Ç.Ü. Sosyal Bilimler Enstitüsü Dergisi, Cilt 22, Sayı 2, 2013, Sayfa 97-110

INTEGRATING ARTIFICIAL NEURAL NETWORK MODELS BY MARKOV CHAIN PROCESS: FORECASTING THE MOVEMENT DIRECTION OF TURKISH LIRA/US DOLLAR EXCHANGE RATE RETURNS

Süleyman Bilgin Kılıç*

ABSTRACT

In this study, we first modeled daily US dollar returns as the discrete state Markov chain process, and second we trained an Artificial Neural Network (ANN) model in order to estimate direction of dollar return. The trained model provides valuable information about the direction of next day return.

Keywords: Artificial Neural Networks, Markov chains, Conditional probability, Exchange rate returns
Jel: C45, C53, F31

1. Introduction

The foreign exchange market is the largest and most liquid financial market in the world. The Bank for International Settlements (BIS) survey which is conducted every three years, reported that the daily turnover in the world's foreign-exchange markets has soared to $4 trillion in April 2010. The U.S. dollar had a share in 85% of all transactions; the Euro gained two percentage points since the 2007 survey to account for 39% of all transactions. The market share of emerging-market currencies increased, with the biggest gains for the Turkish lira and the Korean won. According to BIS survey the biggest increase was seen by the Turkish lira, which saw its share rise to 0.7% from 0.2%, followed by the Korean won, which rose to 1.5% from 1.2% (http://articles.marketwatch.com).

Foreign-exchange rate forecasting is very important for international firms that making import and export, also for governments, individual investors planning to invest abroad, and seek profit or protection through speculative action. Market conditions drive exchange rates up and down every day, imposing risks on participants of the foreign exchange markets. Predicting exchange rates more accurately would allow businesses, investors, and policymakers to make wise decisions when conducting international business and economic policies.

Many important studies attempt to forecast the exchange rate returns by using Artificial Neural Network (ANN), genetic algorithm and machine learning approach. Empirical results of these studies generally suggested that these approaches outperform over the traditional time series models. Among them, Zhang and Michael (1998) found that neural networks outperform linear models, particularly when the forecast horizon is

* Doç.Dr., Çukurova Üniversitesi, İİBF, Yöneylem ABD., sbilgin@cu.edu.tr

short. In addition, the number of input nodes has a greater impact on performance than the number of hidden nodes, while a larger number of observations do reduce forecast errors. Leung at al (2000) examined the forecast ability of a specific neural network architecture called general regression neural network (GRNN) and compare its performance with a variety of forecasting techniques, and they found that GRNN not only has a higher degree of forecasting accuracy but also performs statistically better than other evaluated models for different currencies. Shin and Han (2000) proposed an integrated thresholding design of the optimal or near-optimal wavelet transformation by genetic algorithms (GAs) to represent a significant signal in artificial neural network models. This approach is applied to Korean won/US dollar exchange-rate forecasting. The experimental results show that this integrated approach using GAs has better performance than the wavelet thresholding algorithms. Qi and Wu (2003) employed a neural network to study the nonlinear predictability of exchange rates for four currencies at the 1-, 6- and 12-month forecast horizons. They find that neural network model with market fundamentals cannot beat the random walk in out-of-sample forecast accuracy and their overall results suggest that neither nonlinearity nor market fundamentals appear to be very important in improving exchange rate forecast for the chosen horizons. Panda and Narasimhan (2007) compare the forecasting accuracy of neural network with that of linear autoregressive and random walk models. Using six forecasting evaluation criteria, they find that neural network has superior in-sample forecast than linear autoregressive and random walk models. This finding provides evidence against the efficient market hypothesis and suggests that there exists always a possibility of extracting information hidden in the exchange rate and predicting it into the future. Hussain at al (2008) proposed a novel type of higher-order polynomial pipelined neural network to predict the exchange rate between the US dollar and three other currencies and stated that the network demonstrates more accurate forecasting over a number of benchmarked neural networks. Kadılar et al (2009) proposed to employ ANN method for forecasting Turkish TL/US dollar exchange rate series and their results show that ANN method has the best forecasting accuracy with respect to time series models, such as seasonal ARIMA and ARCH models. Kayacan et al (2010) investigated the accuracies of different grey models such as GM(1,1), Grey Verhulst model, modified grey models using Fourier Series. The United States dollar to Euro parity are used to compare the performances of the different models. The simulation results show that modified grey models have higher performances not only on model fitting but also on forecasting. Among these grey models, the modified GM(1,1) using Fourier series in time is the best in model fitting and forecasting. Anastasakis and Mort (2009) applied both parametric (neural networks with active neurons) and nonparametric (analog complexing) self-organising modelling methods for the daily prediction of the exchange rate market. They proposed a combined approach where the parametric and nonparametric self-organising methods are combined sequentially, exploiting the advantages of the individual methods with the aim of improving their performance. The combined method is found to produce promising results and to outperform the individual methods when tested with two exchange rates: the American Dollar and the Deutche Mark against the British Pound. Sermpinis et al (2012) investigated the use of a neural network architecture, the Psi Sigma Neural Network (PSN), when applied to the task of forecasting and trading the Euro/Dollar exchange

rate to explore the utility of Kalman Filters in combining neural network forecasts. They reported that the PSN outperforms all models' individual performances in terms of statistical accuracy and trading performance and the forecast combinations also present improved empirical evidence, with Kalman Filters outperforming by far its benchmarks. Kempa and Riedel (2013) analyze bilateral Canadian-US dollar exchange rate movements within a Markov switching framework with two states, one in which the exchange rate is determined by the monetary model, and the other in which its behavior follows the predictions of a Taylor rule exchange rate model. They stated that strong evidence of nonlinearities also confirms the notion that exchange rate movements cannot be explained exclusively in terms of any one particular exchange rate model. Yubo (2013) presented a new machine learning model; the polynomial smooth support vector machine. After being solved by Broyden–Fletcher–Goldfarb–Shanno method, he obtained optimal forecasting parameters. In the study, the exchange rate movement direction of Chinese Renminbi vs. United States Dollars is investigated. Six indexes of Dow Jones China Index Series are used as the input. 4 sections with 180 time experiments have been completed. Many results of the study show that the proposed learning model is effective and powerful.

In this study, the daily dollar exchange returns are modeled as a stochastic process with sixthteen discrete state spaces of Markov chain that the conditional probability of any next future return state depends only on the present return of the state and is independent any other states of the past returns. Thus, we transformed returns into sixthteen equal discrete categorical intervals of states, from high loss (negative return) to positive high return. Objective of modeling the returns as Markov chain process is to calculate probability of positive return for the next step (day) given the present state. These probabilities are used as inputs to the ANN models for training process for prediction of next day's return direction (positive/non positive). We trained three ANN models; Model (0), Model (1) and Model (0,1). Classification results are given for the training and testing sample. In the study the whole sample data were randomly divided two equal groups as training and testing sample; training sample was used to train the models, testing sample was used to evaluate the models in terms of the classification achievements.

The rest of this article is organized as follows. Section 2 includes the sample selection, methodology and empirical results; section 3 concludes the article and discusses some future research perspectives.

2. The Sample, Methodology and Results

2.1. The Sample

The sample data covers 4485 daily closing price of US dollar (P_t) between the period of January 02, 1995-October 17, 2012. The data were obtained from the electronic data delivery system of the Central Bank of Turkey.

Daily returns of dollar (R_t) are computed as a percentage change of the daily closing price of dollar (P_t); $R_t = (P_t - P_{t-1})/P_{t-1}$ .Here, t represents the days

(t=1…4485).The average (expected) return is calculated (μ_R) as approximately 0.0009 with a standard deviation of (σ_R) 0.0108 for the period considered. The standard deviation is extremely high in comparison to expected return (approximately 11.9 times of the expected return). So, return exhibits high volatility, implying an enormous risk for the investors.

2.2. The Methodology and Results

2.2.1. Modeling the returns as the discrete categorical states of Markov chain process

In the study, the daily returns are assumed to be a stochastic process with sixthteen discrete state spaces $\{S_1,..,S_{17}\}$ with Markov chain that conditional probability of any next future return state (S_j^t) depends only on the present return of the state (S_i^{t-1}) and is independent any other states of the past returns $P\left(S_j^t \middle| S_i^{t-1}\right)$.

As stated previously, objective of modeling the returns as discrete categorical states is to calculate probability of positive/non positive return for the next step (day) given the present state. In section 2.2.2, these probabilities are used as inputs to the ANN models for training process for prediction of next day's return direction. Thus, we transformed returns into sixthteen equal discrete categorical intervals of states, from high loss (negative return) to positive high return.

In Table 5 of Appendix A the total number of return transitions, occurring from the present day to the next day, from states S_i to S_j , were calculated for the whole period considered. We can easily compute the one step (one day) conditional transition probability matrix $P\left(S_j^t \middle| S_i^{t-1}\right)$ from Table 5 of Appendix A, from state i to j by dividing the row elements by row total. This matrix is given in Table 6 of Appendix A. Here, when the return in state S_1 in the present day, conditional probability of it will be going to S_{17} in the next day is $P(S_{17}^t | S_1^{t-1}) = 0.1485$. Similarly, conditional probability of passing from state S_9 to S_{13} is $P(S_{13}^t | S_9^{t-1}) = 0.0777$.

From Table 6 of Appendix A probability of non-positive return in the next day (step) given the present state i can be calculated by equation (1).

$$P(R_t \le 0) \middle| S_i^{t-1} = \sum_{j=1}^{9} P\left[(S_{ij}^t) \middle| S_i^{t-1}\right] \quad (1)$$

And probability of positive return is given the present state i can be calculated by equation (2).

$$P(R_t > 0) \middle| S_i^{t-1} = 1 - P\left[(R_t \le 0) \middle| S_i^{t-1})\right] \quad (2)$$

Table 1 gives each of the states, return range, percentage of occurrence the states, and probability of positive return in the next day (step) given the present state for the period considered.

Table.1
States, return ranges, percentage of occurrence of states, and probability of positive return in the next step

Present states (S_i^{t-1})	**Return Range**	**% of occurrence of states**	**Probability of positive return for next step given the preset state** $P(R_t > 0)\|S_i^{t-1}$
S_1	$-0.0108 \leq R_t$	5.11	0.3799
S_2	$-0.0108 < R_t \leq -0.0095$	1.81	0.3580
S_3	$-0.0095 < R_t \leq -0.0081$	2.12	0.4211
S_4	$-0.0081 < R_t \leq -0.0068$	2.56	0.4696
S_5	$-0.0068 < R_t \leq -0.0054$	3.55	0.5157
S_6	$-0.0054 < R_t \leq -0.0041$	4.82	0.4352
S_7	$-0.0041 < R_t \leq -0.0027$	6.51	0.5205
S_8	$-0.0027 < R_t \leq -0.0014$	7.54	0.4970
S_9	$-0.0014 < R_t \leq 0.0000$	10.61	0.5672
S_{10}	$0.0000 < R_t \leq 0.0014$	10.99	0.6349
S_{11}	$0.0014 < R_t \leq 0.0027$	10.21	0.6397
S_{12}	$0.0027 < R_t \leq 0.0041$	8.52	0.6728
S_{13}	$0.0041 < R_t \leq 0.0054$	7.09	0.6541
S_{14}	$0.0054 < R_t \leq 0.0068$	4.57	0.5220
S_{15}	$0.0068 < R_t \leq 0.0081$	2.94	0.5909
S_{16}	$0.0081 < R_t \leq 0.0095$	2.52	0.4690
S_{17}	$R_t > 0.0095$	8.54	0.5196

2.2.2. Training of the ANN models

By using the information (probabilities of positive or non positive return in the next step given the present state i) provided by Markov chain process in the previous section, we trained three ANN models; Model (0), Model(1) and Model(0,1). After performing so many experiments.

Table.2
Inputs and outputs of the trained models

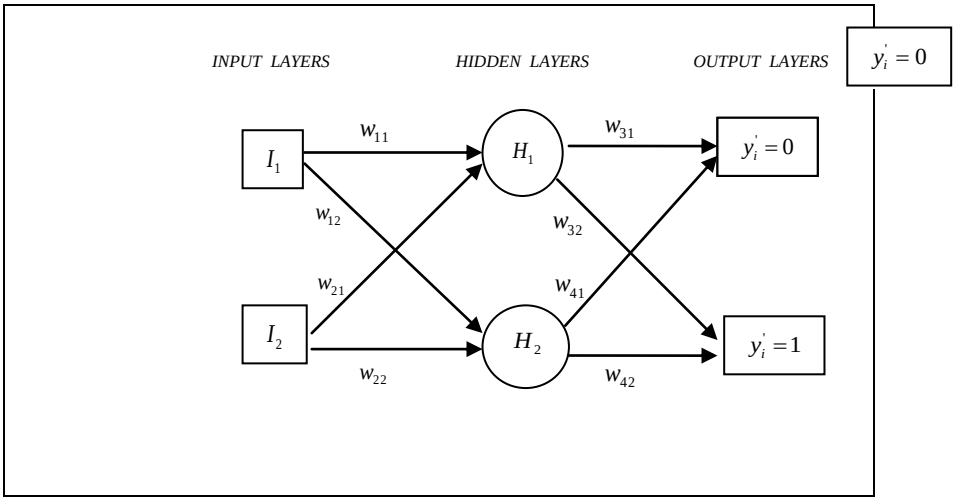
	Inputs	Output
Model (0)	$R_{t-1} \leq 0$	$R_{t-1} \leq 0$
	$P(R_t \leq 0)\|S_i^{t-1}$	$R_{t-1} > 0$
Model (1)	$R_{t-1} > 0$	$R_{t-1} \leq 0$
	$P(R_t > 0)\|S_i^{t-1}$	$R_{t-1} > 0$
Model (0,1)	R_{t-1}	$R_{t-1} \leq 0$
	$P(R_t > 0)\|S_i^{t-1}$	$R_{t-1} > 0$

Inputs of the trained models are given in Table 2. Inputs of the Model (0) are non-positive return of the present state and probability of non-positive return in the next state given the present state. Inputs of the Model (1) are positive return of the present state and probability of positive return in the next state given the present state. Inputs of the Model (0,1) are present state return (either positive or non-positive) and probability of positive return in the next day given the present state i .

Outputs of the three models are the same; prediction of either non-positive or positive returns for the next state and defined by following function:

$$y_i^{'} = \begin{cases} 0, & if\ R_t \le 0 \\ 1, & if\ R\ > 1 \end{cases} \tag{3}$$

Figure1 *Network diagrams of the estimated ANN models*



All of the three models trained have the same architecture that is given Figure 1. The ANN models consist of two input and output nodes in the input and output layer, and one hidden layer with two nodes between input and output layer.

By using gradient decent multilayer perceptron procedure the ANN models can be trained as follows;

Hidden node H_1 contains the following sigmoid function:

$$H_1 = 1/(1+e^{-Z_1}), \quad z_1 = w_{11}I_1 + w_{21}I_2 \tag{4}$$

The other hidden node H_2 contains the following function:

$$H_2, = 1/(1+e^{-Z_2}), \quad z_2 = w_{12}I_1 + w_{22}I_2 \tag{5}$$

Inputs of the output node ($y_i^{'} = 0$) are the outputs from the two hidden nodes H_1 and H_2 which are weighted by w_{31} and w_{41};

$$y^{'}{}_i = w_{31}\left(1/e^{-H_1}\right) + w_{41}\left(1/e^{-H_2}\right) \tag{6}$$

Similarly, inputs of the output node ($y_i^{'}=1$) are the outputs from the two hidden nodes H_1 .and H_2 which are weighted by w_{41} and $w_{42;}$

$$y^{'}{}_i = w_{32}\left(1/e^{-H_1}\right)+ w_{42}\left(1/e^{-H_2}\right) \quad (7)$$

Hence, the prediction error is the difference between the actual direction and predicted direction; $e_i = y_i - y_i{}'$ (8)

Here, y_i is the actual (observed) direction. The total prediction error (E) can be written as a function of ANN weights (w_{ij});

$$E(w) = \sum_{i=1}^{n} e_i^2 \quad (9)$$

Gradient vector of the total prediction error function can be written as;

$$\nabla E(w) = (\partial E/\partial w) \quad (10)$$

ANN weights can be adjusted by the gradient descent method;

$$w_{new} = w_{old} + \alpha(\partial E/\partial w)\Big|_{w_{old}}$$

Here, α = learning parameter ($0 \leq \alpha \leq 1$). Iteration eventually terminates at a local minimum when $w_{new} \cong w_{old}$.

After the training process, adjusted connection weights for Model (0), Model (1) and Model(0,1) are given in the Table 3. For example, for Model (0) connection weights between input node I_1 and hidden layer node H_1 and H_2 (w_{11}, w_{12}) are -0.4985 and -0.4900 respectively. Weights between hidden layer node H_1 and output node $y_i^{'}=0$ and $y_i^{'}=1$ are -0.0370 and -0.6580 respectively. Weights between hidden layer node H_2 and output layer nodes $y_i^{'}=0$ and $y_i^{'}=0$ are 0.4131 and 0.0185 respectively.

Table.3
Adjusted weights for the three ANN models

Model (0)		H_1	H_2	$y_i^{'}=0$	$y_i^{'}=1$
Input Layer	I_1	-0.4985	-0.4900		
	I_2	0.0237	0.1771		
Hidden Layer 1	H_1			-0.0370	-0.6580
	H_2			0.4131	0.0185
Model (1)					
Input Layer	I_1	1.2241	0.7463		
	I_2	-0.6162	-0.1462		
Hidden Layer 1	H_1			-0.5939	
	H_2			-0.6232	
Model (0.1)					
Input Layer	I_1	0.010	-0.309		
	I_2	2.1850	-1.330		
Hidden Layer 1	H_1			-1.511	1.1530
	H_2			0.4650	-0.993

Table 4 gives observed and predicted classification results of daily direction of dollar returns by the estimated three ANNs model. Classification results are given for the training and testing sample. In the study the whole sample data were randomly divided two equal groups as training and testing sample; training sample was used to train (estimate) the models, testing sample was used to evaluate the models in terms of the classification achievements.

In last column of Table 4 for the testing sample we can see that if the present return is non-positive, the Model (0) predicts next day non positive direction ($R_t \leq 0$) 56% correctly for testing sample. If the present return is positive the Model (1) predicts next day positive direction ($R_t > 0$) 97% correctly.

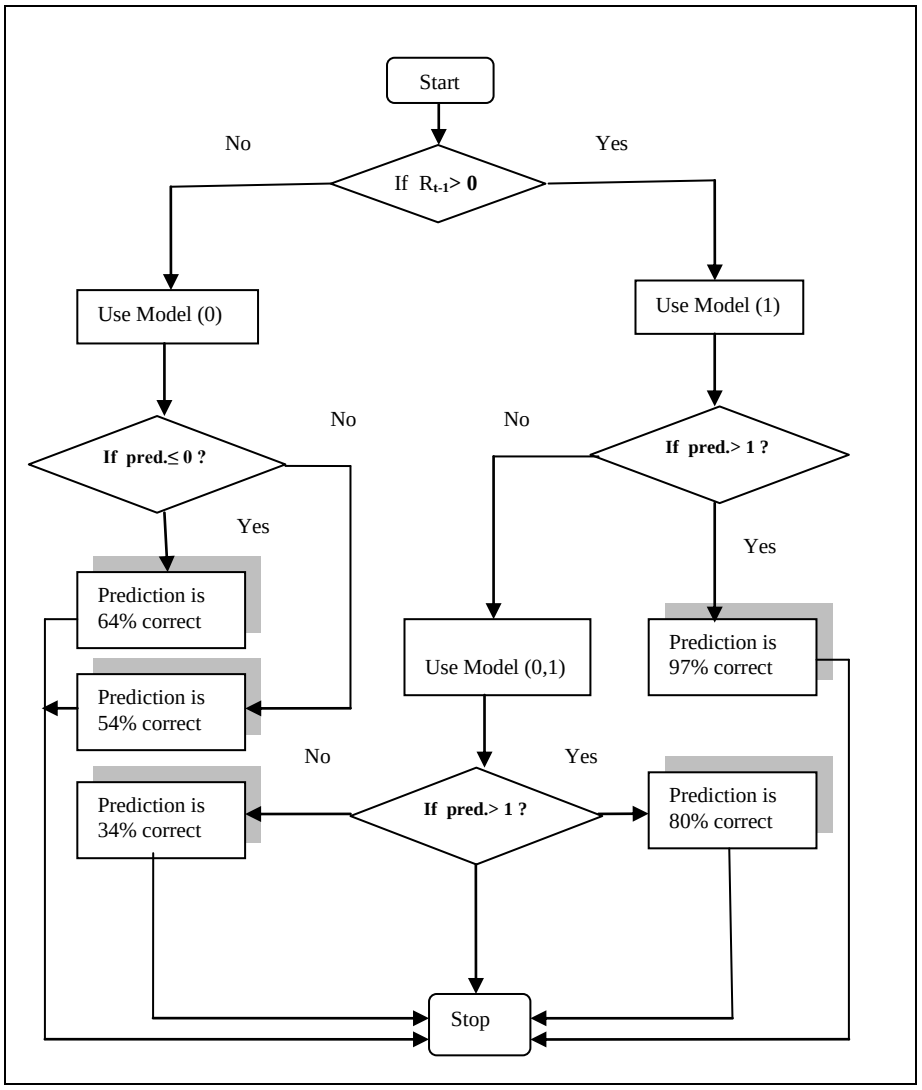
However, if the present return is positive, the Model (1) does not accurately predicts negative direction (8%). If the present return is either positive or non-positive the Model (0,1) correctly predicts next day non-positive return 34%. In order to eliminate these unreliable predictions and to make more accurate prediction, the three models can be integrated together for the prediction process.

Table.4
Classification achievement of estimated ANN models

		Predicted		
Model (0)	**Observed**	*D≤0*	*D>1*	**Correct classification (%)**
Training	*D≤0*	316	175	0.56
	D>1	227	265	0.46
	Overall	55.20%	44.80%	0.51
Testing	*D≤0*	278	219	***0.64***
	D>1	251	217	***0.54***
	Overall	54.80%	45.20%	***0.59***
Model (1)				
Training	*D≤0*	31	468	0.06
	D>1	38	748	0.95
	Overall	5.40%	94.60%	0.61
Testing	*D≤0*	37	425	***0.08***
	D>1	25	765	***0.97***
	Overall	5.00%	95.00%	***0.64***
Model(0.1)				
Training	*D≤0*	316	640	33.1
	D>1	276	995	78.3
	Overall	26.6%	7 3.4%	58.9
Testing	*D≤0*	339	654	***34***
	D>1	256	1009	***80***
	Overall	26.4%	73.6%	***59.7***

Flowchart of integrated use of these models in prediction process is given in Figure 2. Here, if the present return is positive the Model (1) should be used. If the Model (1) predicts positive return, prediction is 97% correct, stop the prediction. If the Model (1) predicts non-positive return do not use model (1); use model (0,1). If Model (0,1) predicts positive direction it is prediction 80% correct, stop prediction. If Model (0,1) predicts non-positive direction it is 34% correct stop prediction.

Figure 2
Flowchart of using the three combined models for prediction



If the present return is non-positive the Model (0) should be used. If the Model (0) predicts non-positive return, prediction is 64% correct stop the prediction. If the Model (0) predicts positive return it is 54% correct, stop the prediction.

Hence, as an average integrated use of the three models provides 65.8% correct prediction for the direction of returns. This means that if an investor determines his/her daily buying-selling strategy according to the prediction of the integrated models, his/her daily investment strategy will be profitable 65.8% of the time in the long run.

3. Conclusion

This study combined Markov chain process with the ANN models. Composite use of ANN models provides valuable information about daily direction of the dollar return given the present state. Similar further analysis can be performed for the returns of individual common stocks and other investment instruments such as gold and other foreign exchange returns.

This study uses daily returns. Further similar analysis can also be performed by considering returns of smaller time intervals, such as an intraday hourly change. Hence, using small time intervals may provide more information.

References

Anastasakis, L. & Mort N.(2009). Exchange rate forecasting using a combined parametric and nonparametric self-organising modelling approach**,** *Expert Systems with Applications*, Volume 36, Issue 10, pp 12001-12011.

Hussain, A.J., Knowles, A., Lisboa, P.J.G., El-Deredy, W. (2008). Financial time series prediction using polynomial pipelined neural networks, *Expert Systems with Applications*, Volume 35, Issue 3, pp 1186-1199.

Kadılar, C., Şimşek, M., Aladağ, Ç.H. (2009). Forecasting the exchange rate series with ANN: the case of Turkey , *İstanbul Üniversitesi İktisat Fakültesi Ekonometri ve İstatistik Dergisi*, Volume 9, pp 17-29.

Kayacan, E., Ulutas, B., Kaynak, O. (2010). Grey system theory-based models in time series prediction**,** *Expert Systems with Applications*, Volume 37, Issue, 2, pp 1784-1789.

Kempa, B. & Riedel, J. (2013). Nonlinearities in exchange rate determination in a small open economy: Some evidence for Canada, *The North American Journal of Economics and Finance*, Volume 24, Issue January, pp 268-278.

Leung M.T., Chen, A.S., Dauk, H. (2000). Forecasting exchange rates using general regression neural networks, *Computers & Operations Research*, Volume 27, Issue 11-12, pp 1093-1110.

Panda, C. & Narasimhan, V. (2007). Forecasting exchange rate better with artificial neural network, *Journal of Policy Modeling*, Volume 29, Issue March–April, pp 227-236.

Sermpinis, G., Dunis, C., Laws, J., Stasinakis, C. (2012). Forecasting and trading the EUR/USD exchange rate with stochastic Neural Network combination and time-

varying leverage, *Decision Support Systems*, Volume 54, Issue 1, pp 316-329

Shin, T. & Han, I. (2000). Optimal signal multi-resolution by genetic algorithms to support artificial neural networks forexchange-rate forecasting, *Expert Systems with Applications*, Volume 18, Issue 4, pp 257-269.

Qi, M., & Wu, Y.(2003). Nonlinear prediction of exchange rates with monetary fundamentals, *Journal of Empirical Finance*, Volume 10, Issue 5, pp 623-640

Yubo, Y. (2013). Forecasting the movement direction of exchange rate with polynomial smooth support vectormachine, *Mathematical and Computer Modelling*, Volume 57, Issue 3-4, 932-944.

Zhang G, B. & Michael Y.H. (1998). Neural network forecasting of the British Pound/US Dollar exchange rate, *Omega*, Volume 26, Issue 4, pp 495-506.

Appendix A

Table 5
Number of occurrence of the transitions from states S_i to S_j

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}	S_{17}	*Row total*
S_1	41	5	7	10	9	17	18	15	20	14	4	12	5	4	8	6	34	229
S_2	7	4	5	2	0	3	9	10	12	3	4	7	3	4	1	2	5	81
S_3	9	2	2	2	9	3	6	11	11	4	9	3	1	7	2	3	11	95
S_4	12	1	1	3	9	11	5	8	11	5	2	8	8	5	7	5	14	115
S_5	6	4	3	3	9	11	14	13	14	14	14	13	5	10	8	2	16	159
S_6	10	4	11	8	11	17	15	23	23	20	20	12	14	9	5	8	6	216
S_7	9	6	6	7	12	18	24	33	25	36	23	28	18	13	13	3	18	292
S_8	7	7	3	14	14	25	31	27	42	41	34	27	19	13	6	6	22	338
S_9	17	8	8	10	14	26	23	40	60	56	45	45	37	24	19	11	33	476
S_{10}	14	4	5	11	11	13	33	31	58	70	65	46	48	31	17	10	26	493
S_{11}	6	3	6	7	13	15	29	37	49	71	72	46	44	21	11	11	17	458
S_{12}	9	4	3	6	8	10	23	21	41	55	50	43	42	20	7	7	33	382
S_{13}	6	4	1	5	9	11	14	26	34	38	57	29	24	16	12	5	27	318
S_{14}	6	6	7	6	9	8	20	13	23	26	22	14	17	6	5	5	12	205
S_{15}	5	3	3	2	6	4	8	12	11	11	10	14	11	9	3	4	16	132
S_{16}	9	6	7	5	4	8	6	5	10	9	5	12	4	5	3	1	14	113
S_{17}	56	10	17	14	12	16	14	13	32	20	22	23	18	8	5	24	79	383

Table 6 : One step conditional transition probability matrix from S_i to S_j

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}	S_{17}	*Row total*
S_1	0,1790	0,0218	0,0306	0,0437	0,0393	0,0742	0,0786	0,0655	0,0873	0,0611	0,0175	0,0524	0,0218	0,0175	0,0349	0,0262	0,1485	1,0000
S_2	0,0864	0,0494	0,0617	0,0247	0,0000	0,0370	0,1111	0,1235	0,1481	0,0370	0,0494	0,0864	0,0370	0,0494	0,0123	0,0247	0,0617	1,0000
S_3	0,0947	0,0211	0,0211	0,0211	0,0947	0,0316	0,0632	0,1158	0,1158	0,0421	0,0947	0,0316	0,0105	0,0737	0,0211	0,0316	0,1158	1,0000
S_4	0,1043	0,0087	0,0087	0,0261	0,0783	0,0957	0,0435	0,0696	0,0957	0,0435	0,0174	0,0696	0,0696	0,0435	0,0609	0,0435	0,1217	1,0000
S_5	0,0377	0,0252	0,0189	0,0189	0,0566	0,0692	0,0881	0,0818	0,0881	0,0881	0,0881	0,0818	0,0314	0,0629	0,0503	0,0126	0,1006	1,0000
S_6	0,0463	0,0185	0,0509	0,0370	0,0509	0,0787	0,0694	0,1065	0,1065	0,0926	0,0926	0,0556	0,0648	0,0417	0,0231	0,0370	0,0278	1,0000
S_7	0,0308	0,0205	0,0205	0,0240	0,0411	0,0616	0,0822	0,1130	0,0856	0,1233	0,0788	0,0959	0,0616	0,0445	0,0445	0,0103	0,0616	1,0000
S_8	0,0207	0,0207	0,0089	0,0414	0,0414	0,0740	0,0917	0,0799	0,1243	0,1213	0,1006	0,0799	0,0562	0,0385	0,0178	0,0178	0,0651	1,0000
S_9	0,0357	0,0168	0,0168	0,0210	0,0294	0,0546	0,0483	0,0840	0,1261	0,1176	0,0945	0,0945	0,0777	0,0504	0,0399	0,0231	0,0693	1,0000
S_{10}	0,0284	0,0081	0,0101	0,0223	0,0223	0,0264	0,0669	0,0629	0,1176	0,1420	0,1318	0,0933	0,0974	0,0629	0,0345	0,0203	0,0527	1,0000
S_{11}	0,0131	0,0066	0,0131	0,0153	0,0284	0,0328	0,0633	0,0808	0,1070	0,1550	0,1572	0,1004	0,0961	0,0459	0,0240	0,0240	0,0371	1,0000
S_{12}	0,0236	0,0105	0,0079	0,0157	0,0209	0,0262	0,0602	0,0550	0,1073	0,1440	0,1309	0,1126	0,1099	0,0524	0,0183	0,0183	0,0864	1,0000
S_{13}	0,0189	0,0126	0,0031	0,0157	0,0283	0,0346	0,0440	0,0818	0,1069	0,1195	0,1792	0,0912	0,0755	0,0503	0,0377	0,0157	0,0849	1,0000
S_{14}	0,0293	0,0293	0,0341	0,0293	0,0439	0,0390	0,0976	0,0634	0,1122	0,1268	0,1073	0,0683	0,0829	0,0293	0,0244	0,0244	0,0585	1,0000
S_{15}	0,0379	0,0227	0,0227	0,0152	0,0455	0,0303	0,0606	0,0909	0,0833	0,0833	0,0758	0,1061	0,0833	0,0682	0,0227	0,0303	0,1212	1,0000
S_{16}	0,0796	0,0531	0,0619	0,0442	0,0354	0,0708	0,0531	0,0442	0,0885	0,0796	0,0442	0,1062	0,0354	0,0442	0,0265	0,0088	0,1239	1,0000
S_{17}	0,1462	0,0261	0,0444	0,0366	0,0313	0,0418	0,0366	0,0339	0,0836	0,0522	0,0574	0,0601	0,0470	0,0209	0,0131	0,0627	0,2063	1,0000