportunity to submit this paper to KSAS.

REFERENCES

[1] Rumelhert, D.E. Hinton, G.E. and Williams, R.J. **1986** Learning Internal Representations by Error Propagation. In : Rumelhert, D.E. and MeCleland, J.L. Eds. *Parallel Distributed Processing.* Massachusetts MIT Press, pp. 318-362.

[2] Chu, C.H. and Widjaja, D. **1994** Neural Network System for Forecasting Method Selection. *Decision Support Systems*, 12, 13-24.

[3] NeuralWare, Inc. **1991** *Neural Computing, Neuralworks Professional II/Plus and NeuralWorks Explorer*. Pittsburgh.

[4] Narendra, K.S. and Parthasarathy, K. **1990** Identification and Control of Dynamical Systems using Neural Networks. *IEEE Transactions on Neural Networks*. 1(1), 4-27.

[5] Fransconi, P. Gori, M. and Soda, G. **1992** Local Feedback Multilayered Networks. *Neural Computation*, 4(1), 120-130.

[6] Medsker, L.R. and Jain, L.C. **2000** *Recurrent Neural Networks : Design and Application.* Florida, The CRC press.

[7] Atiya, A.F. and Parlos, A.G. **2000** New Results on Recurrent Network Training : Unifying the Algorithms and a Accerating Convergence, *IEEE Transaction on Neural Networks*, 11(3), 697-709.

[8] Yamamoto, Y. and Nikiforuk, P.N. **1999** A Learning Algorithm for Recurrent Neural Networks and Its Application to Nonlinear Identification, *Proceeding of IEEE International Symposium on Computer Aided Control System Design*, 551-556.

[9] Makridakis, S. Wheelwright, S.C. and McGee, V.E. **1983** *Forecasting : Methods and Applications*. 2nd Edition. New York, John Wiley.

[10] Gorr, L. **1994** Research rospective on Neural Network Forecasting, *International Journal of Forecasting*, 10, 1-4.

[11] Hann, T.H. and Steurer, E. **1996** Much Ado About Nothing Z. Exchange Rate Forecasting : Neural Networks vs. Linear Models Using Monthly and Weekly Data, *Neurocompting*, 10, 323-339.

[12] Foster, W.R. Collopy, F. and Ungar, L.H. **1992** Neural Network Forecasting of Short, Noisy Time Series. *Computers and Chemical Engineering*, 16(4), 293-297.

[13] Cottrell, M. Girard, B. Girard, Y. Mangeas, M. and Muller, C. **1995** Neural Modeling for Time Series : A Statistical Stepwise Method for Weight Elimination, *IEEE Transaction on Neural Networks* 6(6), 1355-1364.

[14] Armstrong, J.S. and Fildes, R. **1995** Correspondence : On the Selection of Error Measures for Comparisons among Forecasting Methods. *Journal of Forecasting*, 14, 67-71.

[15] Nash, J.E. and Sutcliffee, J.V. **1970** River Flow Forecasting Through Conceptual Models, *Journal of Hydrology*, 10, 282-290.