PREDICTION OF FOREIGN DIRECT INVESTMENT: AN APPLICATION OF LONG SHORT-TERM MEMORY

Sourabh Singha Roy
Faculty of Business Administration, St. Theresa International College, Bangkok, Thailand

ABSTRACT

In any country, Foreign Direct Investment(FDI)plays a crucial role in the development process through the transfer of financial resources, innovative management techniques, technology and raising productivity. Developing countries like Indianeeds inflow of foreign capital to fasten economic growth and development. It also helps the policy makers to fine tune their economic policies to attract more FDI and take comparative advantages in the competitive world. However, today's globalized world makes it more complex, volatile and noisy in nature. Therefore, in reality, it is very difficult to predict the future values of FDI. Linear models sometimes can't capture the complex patterns present in the data. It is expected that estimation of FDI through Deep learning method would not only be able to capture such volatility more efficiently but also be able to predict future inflow of FDI more accurately than conventional forecasting techniques. In this paper, we have implemented Long Short Term Memory (LSTM) algorithm to predict the future inflow of FDI. It has been shown that LSTM proves it's supremacy over Autoregressive Integrated Moving Average(ARIMA), Generalized Autoregressive Conditional Heteroscedasticity(GARCH) and other linear forecasting models.

Keywords: FDI, Long Short-Term Memory, time series decomposition, neural network, RNN.
Article Received: 18 October 2020, Revised: 3 November 2020, Accepted: 24 December 2020

INTRODUCTION:

Foreign Direct Investment[1] (FDI) has been playing a significant role since post second world war period.Until 1980s, the flow of capital was mostly restricted among the developed economies. This trend changed in 1980s, when many Latin American countries started to liberalize their economy to attract FDI to recover their economy from severe debt crisis. After the failure of Cultural Revolution and lack of its capital, Republic of China initiated to open up its economy in 1985. Subsequently, other Asian countries liberalized their economic policies as well towards attracting foreign investment to bridge the gap of shortfall of domestic savings against investment for fixed capital formation. India, with no exception adopted liberalization, privatization and globalization (LPG) in early 1990s with a series of reforms thereafter and attracted FDI not only to recover its currency crisis but also to create additional employment and to accelerate overall economic growth. The flow of FDI is also played an important role to acquire advanced knowhow and knowledge. Precisely, last few decades developing economies have been continuously reforming their economies according to the need of the hour to invite more and more FDI.

Foreign direct investment policy of India can broadly be divided into two distinct phases – pre liberalization (1947 to 1990) and post liberalization period (1991 to till date). Keeping in view the broader objective of becoming 'self-reliant', inviting foreign direct investment always became a tough call for the Indian policy makers. FDI was only encouraged in the area of high

[1]According to the 4th edition of the Organization of Economic cooperation and Development (OECD) benchmark definition, "FDI is *a category of investment that reflects the objective of establishing a lasting interest (direct or indirect ownership of 10% or more of the voting power) by a resident enterprise in one economy (direct investor) in an enterprise (direct investment enterprise) that is resident in an economy other than that of the direct investor."*

technology based capital incentive industries whereas it was not allowed in the area of low technology based labour intensive industries to protect the domestic child industries. The enactment of Foreign Exchange Regulation Act (FERA), 1973 consolidated regulatory framework. Consequently, as a matter of export promotion policy some restrictions were removed for the foreign enterprises engaged in export oriented business. Apart from that, success of FDI for some Asian countries encouraged the Government to establish Special Economic Zones (SEZs) and Export Processing Zones (EPZs). Government also framed liberal policies and provided special incentives for encouraging foreign enterprises those who engaged in export oriented production in these areas.

However, those export promotional policies failed to meet the expectations of the policy makers due to overall restrictive environment of Indian economy. To overcome such challenges, India government reframed its Industrial Policies in 1980 and 1982 and introduced first Technology Policy in 1983. The main characteristic of such policies were to remove some restrictions in trade and investment and make export more competitive. These policies also emphasized on modernization and liberalization which was substantiated by reduction of tariffs and bringing many items from import licensing to Open General Licensing (OGL).

With the aim of integrating Indian economy with the world economy India embraced globalization and liberalization policy in 1991. To enhance FDI a series of reform measures have been taken, some of which are included, (i) approval of FDI through dual rout – automatic rout of Reserve Bank of India and Government's approval rout (ii) permitting 100 percent FDI in high priority sectors by Nonresident Indians (NRIs) and Oversees Corporate Bodies (OCBs), (iii) to boost up the confidence of the foreign entrepreneurs Convention of Multilateral Investment Guarantee Agency (MIGA) was signed (iv) Foreign Investment Promotion Board (FIPB), Secretariat of Industrial Assistance (SIS) and Foreign Investment Implementation Authority (FIIA) established under the government surveillance were established to facilitate approval of FDI through government rout. These efforts were further strengthened by replacing FERA, 1973 by Foreign Exchange Management Act (FEMA), 1999 which is less rigid. Continues financial sector reform abreast paved the way for greater capital account liberalization.

Those relentless efforts towards making FDI friendly economy during pre-liberalization and post liberalization period brought confidence among the multinational entrepreneurs to invest in India which is testified at the time of global financial crisis of 2008- 09 and 2009 - 10. This time when global FDI flow decelerated significantly, decrease of FDI inflow to India was relatively moderate. FDI has become one of the major contributor of Gross Domestic Product (GDP) of India. There has been manifold increase of FDI especially, in the post reform era. In 1990 the net inflow of FDI in India was 0.0747 percent of GDP which stood up to 2.10 percent of GDP in 2015[2]. FDI has become one of the main driving force of economic growth of India. According to the latest world investment report published by United Nations Conference on Trade and Development (UNCTD) India ranked ninth position in terms of inflow of FDI followed by Australia in 2016[3].

An Artificial Neural Network (ANN) (Yu et al. 2017; Box et al. 2015) is a data processing paradigm that is influenced by the working

---

[2]http://databank.worldbank.org/data/reports.aspx?source=sustainable-development-goals-(sdgs)&Type=TABLE&preview=on# (Accessed on 9th January, 2018)

[3]http://www.livemint.com/Politics/78UuHKYx50q7sZDq5tkHrK/India-climbs-to-9th-position-on-FDI-inflow-list-US-retains.html(Accessed on 17thDcember, 2017)

principle of nervous systems of any animal being. ANN highly resembles the structure of nervous systems. It comprises of several sophisticated units or nodes that really perform complicated operations on data. ANN has several variants where each one has its own characteristics features. In this paper, we have implemented Long Short Term Memory (LSTM) neural network. LSTM is a special kind of Recurrent Neural Network (RNN) (Zimmermann et al. 2012), capable of learning long term dependencies. Unlike RNN, LSTM actively maintains self-connecting loops without degrading their performance (Hochreiter et al. 1997). The LSTM network is trained using backpropagation(Werbos 1988) through time and solves the vanishing gradient problem (Bengio et al. 1994). In LSTM, every node acts as a memory that can store the information of the long periods of time. In forecasting, prediction of the future values is a complicated process that depends on several parameters. LSTM is found to be more appropriate to address difficult sequence problems and provides state-of-the-art results.

Long Short Term Memory (Lipton et al. 2015; R2RT Blog 2016) is an astonishing invention for dynamic sequence modeling (Gers et al. 2000; Hochreiter et al. 1997). LSTM is a variant of Recurrent Neural Network (RNN). RNN was invented in 1980s that has adaptive feedback connections and is in principle as powerful as any computer. However, RNN cannot look far back in the past.

Like other neural networks, RNN is suffering from vanishing gradient problem where the weights closer to the end of the network change more than the beginning of the network. Actually, in gradient based learning technique, small change in parameter’s value will reflect the change in output value. However, it may happen, small change in the parameter’s value will cause small change in the output value - the model cannot learn effectively. It has been observed that for a small number of hidden layer, the problem is not so obvious. The problem will exacerbate with the increase in the hidden layers. Here, we can observe how the gradients change over time for a network with one input layer and two hidden layers:

This paper is an attempt to integrate financial theory, econometrics and artificial intelligence to forecast inflow of FDI in India (Zhang et al. 1998). Forecasting (Hochreiter et al. 1997) FDI inflow in India will help economic policy makers to know prospect of this type of investment in near future in a scientific manner. This will also help the policy makers to fine tune their economic policies to attract more FDI and take comparative advantage in the competitive world.

SECTION 2
MATERIALS AND METHODS

Though the literatures on forecasting FDI is limited, but some noteworthy attempts have been made for prediction of FDI inflow. However, most of the researchers used Autoregressive Integrated Moving Average (ARIMA), Generalized Autoregressive Conditional Heteroscedasticity (GARCH) and simple or exponential smoothing techniques to forecast FDI inflow.

The Box-Jenkins based ARIMA model is used to forecast inflow of FDI into Jordon for the time period 2004 – 2025 by using time series data of 1973 – 2003. The study concludes that there would be an increasing trend of FDI over the forecasted span. This study also concludes a positive impact of FDI on other macroeconomic variables of the economy.

TheDouble exponential smoothing model is examined to predict FDI inflow in India. The accuracy of the result is tested by computing autocorrelation coefficients and Ljung-Box Q statistics. This study concludes that in the upcoming decade though volume of FDI inflow of China would be the highest, however India will pace up quickly and would be able to hold one seventh share of total FDI inflow in developing Asia (Sidhu et al. 2009).

The Prediction of FDI inflow into Jordon is examined byARIMA model. This study covers the data from 1980 to 2010 for predicting FDI inflow of Jordon for the years 2011 to 2030. It is observedthat the time series is not stationery and suffers with unit root. The conclusion is that by the end of 2030 the expected total FDI inflow will be 29207.06 million Jordanian dinar with an average annual growth of 3.22 % (Al-rawashdeh et al. 2011).

The ARIMA technique is also applied to forecast FDI inflow into Brazil. This study finds that Brazilian FDI inflow would follow a moving average pattern after adjusting detected outliers. Theil Coefficient is testified the forecast accuracy (Turolla et al. 2011).

China's FDI inflow is predicted on the basis of intervention model and ARIMA-GARCH-M model. This work establishes that FDI inflow of China follows heteroscedasticity and rather it is better to explain the relationship between FDI and random error (Shi et al. 2012).

Holt's approach of double exponential smoothing technique to forecast FDI inflow into Srilanka is applied. This study suggests that Srilankan government should adopt comprehensive development strategy to attract more inflow of FDI (Kumar et al. 2012).

The simple exponential smoothing (SES), Holt-Winters exponential smoothing (HWES) and ARIMA model is observed to forecast FDI inflow into Zambia. This study shows ARIMA is the best fit method as it has minimum error compares to other two methods. This study predicts that there would have a steady growth of annual net FDI inflows of about 44.36% by 2024 in Zambia (Jere et al. 2017).

However, scant attention has been paid on application of Artificial Intelligence (AI) based nonlinear approach to forecast inflow of FDI. The Neuron Network (NN) technique to estimate FDI inflow into fifteen central and Eastern Europe countries is applied. The paper classifies whole input variables into four broad categories (economic, financial, social and gravity) and concludes that NN approach is more efficient than traditional regression methodologies to determine FDI and its trend projection (Plikynas et al. 2005). The NN model is also experimented to forecast FDI for the six Asian economies including India. The study takes time series data for the period 1970 – 2009. The observation is that it is possible to extract information hidden in the FDI and predict FDI inflows in future (Pradhan et al. 2010).

Limited application of AI in this area left ample scope of research to apply latest methods of AI in this field. Especially, prediction of inflow of FDI using Recurrent Neural Networking (RNN) model is nonexistent. In this paper we apply Long Short Term Memory (LSTM) – a special kind of RNN method of AI to predict FDI inflow in India.

SECTION 3

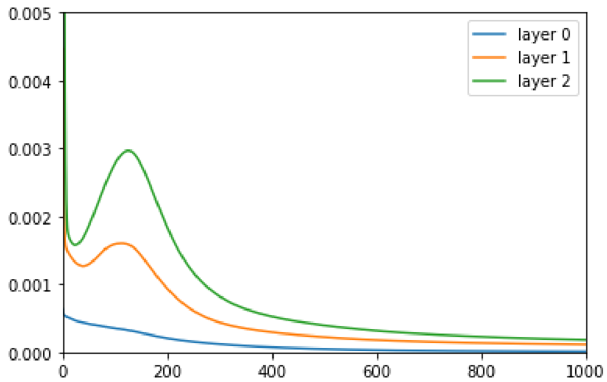
3.1 ARCHITECTURE OF LSTM

3.1.1 Problems of other Neural Networks

Long Short Term Memory is an astonishing invention for dynamic sequence modeling. LSTM is a variant of Recurrent Neural Network (RNN). RNN was invented in 1980s that has adaptive feedback connections and is in principle as powerful as any computer. However, RNN cannot look far back in the past.

Like other neural networks, RNN is suffering from vanishing gradient problem where the weights closer to the end of the network change more than the beginning of the network. Actually, in gradient based learning technique, small change in parameter's value will reflect the change in output value. However, it may happen, small change in the parameter's value will cause small change in the output value - the model cannot learn effectively. It has been observed that for a small number of hidden layer, the problem is not so obvious. The problem will exacerbate with the increase in the hidden layers. The RNN has feedback loops in the network which help it to store information in the memory over time. However, the RNN architecture prevents to memorize the long-term dependencies with time because the gradient of the loss function decays exponentially with time (Bengio et al. 1994; Pearlmutter et al. 1995). Here, we can observe how the gradients change over time for a network with one input layer and two hidden layers:

Figure I: Change of gradients over time



LSTM network is a variant of RNN that uses additional units in conjunction with standard units. LSTM comprises of a 'memory cell' that can maintain information in memory for long periods of time. A number of gates is used to control when information enters the memory, when it's output, and when it's forgotten. This architecture aids to learn longer-term dependencies.

3.1.2 A journey from Simple Neural Network to LSTM

A simple neural network with a single hidden layer takes as input a vector **x**, where we assume it consists of a set of neurons. The input layer is connected to the hidden layer through **W** weight matrix and in turn, hidden layer is connected to the output layer using **V** weight matrix. The output of any hidden layer is called 'logits' or 'activations'. Therefore, the output of the jth hidden neuron is:

$$h_j = \phi(\sum_i w_{ij} x_i)$$

Here ϕ is an activation function. We can transform the output layer via a softmax function. Therefore, the output of the jth output neuron is given below:

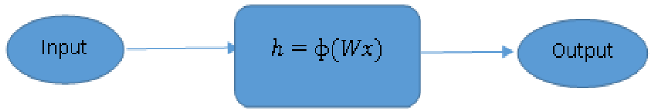
$$y_j = \sum_i v_{ij}\, h_i$$

In matrix notation:

$h = \phi(Wx)$and $y = Vh$

Figure II

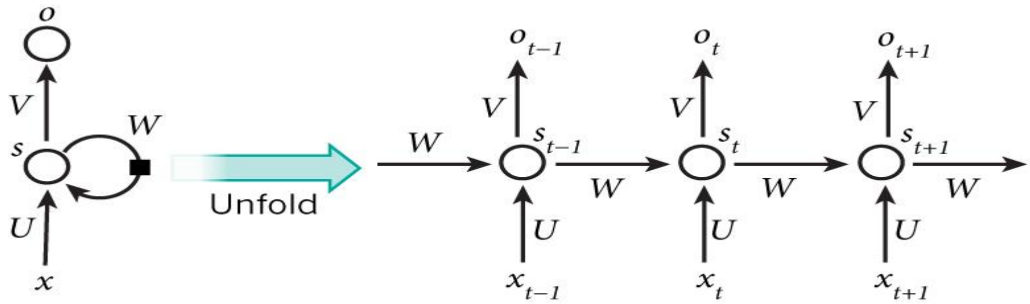
Structure of a simple Neuron network



3.1.3 Remembering Information with RNNs

The basic advantage behind the RNN is to make use of sequential information. In traditional neural network, we assume that all inputs and outputs are independent to each other. Therefore, it is suitable for the situation where current data point doesn't depend on previous data points. However, in forecasting problem, the forecasted value highly depends on previous data points or previous trend. In this situation traditional neural network will fail to predict the future value of a trend. Apart from the given example, various real life problems (image captioning, sentiment analysis, question answering, speech recognition, prediction of next word, anomaly detection in time series etc.) need to know the values of the previous data points that influence to occur the current data point. It is the capability of the RNN that fulfills our requirement in a sophisticated way. Another way to think about RNNs is that they have a "memory" which captures information about any context, it may be the value of the previous data point or what has been calculated so far.

Figure III: Basic Structure of RNN



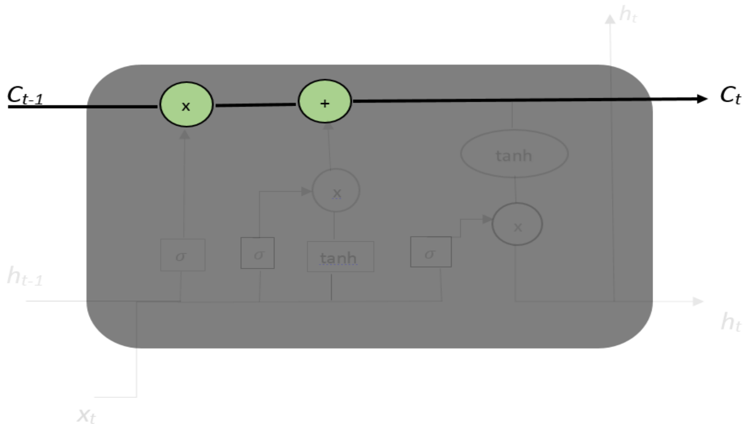
The above figure represents the basic structure of a RNN and its elaboration considering different layers. The unfolded part consists of three hidden layers, namely s_{t-1}, s_t and s_{t+1} at time step t-1, t and t+1 respectively. Each s represents the memory of the network. $s_t$is evaluated based on the values of the previous hidden layer and the current value of the layer using the mathematical expression: $s_t = f(U_{x_t} + W_{s_{t-1}})$. The function f is usually a non-linear function such as tanh or ReLU. x_t is the input at time step t. $o_t$is the output at hidden layer $s_t$andit's calculated as: $o_t = softmax(V\ s_t)$.

Final destiny: Long Short Term Memory

Now, it's time to think about how our model updates its knowledge in a long run. So far, we haven't put any constraints on updates. As a result, its knowledge may change chaotically. The information gets quickly transformed and vanishes in due course of time. Apart from that that, in reality, RNN is limited to look back only a few steps. Therefore, it creates the problem to maintain a long-term memory. To retain the values of previous multiple steps (of long run), we can modify the model as follows (Rybalkin et al. 2017; Kyunghyun et al. 2014; Kyunghyun Cho et al. 2015):

i. Adding a forgetting/remembering mechanism. The crucial concept in LSTM is the cell state, which is the horizontal line running through the top of the Figure 4. The cell state acts as a conveyor belt, which runs straight down the whole chain with some minor linear interactions. The information flows along it unchanged.

Figure IV: A typical LSTM cell state



There is a special regulatory structure called "forgetting/remembering gate" which controls to add or remove information to the cell state. This gate uses a sigmoid function for taking decision. The unique feature of this gate is that, based on context, it has ability to decide which part(s) of information is/are required in future for successful prediction. If it sees the information is not required at all, it simply throws it away from the cell state. Actually, it looks at h_{t-1} and x_t and produces the output in the range 0 to1 for each number in the cell state C_{t-1} where 1 represents "complete keep this" while a 0 represents "completely get rid of this". In the forecasting problem, the prediction of the future data points highly depend on the previous pattern of the trend. Therefore, it is required to consider and remember the former behaviors of the data points.

ii. Adding a saving mechanism. When the model sees a pattern, it needs to learn whether any information about the pattern is worth for future prediction.Therefore, at first, it forgets it's all long-term information as it is no longer required. Then, it takes decision, which parts are required actually for future computation and saves them into its long-term memory.

iii. **Focusing long-term memory into working memory.** Finally, the model needs to take decision which parts of its saved long-term memory are immediately useful. For example, seasonal variation may be a useful piece of information to keep in the long, however it is probably irrelevant for non-seasonal forecasting. Therefore, instead of keeping the full long-term memory all the time, it learns which parts to focus on instead.

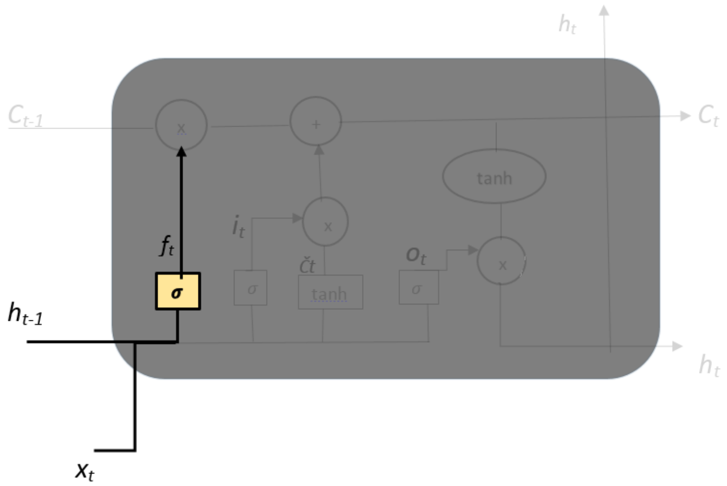
The above features make LSTM superior over RNN. RNN can overwrite its memory content at each time step in an uncontrolled manner whereas LSTM has the ability to remove, update or retain the information in the memory over longer periods of time.

Mathematically

In the first step, LSTM will decide what information it will throw away. This decision is taken by the sigmoid layer called 'forget gate'. It looks at h_{t-1} and $x_t$and produces a number between 0 and 1 for each number in the cell state C_{t-1}. The figure V describes the situation.

$$\boldsymbol{f_t = \sigma(W_f . [h_{t-1}, x_t] + b_f)}$$

Figure V: Sigmoid layer of 'forget gate'

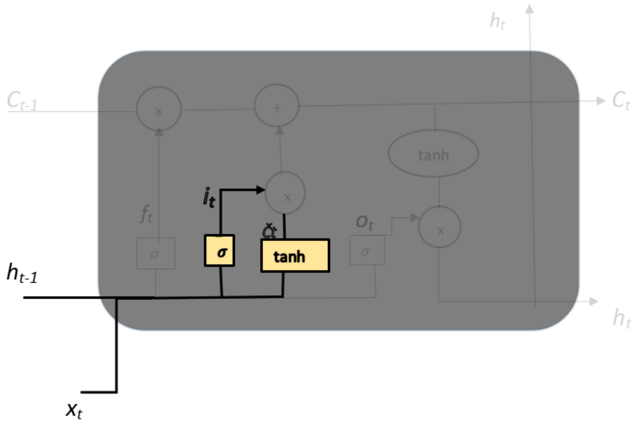


In the second step, LSTM will decide what new information it will store in the cell state. This has been accomplished in two steps. In the first step, a special sigmoid layer called 'input gate layer' will decide which values it will update. In the subsequent phase, a tanh layer creates a vector of new candidate values, $\widetilde{C_t}$, that should be added to the state. Finally, the architecture combines these two to create an update to the state. The process is shown in figure - VI.

$$\boldsymbol{i_t = \sigma(W_i . [h_{t-1}, x_t] + b_i)}$$

and$\boldsymbol{\widetilde{C_t} = \tanh (W_C . [h_{t-1}, x_t] + b_C)}$

Figure VI: Sigmoid of 'input gate layer'

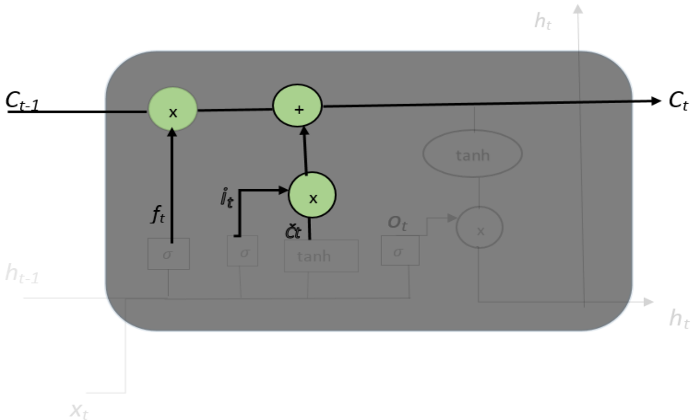


In the next step, LSTM updates the old cell state, C_{t-1}, into the new cell state C_t. LSTM multiplies the old state by $\boldsymbol{f_t}$and adds with $i * \widetilde{C_t}$.The figure VII displays the situation.

$$\boldsymbol{C_t = f_t * C_{t-1} + i * \widetilde{C_t}}$$

Figure VII

Updating old cell to new cell state



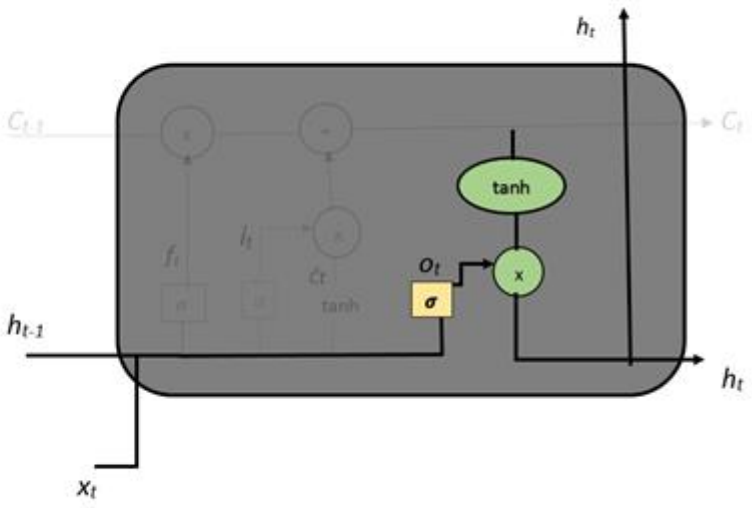
Finally, figure VIII shows howit produces the output based on the cell state. At first, a sigmoid layer will decide which parts are required to pass to the output. Then it put the cell state through tanh and multiply with output of the sigmoid layer.

$$\boldsymbol{o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)}$$

and$\boldsymbol{h_t = o_t * \tanh(C_t)}$

Figure VIII

Production of output based on cell state

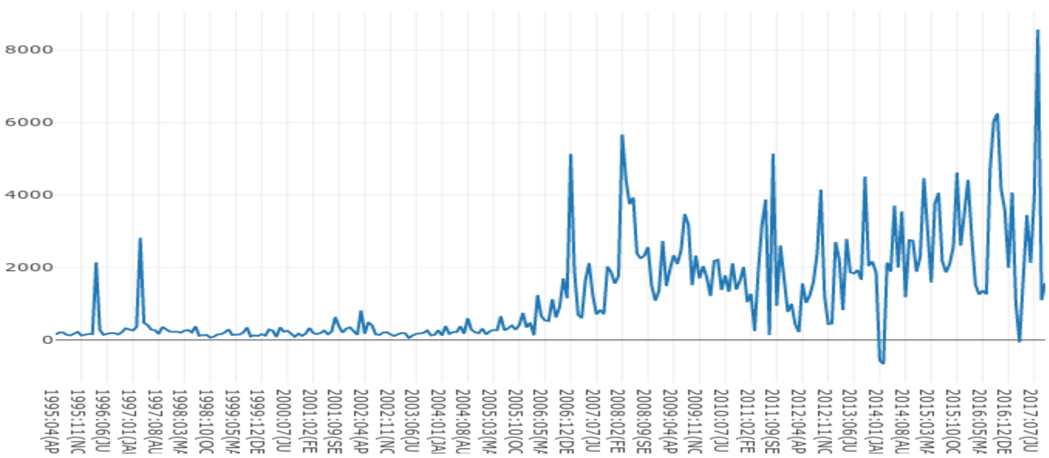


3.2 Methodology

In this research paper, our objective is to forecast the future trend of the net FDI inflowthat helps the policy makers to take decision about its future course of action. Highest accuracy level of the predicted value of FDI inflow is a precondition of success for appropriate policy measures without disturbing other macroeconomic factors. Hence, to take appropriate policy measures regarding inflow of FDI, it is obvious that the prediction should be accurate enough, so that it doesn't hamper the growth of the country. We have collected the data from the Reserve Bank of India's (RBI) official website. Implementation of any machine learning algorithm requires large volume of date to get accurate results. However, in the website we get only few data points. It is a monthly time series data starting from 1995(April) to 2017(October). It has only two fields, namely 'Year' and 'Net FDI inflow'and there are only 270 rows. The Figure IX represents Net inflow of FDI (in Million US Dollar)data distribution over time and it shows irregular pattern.

Figure IX

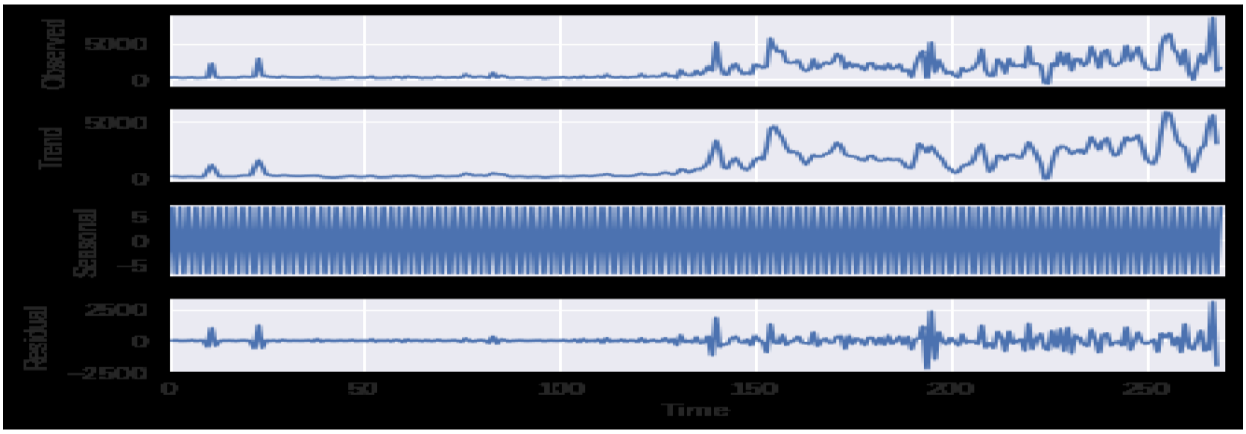
Net FDI inflow distribution over time (From April, 1995 to October, 2017)



3.2.1 Time Series Decomposition

Time series data vastly differs from usual random data. In random data, there is a single component to analyze for various purposes. Unlike random data, time series data mainly comprises three components, namely trend, seasonal and residual (Cleveland et al. 1990; Zhang et al. 2005). Each component has its own characteristics. In forecasting problem, each component has to be predicted to get the actual predicted value. However, in general, we make our time series data stationary, so that we can deal with only residual part of the time series data excluding trend and seasonal components. If we look deep into any time series data for longer period of time, we can see that trend and seasonal components are regular patterns found in the time series data and they must be removed from the data before we can forecast the irregular component.

Figure X
Decomposition of time series data of FDI inflow into trend, seasonal and residual



In this research paper, we have shown two-fold advantages of time series decomposition. Firstly, if the user is interested in the long term behavior of the data then it makes sense to study the trend. Similarly the seasonal fluctuations can be understood better if they are studied in isolation. Secondly, the accuracy or RMSE value gets reduced while dealing with decomposed data. Hence, we have decomposed the FDI time series data in to trend, seasonal and residual components as shown in Figure - 10**.** Here, we have implemented LSTM algorithm on each component of FDI data to show the facts discussed above. It is obvious from the figure that the seasonal part has no irregular component. Therefore, it creates little impact on prediction. When we diagnose the pattern of a trend component, generally we consider the distribution within a long span of time. Due to scarcity of data points, it is very difficult to realize the pattern. Therefore, individually we have implemented LSTM on trend component also.

3.2.2 Hyper-Parameter Tuning in LSTM

The most important hyper-parameters in LSTM algorithms are number of epoch, number of neuron and batch size. These hyper-parameters determine the success of our model. There could be any number of hidden layer and each hidden contains one to multiple number of neuron. Batch size denotes the subset size of our training sample (e.g. 50 out of 10000) which is going to be used in order to train the network during its learning process. Each batch trains network in a successive order, taking into account the updated weights coming from the appliance of the previous batch. Basically there is no thumb rule to select the numbers of these hyper-parameters. In order to secure the ability of the network to generalize, the number of nodes has to be kept as low as possible. If we have a large excess of nodes, our network becomes a memory bank that can recall the training set for perfection, but does not perform well on samples that was not part of the training set. Therefore, we have implemented our algorithm for different values of epoch, neurons and batch size and result is shown in Table I. Apart from these hyper-parameters, another hyper-parameter 'loss' also plays major role in computation. In this experiment we have selected 'loss' function as 'mean_squared_error' as a best choice as shown in the following figure.

Table I

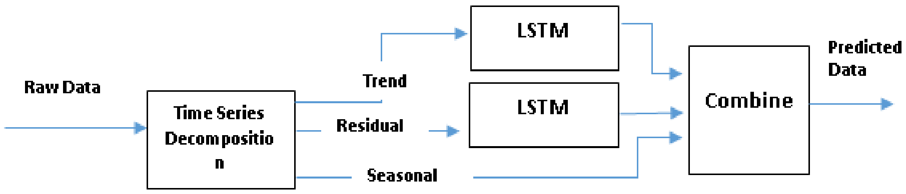
RMSE values for different loss functions

Function	Train RMSE
Mae	573.626
mean_squared_error	590.351
mean_absolute_error	572.140

3.2.3 Flow chart and implementation plan

Figure XI

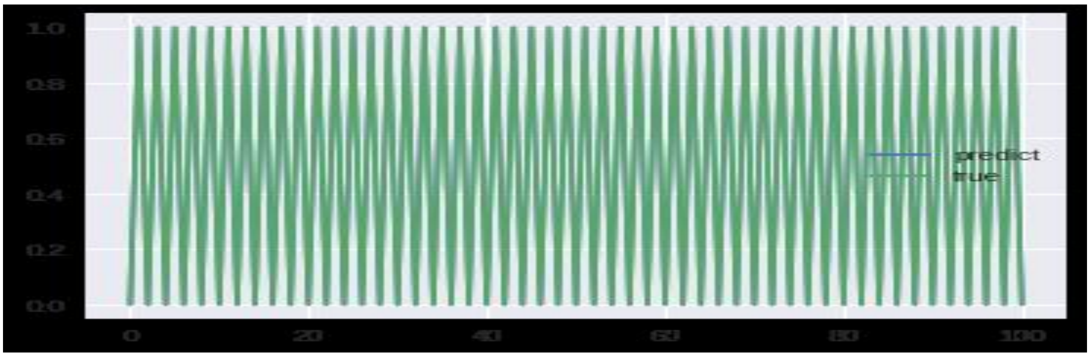
Flow Chart of the proposed model



The above diagram (figure XI) shows the pictorial presentation of our proposed model. Raw data is collected from the RBI website. Our objective is to predict the individual component of the FDI data and then combine to get the actual predicted value of the FDI data. Therefore, raw data of FDI is decomposed into three components, namely trend, residual and seasonal. From the decomposed data, we select the trend and residual parts and get feed individually to the LSTM algorithm. As we described earlier, seasonal part has no irregular pattern, therefore it has not been introduced in LSTM. In fact, we have used the seasonal component and have observed there is an overlap of actual and predicted values as shown in the following figure XII.

Figure XII

Prediction of the Seasonal component

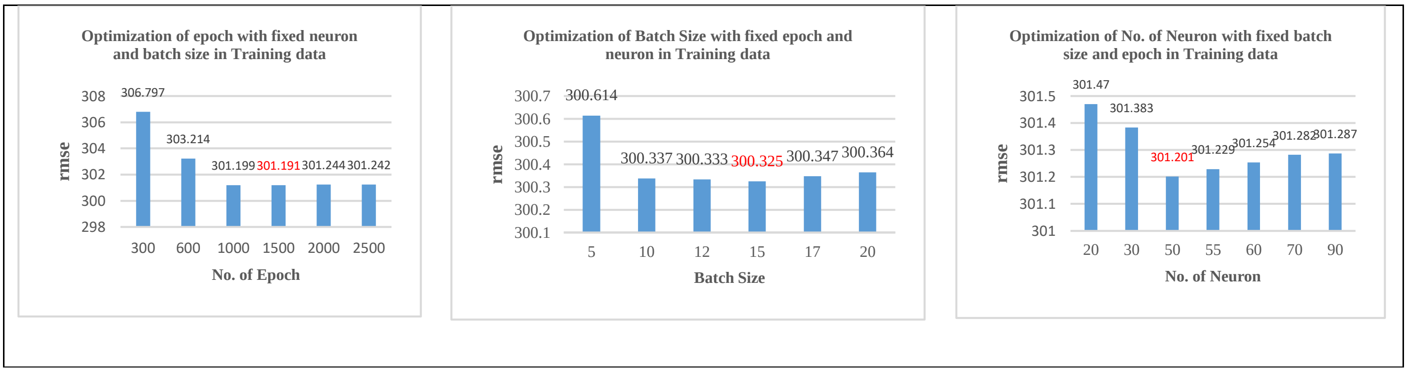


After that we have considered the predicted values of trend and residual components and combined with native seasonal component to get the resultant predicted value of the FDI data.

SECTION 4
RESULTS AND DISCUSSION

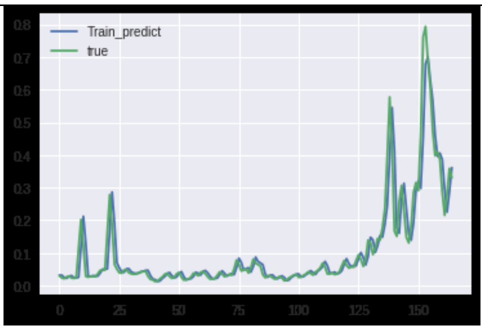
The following section describes various outcomes that have been produced during our experimental process. In this research paper, we have considered the ‘rmse’ as the measure of accuracy. We have implemented the LSTM algorithm for different values of number of epoch, number of neuron and batch size. Then we have selected them based on best rmse values and have found the optimized values are 301.191, 300.325 and 301.201 for number of epoch, batch size and number of neuron respectively. The following Figure represents the ‘rmse’ values for different hyper-parameters.

Figure XIII
RMSE values for different hyper-parameters



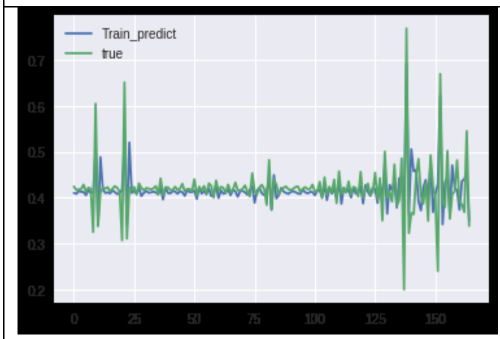
The prediction accuracies for trend and residual components are giving in the following figure XIV.

Figure XIV
Prediction accuracies for trend and residual components

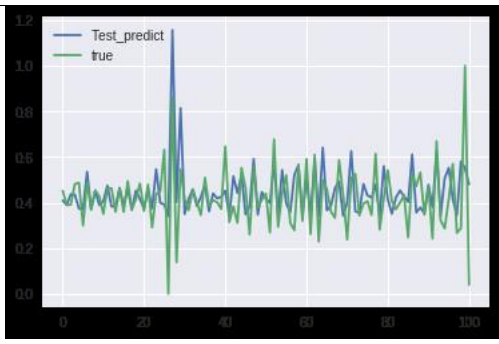


a) Prediction of Trend in Training Phase with RMSE=300.38

b) Prediction of Trend in Testing Phase With RMSE=674.35



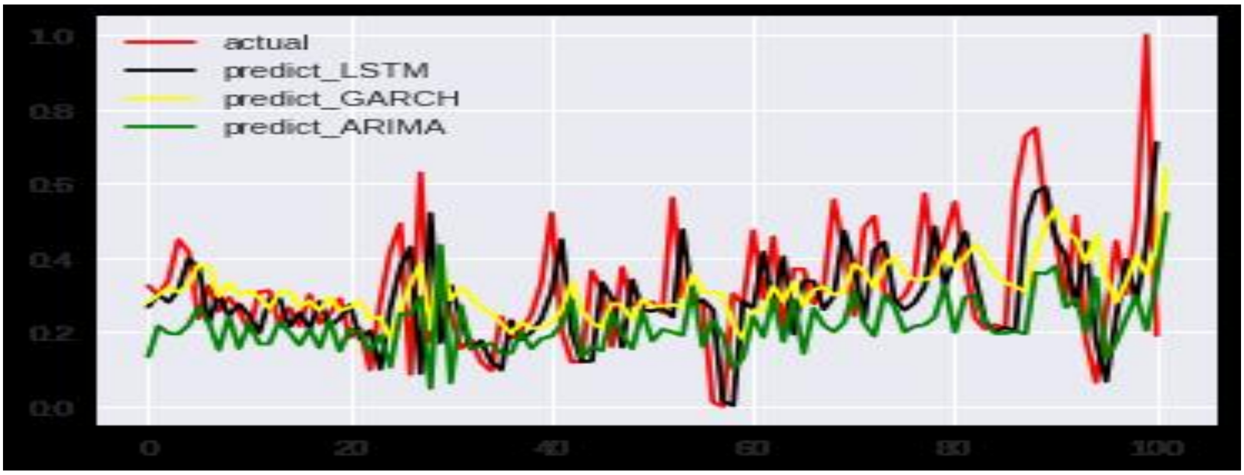
c) Prediction of Residual in Training Phase with RMSE=200.43



d) Prediction of Residual in Testing Phase with RMSE=657.50

There are some traditional forecasting tools available for accurate prediction. ARIMA, GARCH are well known linear models that perform well for small amount of data (Zhang 2013). However, performance of these models degrade with huge volume of data and data is of complex in nature (Petrica et al. 2016). In reality, FDI data is highly inflated with lot of noises and influenced by a lot of factors. This type of data is hardly predicted accurately using linear model. Moreover, ARIMA and GARCH models require a series of parameters which must be calculated based on data. Therefore, it is a wise decision to apply non-linear models (neural network) that can take care of irregular patterns and/or several influencing factors of the data and no need to estimate the parameters during forecasting. LSTM, a specialized neural network, needs only estimating the hyper-parameters. Figure XV shows the accuracy of prediction of different linear and non-linear models.

Figure XV
Prediction pattern of various forecasting tools



SECTION 5
CONCLUSION

In this research paper, our objective is to forecast the FDI values using deep learning technique. Conventional linear models are inefficient to forecast when data contains too much noise, unusual patterns. LSTM, a special kind of neural network, is a very efficient technique that has a unique remembering skill to memorize patterns occurred long back in time scale. It is true that LSTM takes more time compare to other linear models and it is really an overhead when data volume is less. However, in forecasting technique, most important thing is accurate prediction rather than the execution time. To prove the supremacy of the LSTM, we have implemented ARIMA and GARCH techniques and we have found that LSTM outperforms the ARIMA and GARCH models in terms of accuracy as shown in the figure XV.

Accuracy of any machine learning algorithm including LSTM will increase with the increase volume of data set. However, the data set that we have used in this experiment is small. Therefore, we can increase the accuracy with huge volume of data set along with proper tuning of different hyper-parameters.

REFERENCES

[1] Al-rawashdeh, S. T., Nsour, J. H., &Salameh, R. S. (2011). Forecasting Foreign Direct Investment in Jordan for the Years (2011-2030). *International Journal of Business and Management*, *6*(10), 138 - 154.

[2] Bengio, Y., Simard, P., Frasconi, P.(1994). Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Netw. 5 (2), 157–166.

[3] Box, G. E. P., Jenkins, G. M., Reinsel, G. C., Ljung, G. M.(2015). Time Series Analysis: Forecasting and Control. John Wiley & Sons.

[4] Cleveland, R. B., Cleveland, W. S., Terpenning, I. (1990). STL: A seasonal-trend decomposition procedure based on loess. J. Off. Stat. 6 (1), 3.

[5] G. P. Zhang. (2003). Time series forecasting using a hybrid arima and neural network model. Neurocomputing, 50, 159–175.

[6] G. Zhang, B. E. Patuwo, and M. Y. Hu. (1998). Forecasting with artificial neural networks:: The state of the art. International journal of forecasting, 14(1), 35–62.

[7] Gers, F. A., Schmidhuber, J., Cummins, F.(2000). Learning to forget: continual prediction with LSTM. Neural Comput. 12 (10), 2451–2471.

[8] Hochreiter, S. and Schmidhuber, J. (1997). LSTM can solve hard long time lag problems. In Advances in Neural Information Processing Systems 9. MIT Press, Cambridge MA. Presented at NIPS 96.

[9] Hochreiter, S., Schmidhuber, J.(1997). Long Short-Term memory. Neural Comput. 9 (8), 1735–1780.

[10] http://databank.worldbank.org/data/reports.aspx?source=sustainable-development-goals-%28sdgs%29# (Accessed on 27th December, 2017)

[11] http://www.oecd.org/investment/fdibenchmarkdefinition.htm (Accessed on 29th December, 2017)

[12] Hyndman, R. J., Athanasopoulos, G.(2014). Forecasting: principles and practice. OTexts.

[13] Jere, S., Kasense, B., &Chilyabanyama, O. (2017). Forecasting Foreign Direct Investment to Zambia: A Time Series Analysis. *Open Journal of Statistics*, *7*(01), 122.(DOI: 10.4236/ojs.2017.71010).

[14] Kumar, G., &Dhingra, N. (2012). Growth performance and forecasts of FDI inflows to Sri Lanka. *E3 Journal of Business Management and Economics*, *3*(8), 307-317.

[15] Kyunghyun Cho, Aaron Courville, and YoshuaBengio. (2015). Describing multimedia content using attention-based encoder-decoder networks. IEEE Transactions on Multimedia, 17 (11):1875–1886.

[16] Kyunghyun Cho, Bart Van Merri¨enboer, CaglarGulcehre, DzmitryBahdanau, FethiBougares, HolgerSchwenk, and YoshuaBengio. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078,

[17] Lipton, Z. C., Kale, D. C., Elkan, C., Wetzel, R.(2015). Learning to diagnose with LSTM recurrent neural networks.

[18] Pearlmutter, B. A. (1995). Gradient calculations for dynamic recurrent neural networks: A survey. IEEE Transactions on Neural Networks, 6(5), 1212-1228.

[19] Petrica, A., Stancu, S., Tindeche, A.(2016). Limitation of ARIMA models in financial and monetary economics, Theoretical and Applied Economics VolumeXXIII, No. 4(609), winter, 19-42.

[20] Phung, Hang Bich, "Determinants of FDI into Developing Countries" (2016). Mark A. Israel '91 Endowed Summer Research Fund in Economics. 4. (http://digitalcommons.iwu.edu/israel_economics/4).

[21] Plikynas, D., & Akbar, Y. H. (2005). Explaining the FDI patterns in central and Eastern Europe: a neural network approach. *Ekonomika*, *69*, 78-103.

[22] Plikynas, D., & Akbar, Y. H. (2006). Application of modified MLP input weights' matrices: an analysis of sectorial

investment distribution in the emerging markets. *Neural Computing & Applications*, *15*(3-4), 183-196.(https://doi.org/10.1007/s00521-005-0018-6).

[23] Plikynas, D., & Akbar, Y. H. (2006). Neural Network Approaches to Estimating FDI Flows: Evidence from Central and Eastern Europe. *Eastern European Economics*, *44*(3), 29-59.(DOI: 10.2753/EEE0012-8755440302).

[24] Pradhan, Rudra Prakash, Forecasting Foreign Direct Investment in the Asian Economy: An Application of Neural Network Modeling (March 25, 2010). The IUP Journal of Computational Mathematics, Vol. 3, No. 1, pp. 39-53, March 2010. Available at SSRN: https://ssrn.com/abstract=1578071

[25] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu. (2017). Deep learning: A generic approach for extreme condition traffic forecasting. In Proceedings of the 2017 SIAM International Conference on Data Mining, 777–785.

[26] R2RT Blog.(2016). written memories: Understanding, deriving and extending the LSTM - R2RT. https://r2rt.com/written-memories-understanding-deriving-and-extending-the-lstm. html, accessed: 2017-9-10.

[27] Rybalkin, V., Wehn, N., Yousefi, M.R., Stricker, D.,2017, Hardware architecture of bidirectional long short-term memory neural network for optical character recognition, Proceedings of the Conference on Design, Automation & Test in Europe, March 27-31, Lausanne, Switzerland

[28] Shi, H., Zhang, X., Su, X., & Chen, Z. (2012). Trend prediction of FDI based on the intervention model and ARIMA-GARCH-M model. *AASRI Procedia*, *3*, 387-393.(https://doi.org/10.1016/j.aasri.2012.11.061).

[29] Sidhu, H. S., &Dhingra, N. (2009). Foreign direct investment inflows to India: growth and forecasts. *Foreign Trade Review*, *44*(3), 24-56.(https://doi.org/10.1177/0015732515090302).

[30] Turolla, F. A., &Margarido, M. A. (2011). Modeling and forecasting foreign direct investment into Brazil with ARIMA. *Economia Global e Gastao*, *16*(2), 86-100.

[31] Werbos, P. J. (1988). Generalization of backpropagation with application to a recurrent gas market model. Neural Networks, 1.

[32] Zhang, G. P., Qi, M.(2005). Neural network forecasting for seasonal and trend time series. Eur. J. Oper. Res. 160 (2), 501–514.

[33] Zimmermann, H.-G., Tietz, C., Grothmann, R. (2012). Forecasting with recurrent neural networks: 12 tricks. In: Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 687–707.