Applicability of Machine Learning Tools in Foreign Exchange Market

*[1]Basav Roychoudhury, [2]Sharad Nath Bhattacharya, [3] Mousumi Bhattacharya
*[1], *Indian Institute of Management Shillong, India, brc@iimshillong.in*
[2], *Indian Institute of Management Shillong, India, snb@iimshillong.in*
[3], *Indian Institute of Management Shillong, India, msb@iimshillong.in*

Abstract

Exploring and understanding the characteristic features of foreign exchange market is appealing for derivative market participants, risk managers and asset allocation decisions makers, whose interest is to reasonably forecast currency movements. In this paper, the authors studied the applicability of machine learning algorithms in predicting currency spot prices, and tried to understand the usability of historical prices in such models. The study was done in Indian context, using the prices of four currencies traded in Indian forex markets. The machine learning algorithms used in this paper were Artificial Neural Networks and Random Forests, the latter providing better predictive performance across all the experiments conducted. The results obtained are also an improvement over those quoted in the literature. These two algorithms were chosen due to the fact that both these do not pre-suppose any particular relationship, in the form of linear, exponential, etc., between the independent and dependent variables, and thereby can model a much complex relationship, if needed. This paper also presents an empirical study on the number of historical currency price data to be included in the model for best results, given the overall historical trend in prices of the concerned currency.

Keywords*: Machine Learning, Data Mining, Forecasting*

1. Introduction

Prediction of forex movements is always challenging since forex rates often show nonlinear dynamics and are often noisy, non-stationary and deterministically chaotic. It is often argued that non-linearity of forex data cannot be efficiently predicted by available statistical models [1]. The conventional econometric models using time series data generally show lower levels of accuracy for forecasting noisy financial data. The Artificial Neural Networks (ANN), due to its unique non-parametric, non-assumable, noise-tolerant and adaptive properties, may map any nonlinear function without a priori assumptions but can also estimate Bayesian a posteriori probabilities when given an appropriately formulated problem [2]. It has shown great applicability in time-series analysis and forecasting due to its pattern recognition capability. At the same time, Random Forest (RF) [3] has also been accepted as an effective machine learning algorithm providing good regression and classification performance. While ANN is susceptible to over-fitting to training data, RF is known to be immune to it.

In this paper, we have evaluated and compared two machine learning algorithms - ANN and RF – for predicting the currency spot price of $(n+1)^{th}$ day based on the spot price of previous n days. The motivation of the work stems from inconclusive findings on the forex forecasting issues which needs further comprehensive investigation. This is especially useful for hedgers who hedges their forex exposure by taking long or short positions in forex contracts.

In our study, we developed models with different values of n. This was done in order to understand and estimate the effect of including different amount of historical data in the models in their predictive performance. We considered these two algorithms (ANN and RF), since both of these do not pre-suppose any particular relationship – linear, exponential, etc. – between the independent and dependent variables. We also looked into the overall historical trend of the concerned currency prices and tried to relate the same in terms of the predictive performance of the aforesaid models.

This paper differs and improves the existing literature in the sense that it comes up with models providing better predictive performance, but more importantly, it provides an empirical study in terms of the amount of historical data to be included in the model for better predictive performance, given the overall historical trend of the concerned currency. In other words, this paper presents a rule that would allow one to look into the historical trend of a given currency prices and decide on the number of previous days (historical) data to be included in the model for better predictive performance. This can be considered to be the major contribution of this paper.

The rest of the paper is organized as follows: section 2 provides a brief overview of current literature in the area, section 3 explains the machine learning tools used in this paper as well as the measure used for comparison of models, the methodology followed is detailed in section 4, section 5 provides the results obtained and discussion thereon, and section 6 concludes the paper.

2. Literature Review

The computational intelligence based techniques not only provide predictions that are more robust than conventional methods, but also explain how each predictor variable impacts the prediction. White [4] possibly inspired application of artificial intelligence in finance when he compared of neural network's performance on stock market predictions with statistical multiple regressions. The ANN models have been proven successful in many situations like forecasting international equity prices [5], forecasting returns of stocks and corporate bonds [6] and also forecasting exchange rates [7]. Feed forward networks was used successfully to estimate a pricing formula for options with good out of sample pricing and delta hedging performance [8]. Overall, machine learning tools showed good application for nonlinear time series forecasting [9]. Application of ANN for forex predictions have concentrated mainly for developed countries [10][11][12]. Theofilatos et al. [13] applied five learning classification techniques (K-Nearest Neighbors algorithm, Naïve Bayesian Classifier, Artificial Neural Networks, Support Vector Machines and Random Forests) and observed that techniques like Support Vector Machines and Random Forests clearly outperformed all other strategies in terms of annualized return and sharp ratio. Qin et al. [14] applied random forest method (Gradient Boosted Random Forest) as a nonlinear trading model to the stock market return of Singapore stock exchange and suggested that the proposed trading methods outperformed buy and hold strategy for similar period. A hybrid models using chaos, neural network and particle swarm optimization were used and were observed to have superior power over other ANN models [15]. However [16] concluded that recurrent Cartesian Genetic Programming-based artificial neural networks is the best option for time series data prediction. The present study aims to conduct a comprehensive research in Indian context with a view to cut out a clearer perspective on the predictability of currencies using machine learning. The motivation for the paper stems from inconclusive findings on the performances of machine learning tools in predicting currencies especially in the developing world.

3. Machine Learning Tools

3.1. Multi-layer Perceptrons (MLP)

The simplest and the most commonly used ANN model is the Multi-layer Perceptrons or the Multilayer Feed Forward Network, where the training of the network is achieved through backpropagation of errors.

The underlying structure here is a directed graph consisting of vertices (called neurons) and directed edges (called synapses) resulting in the network. The neurons are arranged in multiple layers and are connected to each other.

There will be an input layer consisting of one neuron for every input provided to the network. In our models, the input layer will consist of *n* inputs; the currency prices of preceding *n* number of days which is used to predict the price on the *(n+1)*th day. There will also be an output layer consisting one neuron for every output expected from the network. The output layer here will have only one neuron providing the predicted price of the concerned currency. Between these input and output layers, one can have zero or more number of hidden layers based on the configuration of the network– with each layer having one or more neurons.

The output of one layer is the input to the next layer. A weight is associated with each synapse and a bias is linked with each neuron. The weights and biases have their values in the range 0 to 1. In addition, an activation function is also decided as a parameter for the model. The *n* inputs provided to the neurons in the input layer passes to the neurons in the next layer through the synapses. As they travel, their values are modified by the weights and the biases of the components in their path, as well as the activation function at each neuron. Thus, the output of a neuron has contributions from each of the input neurons.

This replicates the process of crossover of genes in genetic evolution of any species. In general, output of node *j* will be

$$output_j = g\left(\theta_j + \sum_{i=1}^{p} w_{ij} x_i\right) \qquad (1)$$

where *g(s)* is the activation function, θ_j is the bias at neuron *j*, w_{ij} is the weight at the synapse connecting neuron *i* with neuron *j*, and x_i is the input at neuron *i*, and *p* is the number of neurons which provide input to neuron *j* (Figure 1). The activation function can be a linear function, an exponential function, or a logistic function; the latter being the most commonly used and the one used here.

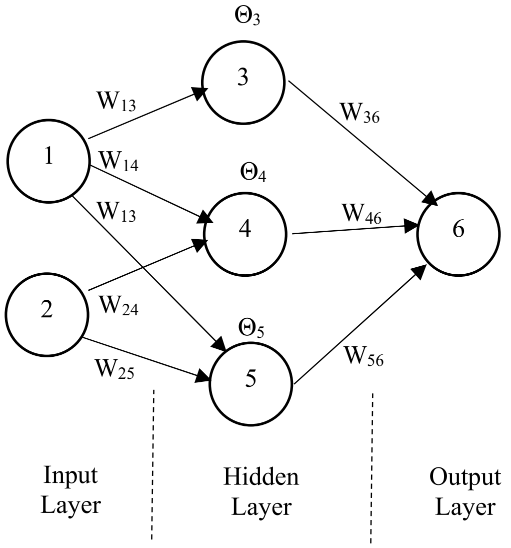

Figure 1: Structure of Multi-Layer Perceptron (MLP)

Before the network is used for prediction, it needs to be trained using historical data where the value of the outcome variable is known for a given set of predictor values. The training involves multiple iterations, where the network learns from each record of the historical data. The network starts with some arbitrary values for the weights and biases. The input values (predictors) get modified by the weights, biases and the activation function as it traverses the network. The output computed at the output layer is then compared with the observed value, and the error is back propagated to adjust the values of the various weights and biases. This process continues till the change in the weights and biases are less than a threshold value. Once this is achieved, the network is said to be *trained.*

The increase in the number of hidden layers and the corresponding neurons increases the complexity of the network. A more complex network is better suited to express a very complex relationship between the predictor variables and the outcome variable. However, a complex network may also take the noise present in any real life data and include it in the relationship. This will result in over fitting of the model on the training data, resulting in poor predictive performance. Thus, the model generation needs to balance between the accuracy of the result and complexity of the network. The decision about the number of layers and the number of neurons in each of them is usually a trial and error method, especially in the MLP model.

3.2. Random Forest

Random Forest is an ensemble method, whereby a combination of Classification and Regression Trees (CARTs) are used; with the individual outputs from each of the CARTs finally combined to generate the output for the Random Forest. The results are combined by a method of voting for classification, and by a method of averaging the individual outputs in case of regression to arrive at the final result, the latter being the one used in our models.

Each of the CARTs in Random Forest are grown randomly from the training dataset provided to train the Forest. The individual trees are grown using different training sets. A random vector Θ_k is generated to grow a tree from the training set provided to train the Random Forest. Θ_k is independent of past random vectors Θ_1, Θ_2, …, Θ_{k-1}, but follow the same distribution. The training sets used to develop the various trees are derived by randomly drawing the records, with replacement, using the random vector Θ_k, from the training set originally provided for the Random Forest. A new tree is grown with each of these new training sets using random feature selection. These trees are allowed to grow without pruning. Each individual tree is thus a classifier or regressor of the form $\{h(x, \Theta_k), k = 1,2,..\}$.

It has been shown that for a large number of trees, because of the law of large numbers, Random Forest does not overfit, instead, produce a limiting value of generalized error.

We considered Random Forest for prediction of the currency prices since it is quite robust against outliers and noise, and time series data on currency prices do tend to have such outliers and noises present.

3.3. Comparison of models

For comparing the models, we used Mean Average Percentage Error (MAPE) as our measure. MAPE is an accuracy measure to evaluate a regression model and gives a percentage score of how the predicted values deviate, on average, from the observed values, a smaller value indicating a better model:

$$= \frac{1}{p}\sum_{i=1}^{p}\frac{|y_i - \hat{y}_i|}{y_i} \times 100\% \qquad (2)$$

where

p is the total number of cases predicted by the model
y_i is the observed value of the outcome variable of the i^{th} case
$\hat{y}_i$ is the predicted value of the outcome variable of the i^{th} case

4. Methodology

We considered currency prices of 1000 days from December 2010 to February, 2015. Four different currency prices – US Dollar (USD), Euro, Pound Sterling (PS), and Japanese Yen (JY) – vis-à-vis Indian Rupees were considered. These four currencies were selected because in Indian market, derivatives contracts are available only on these four currencies and any impact of derivatives trading on the chosen currencies are taken into consideration.

The time series data on currency prices was first formatted into *(n+1)* column data, where *n* is the number of previous days' prices considered to predict the price of $(n+1)^{th}$ day. In our experiments, we considered twelve distinct scenarios with the values of *n* set at $3 \leq n \leq 14$. The idea was to evaluate the prediction efficiency when a less, moderate and more number of previous days' prices were considered to predict the price for a given day.

As both ANN and RF models are trained through supervised learning, the dataset was partitioned into two partitions - training and validation. The training partition consisted of about 70% of the records picked at random without replacement, the rest going to the validation partition.

As we were using the data for 1000 days, once we re-format the data into *(n+1)* columns as mentioned above, we were left with different number of *(n+1)* column records for different values of *n*. For example, we had 997 records for n=3, 996 records for n=4, and so on. To ensure that all our models were trained and validated on same number of records, we first picked up 985 records at random without replacement in each case, before splitting these 985 records into training and validation partitions.

For each of the twelve scenarios mentioned earlier, we ensured that the training and validation partition contained the same set of records in all experiments involving a particular scenario. In other words, if a given set of prices of *(n+1)* consecutive days appeared in the training partition in case of a model with a particular currency for a given value of *n*, the said prices on those days will always appear in the training partition for all the models involving the same value of *n*, for all the other three considered currencies. This was done to ensure comparability of results across the experiments for a particular value of *n* for all currencies.

We used the statistical package R for all our experiments. The experiments on ANN were done using the RSNNS library [17] and those on RF were done using the Random Forest library [18]. We also used

R for generating the training and validation partition, and had kept the same value of seed for picking up records at random across the experiments.

As ANN works best with inputs and outputs in the range 0 to 1, we scaled the data to that interval while using MLP models. The corresponding output, while using the model for predicting the prices, was converted back to the original scale for comparison with the observed values. For RF based models, no such scaling and re-conversion back to the original was involved.

The configuration of the MLP and RF used were set as:

- MLP – we had used the MLP with one hidden layer of 10 nodes
- RF - we kept the default value of 500 trees for generating the forest

5. Results and Discussion

The spot prices and their variations across the aforesaid 1000 days are plotted in figure 2.

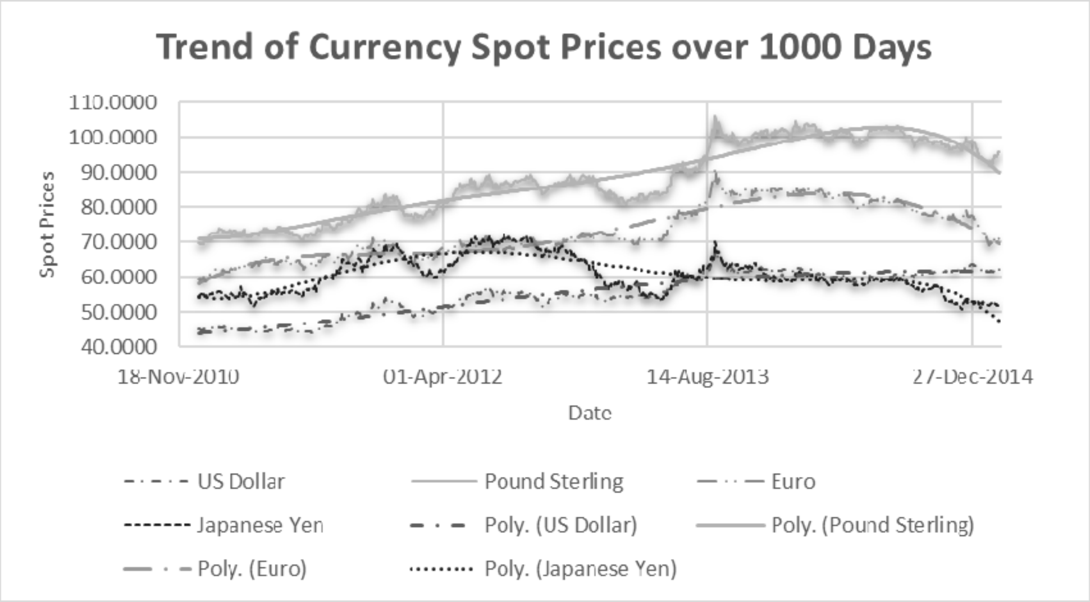

Figure 2: Currency Spot Prices from Dec 27, 2010 till Feb 16, 2015

Figure 2 also contains the degree five polynomial trend line for each of the four currency considered. Using polynomial of degree five allowed the trend line to exhibit short to medium term fluctuations apart from showing the overall picture. The currencies Pound Sterling and Euro are observed to exhibit a similar trend with an almost linear increase for most of the part followed by a dip. The trend for US Dollar is a slow increase over the period. The most turbulent currency has been the Japanese Yen over this time, with the trend showing an increase followed by a decrease, a plateau of stabilization, and finally a quite sharp dip.

With these data, we developed and trained models to predict the price on *(n+1)*th day, using the actual prices of immediate previous *n* days. Thus for *n=3*, we used the actual prices of day 1, day 2 and day 3, to predict the price on day 4. We thereby developed two models for each value of *n* - one using ANN, and the other using RF. Each of these models were then used for the four currencies considered – four currencies with twelve values of *n* as mentioned earlier – resulting in a total of forty eight regression models. These models were trained using the data in the training dataset, the same set of records used for corresponding ANN and RF models for each of the cases. We then computed the MAPE based on the price predicted by each of these models for records in the validation dataset. Table 1 summarizes the results in terms of MAPE we obtained for each of the cases.

Table 1. Prediction efficiency of Models based on MAPE (%)

Previous Days (*n*)	Euro		Japanese Yen		Pound Sterling		US Dollar	
	ANN	RF	ANN	RF	ANN	RF	ANN	RF
3	0.4975	0.2341	0.6516	0.2986	0.4938	0.2328	0.4853	0.2161
4	0.4781	0.2423	0.7195	0.3134	0.4893	0.2411	0.4431	0.1993
5	0.4994	0.2473	0.6819	0.3287	0.4958	0.2487	0.4581	0.2170
6	0.4920	0.2235	0.6944	0.3028	0.4746	0.2238	0.4732	0.2067
7	0.4663	0.2187	0.6382	0.2902	0.4687	0.2191	0.4280	0.1959
8	0.4892	0.2295	0.6938	0.3010	0.4751	0.2278	0.5118	0.2021
9	0.4794	0.2194	0.6855	0.2975	0.4839	0.2169	0.4619	0.2087
10	0.4900	0.2169	0.6872	0.2844	0.4908	0.2159	0.4759	0.2045
11	0.4505	0.2126	0.6879	0.2885	0.4499	0.2154	0.4616	0.1970
12	0.5006	0.2168	0.6344	0.2824	0.4846	0.2197	0.4732	0.2005
13	0.5122	0.2329	0.6266	0.2863	0.5068	0.2323	0.4567	0.1931
14	0.5263	0.2387	0.6869	0.2907	0.5272	0.2382	0.5153	0.1995

For easy visualization, the values of Table 1 are plotted in Figures 3 (for ANN) and 4 (for RF).

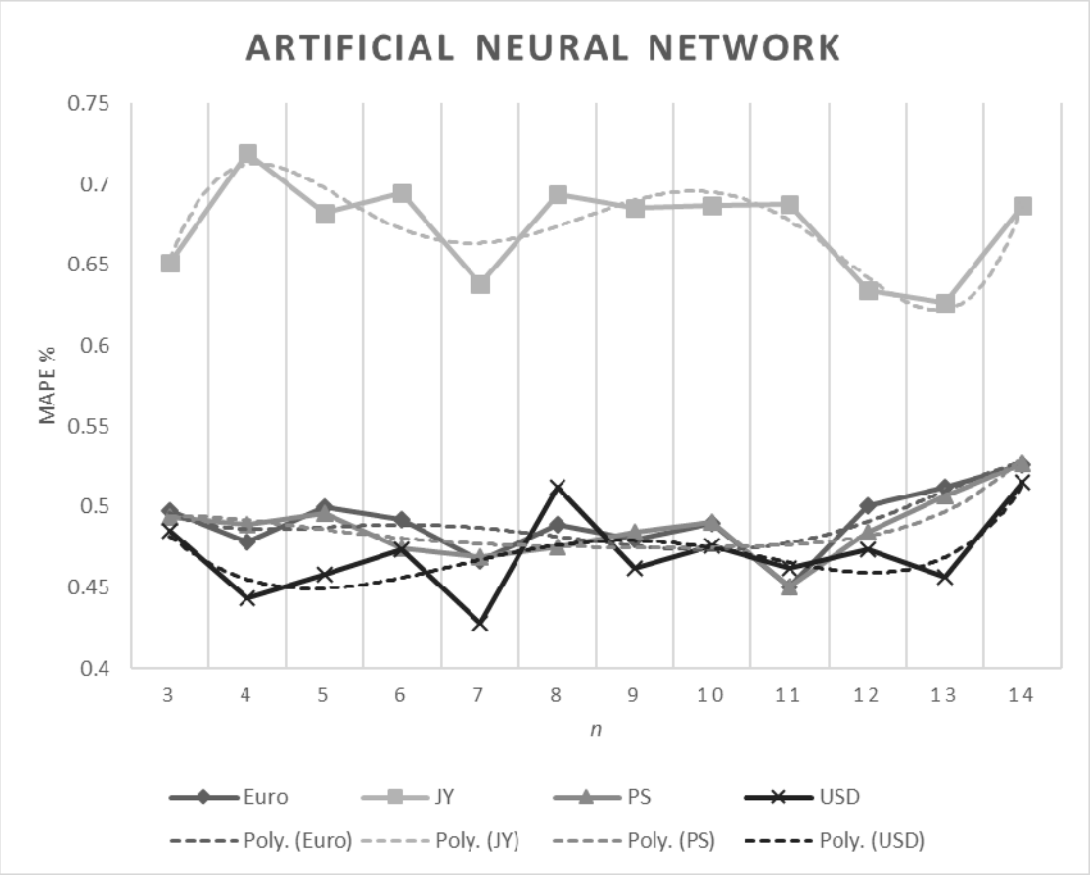

Figure 3: MAPE of the various ANN models along with trend line

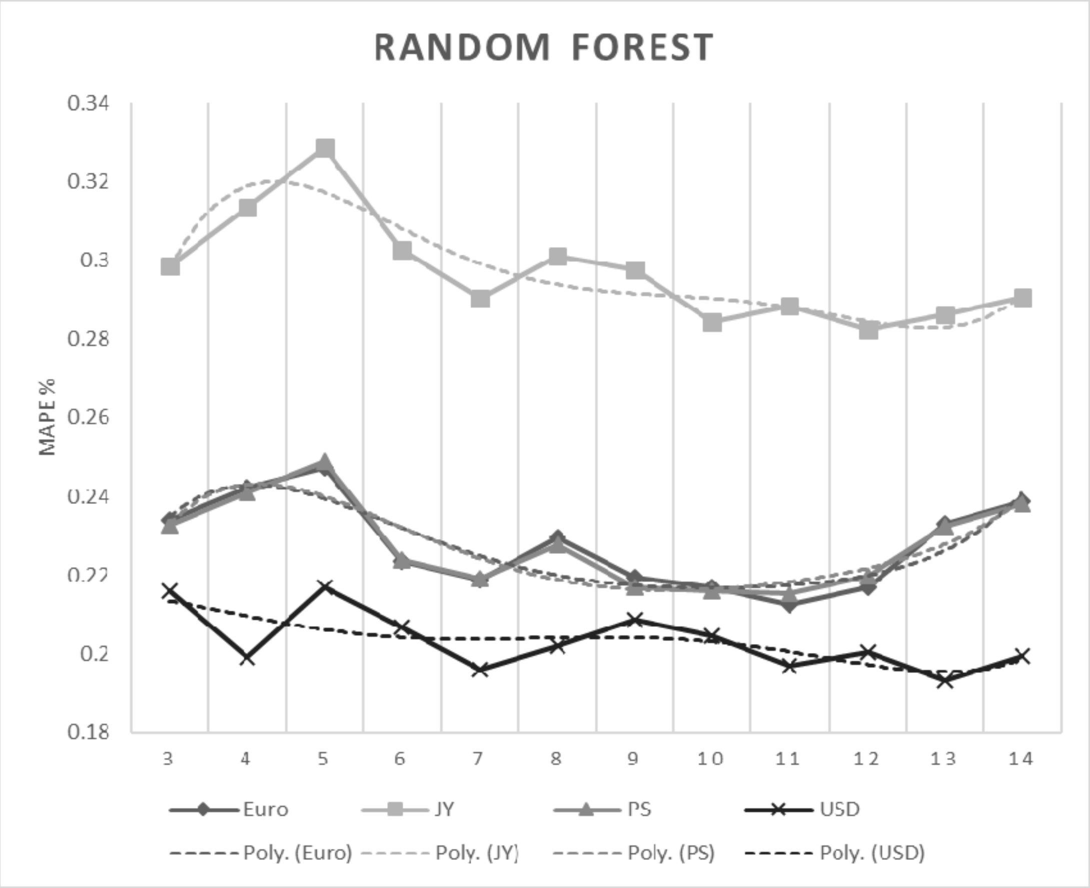

Figure 4: MAPE of the various RF models along with trend line

As seen from Table 1, the RF models are seen to be considerably better across all cases as compared to the ANN models. The RF models' maximum MAPE of 0.328% (mean: 0.238%, median: 0.223%) is an improvement over earlier empirical studies. Even the performance provided by ANN models with maximum MAPE of 0.719% (mean: 0.532%, median: 0.494%) are satisfactory when compared to existing literature.

However, a more interesting outcome and the major contribution of this paper is seen from figures 3 and 4. On observing the MAPE values and the trend line (degree five polynomial), there are clear trends as far as Pound Sterling, Euro and Japanese Yen is considered. The same is not so definite in case of US Dollar, as the trends are little different in the two figures. Following are the observations from these figures:

1. Pound Sterling and Euro shows very similar trends as far as the MAPE by various models are concerned. These currencies had also shown similar trend in prices over the 1000 days considered (figure 2). The models using moderate value of *n* (in the range 9 to 11) provides the lowest MAPE for these currencies.
2. For Japanese Yen, the models using high value of *n* (near 13) provides the lowest MAPE.
3. For US Dollar, the ANN model shows a clear trend and provides lowest MAPE with models which use small value of *n* (around 4 to 7). The RF model fail to show clear trend, but provides very small MAPE values across the models.

While one would like to study this trend further with other markets beyond India, the current result allows one to infer, based on the trend of change in the spot prices over 1000 days as seen in figure 2, the following:

1. For smooth trend in currency prices, which do not include too many ups and downs, one may use moderate values of *n* for better prediction of spot price of *(n+1)*th day..

2. For trends showing irregularity with quite a few ups and downs, one may use large value of *n* for the prediction. However, the prediction performance in such cases or irregular trend will be inferior when compared to other cases.
3. For trends which are almost linear, with no major fluctuations, one may prefer small values of *n* to have better prediction performance.

6. Conclusions

The findings show that in terms of MAPE, RF performs significantly better in predicting the future currency prices compared to ANN. Also the possibility of a relationship between trends of currency prices with the number of days used in the model for prediction cannot be ruled out. The findings are cross-validated across all four currencies. Using these tools may be computationally expensive, but it has the advantage that the model complexity and the number of observations needed for the in-sample estimation are determined optimally which prevents over fitting in noisy environments. The findings may be used by currency traders as an additional tools to obtain better prediction results and similar users can design an appropriate hedging strategy.

7. References

[1] Philip, A. A., Taofiki, A. A., Bidemi, A. A., “Artificial Neural Network Model for Forecasting Foreign Exchange Rate”, World of Computer Science and Information Technology Journal (WCSIT), vol. 1, no. 3,pp.110-118, 2011

[2] Ruck, D.W., Rogers, S.K. & Kabrisky, M, “Feature Selection Using a Multilayer Perceptron”, Journal of Neural Network Computing, vol. 2, no. 2, pp. 40-48, 1990

[3] Breiman, L., “Random Forests”, Machine Learning, vol. 45, no. 1, pp.5–32,2001

[4] White H, “Economic prediction using neural networks: The case of IBM daily stock returns”, IEEE International Conference on Neural networks, vol. 2, no. 1, pp. 451–458, 1988

[5] Cogger, Kenneth O, Paul D. Koch, and Diane M. Lander, “A Neural Network Approach to Forecasting Volatile International Equity Markets”, Advances in Financial Economics, vol. 1, no. 3, pp.117-57, 1997

[6] Desai,V. S. and R. Bharati, “The Efficacy of Neural Networks in Predicting Returns on Stock and Bond Indices”, Decision Sciences, vol. 29, no. 2, pp. 405-424, 1998

[7] Hu, M.Y., Zhang, G., Jiang, C.X., Patuwo, B.E., “A cross validation analysis of neural network out-of –sample performance in exchange rate forecasting”, Decision Sciences, vol. 30, no. 1, pp.197-216, 1999

[8] Garcia, R. and Gen ̧cay, R., “Pricing and hedging derivative securities with neural networks and a homogeneity hint”, Journal of Econometrics, vol. 94, no. 1-2, pp. 93–115, 2000

[9] Shin, T., Han, I., “Optimal signal multi-resolution by genetic algorithms to support artificial neural networks for exchange-rate forecasting”, Expert Systems with Applications, vol. 18, no. 4, pp. 257-269, 2000

[10] Kamruzzaman, J., Sarker R. and Begg, R., “Modelling and Prediction of Foreign Currency Exchange Markets”, Artificial Neural Networks in Finance and Manufacturing, Idea Group Publishing (USA), 139-150, , 2006

[11] Naeini, M. P., Taremian, H., and Hashemi, H. B., “Stock market value prediction using neural networks”, Computer Information Systems and Industrial Management Applications (CISIM), 2010 Int. Conf. on, pp. 132-136, 2006

[12] Wei, L. Y. and Cheng, C. H., “A Hybrid Recurrent neural networks model based on synthesis features to forecast the Taiwan stock market,” International Journal of Innovative Computing Information and Control, vol. 8, no. 8, pp.5559-5571, 2012

[13] Theofilato, K., Likothanassis, S. and Karathanasopoulos, A., “ Modeling and Trading the EUR/USD Exchange Rate Using Machine Learning Techniques”, ETASR - Engineering, Technology & Applied Science Research, vol. 2, no. 5, pp. 269-272, 2012

[14] Qin Qin, Qing-Guo Wang, Jin Li, Shuzhi Sam Ge, “.Linear and Nonlinear Trading Models with Gradient Boosted Random Forests and Application to Singapore Stock Market”, Journal of Intelligent Learning systems and Applications, vol. 5, no. 1, pp. 1-10, 2013

[15] Dadabada Pradeepkumar, Vadlamani Ravi, "FOREX Rate Prediction Using Chaos, Neural Network and Particle Swarm Optimization", Advances in Swarm Intelligence Lecture Notes in Computer Science, vol 8795, pp 363-375, 2014

[16] Faheem Zafari , Gul Muhammad Khan , Mehreen Rehman , Sahibzada Ali Mahmud, "Evolving Recurrent Neural Network using Cartesian Genetic Programming to Predict The Trend in Foreign Currency Exchange Rates", Applied Artificial Intelligence, vol. 28, no. 6, 2014

[17] Bergmeir, C., Benitez, J.M., “Neural Networks in R Using the Stuttgart Neural Network Simulator: RSNNS”, Journal of Statistical Software, vol. 46, no. 7, pp.1-26, 2012

[18] Liaw, A., Wiener, M., “Classification and Regression by randomForest”, R News, vol. 2, no. 3, pp. 18-22, 2002