Time Series Prediction of Population using LSTM

Ankit Bisht, Prakash Kantheti, Nisarg Chitaliya, Mukul Dang

Abstract— **Prediction has been one of the most sought ability of the 21st century. Harnessing the data of the past and obtaining actionable results for the future is important as it gives the ability to foresee the future and prepare for its challenges. One such time series phenomenon that is of great importance is the prediction of population of a country, especially given the alarming rate of population expansion in the world. While significant amount of work has been done in the past in forecasting the population of a given country, limited work has been done in harnessing neural networks for forecasting population of the future. This work makes use of LSTMs which have been proved to be promising for such forecasting purposes and tries to build an efficient tool to deal with this challenge of forecasting. We build different variant models of LSTMs and use those models to predict the population of different countries. We then compare the performances of each of those models on the dataset and discuss their relevant results.**

Keywords: Neural Networks, LSTM, Time Series Prediction

I. INTRODUCTION

Population rise is one of the largest challenges the world is fighting with today. In a world where the resources for the mankind are depleting, population is surging and there is a dearth of sustainable solutions for meeting the needs of the population, we consider that population raise is a serious concern. While we know the population statistics today, it becomes important that we understand the future trend of the population so that we are prepared for the challenges it offers. This work is in lines with this objective to develop an efficient tool towards forecasting the population in the future by using the data of the past.

Several tools are in existence and are aimed at addressing this problem of forecasting population. While each of these tools follow different methodologies to predict the future, they have their own pros and cons. Forecasting has been one of the most sought ability today as it gives us the power to foresee the circumstances. Be it stock market, business holdings of a firm, climatic conditions at a place or the behaviour of the consumer market, prediction has an application in almost every field. Several techniques are being used in each of these fields and each of them has its own advantages and limitations.

In this work we use using Neural Networks which have proved to be providing promising results in such prediction related activities. One such technique of Neural Networks which have received little attention towards forecasting population are LSTMs. LSTMs are one of the popular techniques that have earlier been used in a lot of forecasting applications and have been found to be delivering efficient results.

For the forecasting in this study, we have used the population data of three countries China, United Kingdom and United States. The dataset includes the population values of each of these countries for each year from 1820 to 2008 provided by [3]. This data has been used to train the LSTM model and further predict the population of these countries. We also consider 5 different variants of LSTMs which have different architectures and approaches towards the prediction problem. The details of each of these LSTM models have been discussed in the Design and Implementation section of this work.

The results obtained with each of the models used in this work has been discussed later in the Results and Discussion section. In general , the LSTM model gave an efficient prediction on the test case and proved to be a reliable approach for forecasting population. While there was a ambiguity in the results of prediction in some of the models the reasons for the low accuracy in those models has also been discussed in the results section.

II. BACKGROUND AND RELATED WORK

This section discusses different approaches for time series prediction that have been reported in the past and have been proven to be effective.

Elman (1990) mentioned in his paper a Time Window Approach in which static pattern matching device with a fixed time window of recent inputs serve as temporal sequence processing. The problem is that it is difficult to determine the required window size and also sometimes a long window is required. Although we can use multiple windows but it is useful only when long term dependencies of task are known. In this each unit feeds into every other via time delayed connections. Elman nets are trained by backpropagation and therefore do not propagate errors.

Time delay neural networks (TDNNs) (Haffner Waibel, 1992) allow access to past events via cascaded internal delay lines. Since the interval which could be accessed depends on network topology, resulting into time window problem.[16][17]

In (Kondratenko, 2003) the authors presents a prediction method for exchange rates between American Dollar and Japanese Yen, Swiss Frank, British Pound and EURO using a recurrent neural network. Before putting the actual data series, some pre processing was done based on normalization, calculation of Hurst exponent, Kolmogorov-Smirnov test in order to remove the possible correlations. In Elman-Jordan recurrent network, the forecast of a one day ahead value of moving average returning the window equal to 5 observations was calculated. The number of hidden neurons was chosen equal to 100 with a linear activation of the input layer and logistic activation of the hidden and output layer. The neural network predicted the increments sign with a probability of approximately 80 percent.[18]

Chen (2003) compared the performance of a Probabilistic Neural Network with a GMM-Kalman Filter and random walk approach in order to predict the direction of return on market index of the Taiwan Stock Exchange. It was concluded that PNN has a stronger forecasting power than both the GMMKalman filter and the random walk models as PNN has a better capability to identify erroneous data and outliers. PNN also doesnt require any apriori information about the underlying probability density functions of the data.[19]

Philip (2011) showed in paper the comparison between a neural network and a Hidden Markov Model used for foreign exchange forecasting is also given in. The results of the study show that while the Hidden Markov Model achieved an accuracy of 69.9 percent the neural network had an accuracy of 81.2 percent.[20]

ARIMA-PNN presented (Khashei, 2012) a hybrid model. The values estimated with the ARIMA model are changed based on the trend of the ARIMA residuals detected by a PNN and optimum step length obtained a mathematical programming model.[21]

(Proietti, 2013) used Data transformation in order to improve the accuracy of the forecast. Considering the Box-Cox power transformation, author showed how forecasts are improved significantly compared to the un-transformed data at the one-step-ahead horizon.[22]

(Man-Chung, 2000) used a conjugate gradient learning algorithm with a restart procedure to improve the convergence. Author does not use classical random initialization instead used a multiple linear regression for weight initialization. The network was used to predict the daily trading data of some companies from Shanghai Stock Exchange.[16][17][23]

III. Long Short Term Memory (LSTM)

Traditional neural networks can not handle the dependencies in the data well, meaning that the traditional networks cannot use their reasoning about previous events in the data to inform the later ones. Recurrent neural networks have addressed the issue by containing a network with loops in them and enabling information to persist. In a recurrent neural network, a chunk of network looks at some input and outputs a value. The loop allows information to be passed from one step of network to the next. A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to the successor. A typical unrolled loop of a neural network looks as below.

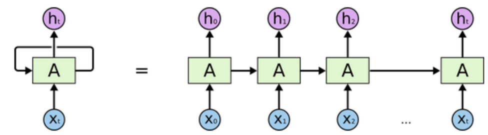

Fig. 1. Unrolled recurrent neural network

This chain like nature reveals that recurrent neural networks are intimately related to sequences and lists. They are the natural architecture of neural network to use for such data. In the last few years, there have been incredible success applying RNNs to a variety of problems: speech recognition, language modelling, translation, image recognition etc. Essential to the success of the use of Recurrent neural networks is the existence of LSTMs, a very special kind of recurrent neural network which works for many tasks, much better than the standard version. This work on predicting time series population data will explore the power of LSTMs.

Long Short Term Memory Networks usually called LSTMs are a special kind of RNN, capable of learning long term dependencies, introduced by Hochreiter Schmidhuber (1997). A typical LSTM containing 4 layers is as shown below.

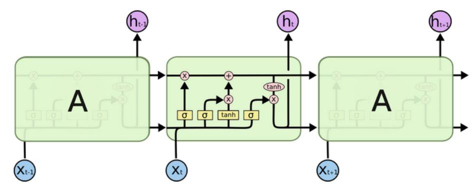

Fig. 2. Repeating Module in an LSTM containing four interacting layers

These LSTMs work tremendously well on a large variety of problems and are now widely used. Remembering information for long periods of time is one of their default and most prominent behaviour. LSTMs have a chain like structure where the repeating module has a different structure from that of a normal RNN. In LSTMs, instead of having a single neural network layer, there are four layers interacting in a very special way.

LSTM is trained using Backpropagation through time and it overcomes the vanishing gradient problem. It can be used as such to address difficult sequence problems in machine learning and achieve state-of-the-art results.

IV. Design and Implementation

A. Approaches

This work harnesses the power of LSTMs to predict the Population changes of different countries in the world. We can phrase the problem as a regression problem i.e. Given the population of a Country in a given year, what is the population of the country in the next year? To solve this problem, we use different variants of LSTM on the same dataset and understand the performance of each of the models on our dataset. The different model architectures used in this work include

1) LSTM network for Regression
2) LSTM for regression using the window method.
3) LSTM for regression with Timesteps.
4) LSTM with Memory between batches.
5) Stacked LSTMs with Memory Between Batches.

The different architectures in each of these models and the techniques used are discussed below.

1) **Long Short Term Memory for Regression**
This method considers the problem as a simple regression problem i.e. Given the population of a country in a given year, what is the population in the next year? To deal with this problem, this method creates a function which converts the data into a pattern where it creates two columns of the same data. The first column containing this years (t) population and the second column containing next years (t+1) population count, to be predicted. The training of this method is done using the following methodology. The model looks at the population of a given year and trains the model to find the parameter. The model makes a prediction of population for the next year and based on the actual population values, it updates the parameters of the model. This process is repeated for the whole training data and this builds our Model of LSTM based on regression.
2) **Long Short Term Memory for Regression with Window Method**
In this method, the problem is framed such that multiple recent time steps are used to make the prediction for the next time step. Otherwise stated, the problem is as follows: Given the population of a country in a year (t) and we want to find the population of the next year (t+1), we use the current population (t) as well as two prior year populations (t-1 and t-2). These earlier year populations that are used in this problem is called a window, and the size of the window is a parameter that can be tuned for each problem.
3) **Long Short Term Memory for Regression with Time Steps**
Our data for the LSTM network includes time steps. Time steps provide another way to phrase our time series problem, Like above in the window method, we can take prior time steps in our time series as inputs to predict the output at the next time step. Instead of phrasing the past observations as separate input feature, we can use them as time steps of the one input feature, which is indeed a more accurate framing of the problem. This method is implemented by using the same data representation as in the previous window example, except when we reshape the data, we set the columns to be the time step dimension and change the feature dimension back to 1.
4) **Long Short Term Memory with Memory Between Batches**
LSTMs network include memory units in their architecture . This enables them to remember information across long sequences. Usually when fitting the model, the states of the networks are reset immediately after each training batch or epoch. By making the LSTM layer Stateful a finer control could be gained over the internal state of the LSTM network. This implies that the LSTM network and build a state over the entire training sequence and then maintain the same state if needed to make predictions. In this method it is required that the training sequence to not be shuffled when training the network. This is because of the dependencies within the input set, which could be lost if the training set is shuffled. This model also requires an explicit method to reset the states of the network after each exposure to the training data i.e. epoch. This could be done by making a model.resetstates() call. This again implies that we are required to create our own outer loop for epochs and then within each iteration we make a function call to fit the training data to the network and then call the function to reset the states.Finally, after creating the LSTM layer, we have to set the Stateful parameter to true to enable Stateful LSTMs. Also instead of specifying the input dimensions, we had to hardcode the number of samples included in a specific batch, the number of time steps in a particular sample and the number of features in that time step. This is done by setting the batchinputshape parameter.
5) **Stacked Long Short Term Memory with Memory Between batches**
This model can be considered as an extension to the model discussed above . This model provides us an opportunity to look at one of the largest benefits of LSTMs, that is the fact that they could be successfully trained while being stacked into deep network architectures. Keras provides us with the opportunity to stack LSTM networks in the same fashion as any other layer types that can be stacked. There is only on requirement for this architecture, which is that it is required that an LSTM layer prior to each subsequent LSTM layer must always return the sequence. This is achieved by setting the returnsequences parameter on the layers to True.In this approach we have extended the previous section configuration of the LSTM network to accommodate two layers. Here both layers have exact same configuration except for the layer that is prior to the second layer has its parameter returnsequences set to True as discussed above.

B. *Implementation*

Time series prediction problems are a difficult type of predictive modelling problem. Unlike regression predictive modeling, time series additionally adds to the complexity of a sequence dependence among the input variables. The problem we are going to look in this paper is the Population Prediction of various countries. This a problem where, given the country and its previous years' population data, we predict its population for the years that follow. The population data of the countries range from years 1820 - 2008 and the countries considered for the prediction problem are China, USA and United Kingdom. We train the data for 100 epochs and then test the data. We develop LSTM models in Python using the Keras deep learning toolkit to address the time-series prediction problem. Algorithm 1 shows the algorithm pseudocode we used to predict the population using LSTM network. We consider five different LSTM models for this study and each of these models are run on the same dataset to obtain the required results. These five models have different model architectures and methodologies towards the time series prediction and hence result in different results of prediction.

Algorithm 1: LSTM Network

Input : Use .csv file containing the data
Output: Predicted Model

1. Normalize the dataset :
   $scaler \leftarrow MinMaxScaler(feature)$
   $dataset \leftarrow scaler.fitTransform(dataset)$
2. Split it into test and train sets : trainSize = 70 percent of dataset testSize = 30 percent of dataset
3. Create and Train the LSTM Networks : model = sequential()
   $model.add(LSTM(output_shape, inputShape))$
   $model.compile(loss = MSE, optimizer = adam)$
   $model.fit(trainX, trainY, epochs = 100)$
4. Make Predictions :
   $trainPredict \leftarrow model.predict(trainX)$
   $testPredict \leftarrow model.predict(testX)$
5. Invert Predictions :
   $trainPredict \leftarrow performinverseTransform$
   $testPredict \leftarrow performinverseTransform$
6. Calculate MeanSquareError :
   $trainScore \leftarrow MSE(trainY, trainPredict)$
   $testScore \leftarrow MSE(testY, testPredict)$
7. Show Results: plotResults()

V. Results and Discussion

The five different models discussed earlier have been implemented on population data of each of the three countries China, United Kingdom and United States. These approaches were as discussed in above sections. These approaches are mentioned below:

1) LSTM for Regression
2) LSTM for Regression using window method
3) LSTM for regression with Time Steps
4) LSTM with Memory Between Batches
5) Stacked LSTM with Memory Between Batches

The dataset that we have used consists of data for 3 countries, namely China, United Kingdom and the United States of America. Fig 3 gives a time series plot of how the population has varied over the period of 1820 to 2008 for these three countries.

The figures below show our different LSTM model's prediction on the training and testing datasets for each of the different countries considered.

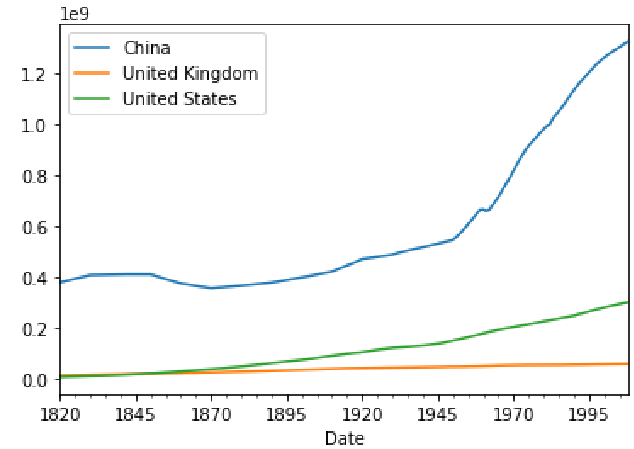

Fig. 3. Population Time Series Graph

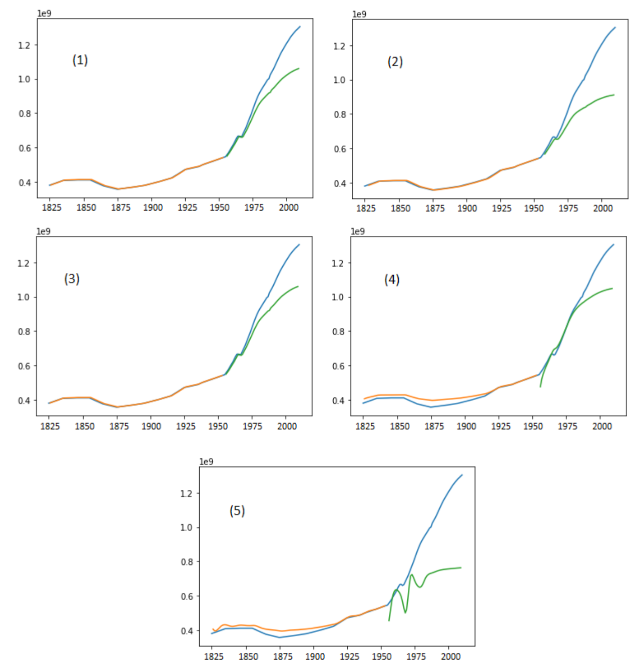

Fig. 4. Prediction Results for China
X- axis : Population in years (1820 - 2008)
Y - axis : Population (in 10^9)
Blue Line: Actual Data, Orange Line: Prediction on Training Data, Green Line: Prediction on Testing Data

Figure 4 shows the results for China. We can see the graph plotted by each approach used and are numbered corresponding to the respective Model numbers.
Figure 5 shows the results for United States of America. We can see the graph plotted by each approach used and are numbered corresponding to the respective Model numbers.

Figure 6 shows the results for United Kingdom. We can see the graph plotted by each approach used and are numbered corresponding to the respective Model numbers.

As observed in each of the images above, for this dataset we got better results using the approaches of Model 1 and 3 i.e. LSTM for Regression and LSTM for Regression with

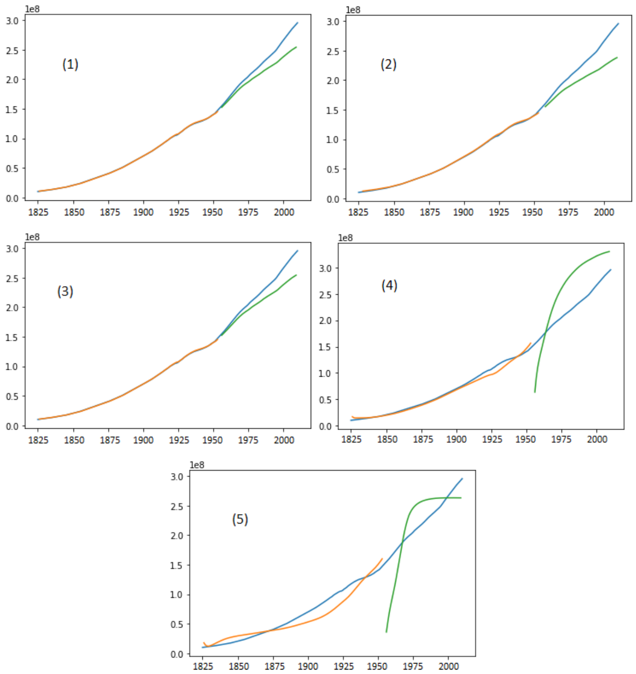

Fig. 5. Prediction Results for USA
X- axis : Population in years (1820 - 2008)
Y - axis : Population (in 10^8)
Blue Line: Actual Data, Orange Line: Prediction on Training Data, Green Line: Prediction on Testing Data

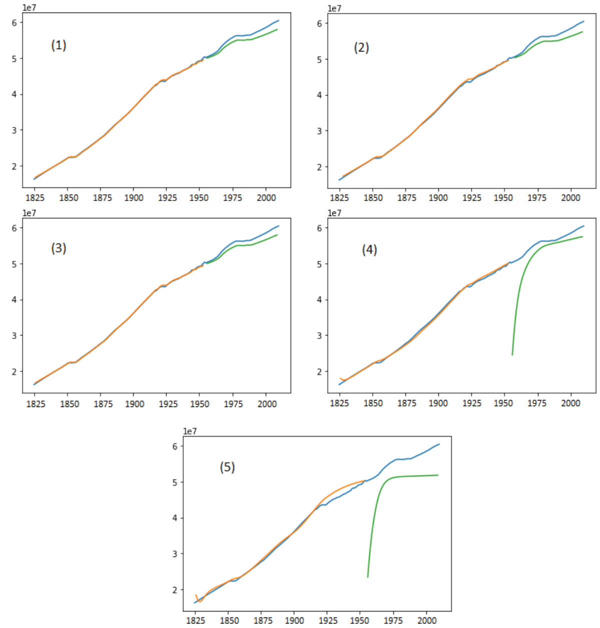

Fig. 6. Prediction Results for United Kingdom
X- axis : Population in years (1820 - 2008)
Y - axis : Population (in 10^7)
Blue Line: Actual Data, Orange Line: Prediction on Training Data, Green Line: Prediction on Testing Data

TABLE I
TRAINING AND TESTING RESULTS
(MEAN SQUARED ERROR IN THOUSANDS)

Country/ Approach	China	United Kingdom	United States
LSTM for-regression	Train: 2.132 Test: 117.98	Train: 0.145928 Test: 1.42	Train: 0.44 Test: 19.11
LSTM-Window Method	Train:2 0.633 Test: 205.56	Train: 0.263 Test: 1.68	Train: 0.89 Test: 29.12
LSTM-TimeStep	Train: 2.13 Test: 117.98	Train: 0.14 Test: 1.42	Train: 0.44 Test: 19.11
Memory Between-Batches	Train: 22.32 Test: 109.32	Train: 0.41 Test: 63.26	Train: 4.97 Test: 54.01
Stacked Memory-Between Batches	Train: 20.48 Test: 298.91	Train: 0.92 Test: 8.33	Train: 3.48 Test: 37.53

TimeSteps. With these approaches we were able to predict the population variations with an accuracy of approximately 70%. The table below shows the mean square error received for training and testing data for each country. Since the data in our data set is in millions prediction results are also comparable and still giving a decent score on the predictions. This however cannot be said for the approaches including Window Method, Memory between Batches and Stacked Memory between Batches. In the other 3 models, it was observed that the model has been over fitting for the training data and hence was performing weakly on the testing data, resulting in a very high error.

VI. CONCLUSION

We have successfully implemented a LSTM model for predicting the population of few countries. The work can now be used to determine how population of each country changes with time and this ability to determine the change becomes important to address the problem of population surge in several countries. We considered different variants of LSTM and obtained results in which two of the models turned to be effective. While this model could predict the population change, it can be improved further by considering several other factors that contribute towards a country's population and also by applying advanced Ensemble Modeling techniques.

REFERENCES

[1] Xiao ding, Yue Zhang, Ting Liu, junwen Duan, Deep Learning for Event-Driven Stock Prediction,Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)
[2] W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, 1993, pp. 123135.
[3] DataMarket - Total Population Datahttps://datamarket.com/
[4] M. J. Beal. Variational algorithms for approximate Bayesian inference. PhD thesis, University of London, 2003
[5] Time Series Prediction http://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/
[6] An-Sing Chen, Mark T. Leung, Hazem Daouk, Application of neural networks to an emerging financial market: forecasting and trading the Taiwan Stock Index, Computers Operations Research 30, pp. 901 923, 2003.
[7] Julian Faraway, Chris Chatfield, Time Series Forecasting with Neural Networks : a Comparative Study using the Airline Data
[8] Torben G Andersen and Tim Bollerslev. Intraday periodicity and volatility persistence in financial markets. Journal of empirical finance, 4(2):115158, 1997.[Andersen and Bollerslev, 1997]
[9] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013 [Mikolov et al., 2013].
[10] Hong Tan, Neural Network for Stock Forecasting, Master of Science in Electrical Engineering Thesis, Faculty of Texas Tech University, 1995.

[11] C. E. Rasmussen. Gaussian processes for machine learning. 2006.
[12] A. McGovern, T. Supinie, I. Gagne, M. Collier, R. Brown, J. Basara, and J. Williams. Understanding severe weather processes through spatiotemporal relational random forests. In 2010 NASA conference on intelligent data understanding, 2010.
[13] C. E. Rasmussen. Gaussian processes for machine learning. 2006.
[14] De Lucia A, United Kingdomse R, Passero I, Tortora G. Development and evaluation of a virtual campus on Second Life: The case of SecondDMI.
[15] L. Chen and X. Lai. Comparison between ARIMA and ANN models used in short-term wind speed forecasting. In Power and Energy Engineering Conference (APPEEC), 2011 Asia-Pacific, pages 14. IEEE, 2011.
[16] Felix Gers. Long Short-Term Memory in Recurrent Neural Networks. Lausanne, EPFL 2001. THESE N 2366 (2001).
[17] Bogdan Oancea. Time Series Forecasting Using Neural Networks. Stefan Cristian Ciucu.
[18] V.Kondratenko, Yu. A Kuperin, Using Recurrent Neural Networks To Forecasting of Forex, Condensed Matter, Statistical Finance, 2003.
[19] An-Sing Chen, Mark T. Leung, Hazem Daouk, Application of neural networks to an emerging financial market: forecasting and trading the Taiwan Stock Index, Computers Operations Research 30, pp. 901923, 2003.
[20] Adewole Adetunji Philip, Akinwale Adio Taofiki, Akintomide Ayo Bidemi, 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.
[21] Mehdi Khashei, Mehdi Bijari, Gholam Ali Raissi Ardali, Hybridization of autoregressive integrated moving average (ARIMA) with probabilistic neural networks (PNNs), Computers Industrial Engineering 63, pp. 3745, 2012.
[22] Tommaso Proietti, Helmut Ltkepohl, Does the BoxCox transformation help in forecasting macroeconomic time series?, International, Journal of Forecasting 29, pp. 8899, 2013.
[23] Man-Chung Chang, Chi-Cheong Wong, Chi-Chung Lam, Financial Time Series Forecasting by Neural Network Using Conjugate Gradient Learning Algorithm and Multiple Linear Regression Weight Initialization, Computing in Economics and Finance 2000(61), Society for Computational Economics, July 2000.