Periódico do Núcleo de Estudos e Pesquisas sobre Gênero e Direito
Centro de Ciências Jurídicas - Universidade Federal da Paraíba
V. 9 - Nº 02 - Ano 2020
ISSN | 2179-7137 | http://periodicos.ufpb.br/ojs2/index.php/ged/index

COMPARING THE PREDICTION ACCURACY OF LSTM AND ARIMA MODELS FOR TIME-SERIES WITH PERMANENT FLUCTUATION

Ghahreman Abdoli[1]
Mohsen MehrAra [2]
Mohammad Ebrahim Ardalani[3]

Abstract: In developing countries with an unstable economic system, permanent fluctuation in historical data is always a concern. Recognizing dependency and independency of variables are vague and proceeding a reliable forecast model is more complex than other countries. Although linearization of nonlinear multivariate economic time-series to predict, may give a result, the nature of data which shows irregularities in the economic system, should be ignored. New approaches of artificial neural network (ANN) help to make a prediction model with keeping data attributes. In this paper, we used the Tehran Stock Exchange (TSE) intraday data in 10 years to forecast the next 2 months. Long Short-Term Memory (LSTM) from ANN chooses and outputs compared with the autoregressive integrated moving average (ARIMA) model. The results show, although, in long term prediction, the forecast accuracy of both models reduce, LSTM outperforms ARIMA, in terms of error of accuracy, significantly.

Keywords: Prediction Model, LSTM, ARIMA, Forecast Accuracy, Tehran Stock Exchange.

Introduction

The Stock market prediction itself is violent and high risk, due to equivocal and unforeseeable nature of it, as well as complicated financial indicators. It is affected by noise and many hidden factors. No one knows the exact time of bearish or bullish the market. The Forecast situation becomes more difficult when the exchange market

1 Professor at University of Tehran, Faculty of Economics, email address: abdoli@ut.ac.ir
2 Professor at University of Tehran, Faculty of Economics, email address: mmehrara@ut.ac.ir
3 Ph.D. Candidate at University of Tehran at Alborz campus, email address: ardalani@ut.ac.ir

reflects an economic system without stability, regulation and money discipline. Developing countries with permanent fluctuation in economic data are the first candidate. Developing countries involve a variety of fundamental concerns in economics such as poverty, the relations between state and market, the gap between rich and poor, population growth, structural change, international debt and finance and in some cases permanent sanctions. Somewhat because of this, economic data and time-series generated in these countries, volatile abruptly over time in response to political currents, in such a way that causes the major roots of changes to be hidden. (Mukherjee et al., 2013) In this situation, even if a research group wants to make a prediction model, in the process of converting related qualitative indexes into quantitative indexes, it is inevitable to keep all the time-series nature. Data carries all the economic, social and even political events that have happened through the time. Regarded to permanent fluctuations and unrecognized independent variables, we must make data fit for any regression or modeling. It means, differentiation comes to help and subsequently we ignore the core of data. We go just for making a statistical model which most of the time does not reflect the reality of the market. Thus; making a robust forecasting model for a stock market in developing countries given by unmodified data is a challenging issue.

Linear regression is one of the solutions for stock market prediction however, the idea works better for short terms. (Ariyo et al., 2014) Besides, as mentioned above, the linearization of nonlinear multivariate economic time-series to regress, may not give the best response.

Cutting-edge approaches in data analysis allow gaining better forecast models. With the emerge of neural networks, researchers went for assessing its accuracy of prediction versus linear forecasting models such as ARIMA. Results in most of the time talked about the benefits of non-linear solutions. (Lin and Yeh, 2009; Binner* et al., 2005) However, in some research, ARIMA (C.-C. Wang et al., 2011) and sometimes hybrid models (Hansen and Nelson, 2003) resulted better.

There is a wide spectrum of applications for Deep-learning models in which financial forecasting is one of them. However, pre-request for using it, is big and clear data which again is an issue to be prepared. Although the idea of deep learning announced for more than two decades, with the help of developed hardware, its use is being increased.

In this paper, we chose Long Short-Term Memory (LSTM) as one of the methods of deep learning for forecasting on intraday stock price from Tehran (IRAN) exchange market as the representative of all factors of developing countries, plus permanent political issue, sanction and permanent rhetoric in international relations which reflects on the stock market. LSTM is supposed to train by daily opening price in 10 years and make a model to forecast the next 2 months after training data. Then the result compares with the ARIMA solution which is trained and calculated in the same data, program, and hardware.

The next section of this paper focuses on related work. Then, the methodology of LSTM (in comparison with RNN) and ARIMA will describe. After that, we will explain our experimental work and focus on our results, and the last part goes to our conclusion.

1.1. LSTM

Our algorithm is implemented in python pursuing Keras (TensorFlow Back-End) and categorized in 3 parts with the details as below:

2. Related Work

Before explaining recent works, it is worth mentioning that interest of using LSTM for the past year, has surprisingly gone up and its accuracy in prediction is going to prove its power in almost every field of science. It took around two decades until this approach became global. Here is some related research to LSTM or ARIMA for stock forecasting.

(Long et al., 2019) offered a new model by the integration of convolutional
and recurrent neurons called multi-filters neural network. Results show the proposed model outperforms LSTM and other machine learning alternatives. (Cao et al., 2019) reported a combined model of empirical mode decomposition and LSTM gave better prediction in one-step-ahead than

multi-layer perceptron, support vector machine and single LSTM. (K. Zhang et al., 2019) presented a new model of Generative Adversarial Network (GAN) with MLP as the discriminator and the LSTM as the generator for making a stock prediction model. The outcome shew proposed solution outperformed other alternatives in machine learning, significantly. (T. Kim and Kim, 2019) reported that candlestick was the most suitable chart for predicting with using a fusion CNN-LSTM model. The research shows, forecast error decreases by running a mix of temporal and image features of the same data in comparison with using them separately. Also, (Eapen et al., 2019) and before that (Zhou et al., 2018) presented a CNN-LSTM model can perform better result in terms of prediction accuracy. (H. Wang et al., 2019) reported an LSTM model for forecasting a stock market in which indexes come through analysis of text information. Results show, proposed approach decreases both MSE and MAE. (S.-S. Yu et al., 2019) extracted the characteristics of stock data with convolutional recurrent neural and then input the feature in LSTM. They finally got 3.449 for RMSE as the result of their forecast model. (Sang and Di Pierro, 2019) as well, found using both neural network and traditional technical analysis, performs better than each of them, separately. (D. Kim and Baek, 2019) based on LSTM, offered a model for recognizing fluctuation named factor-augmented HAR. They believe; this approach could give a better result for stock prediction. (Chen and Ge, 2019) with proposing attention mechanism in LSTM forecasted Hong Kong stock and compared the result with single LSTM. Results show the ability of the approach, which offered in their paper. (Borovkova and Tsiamas, 2019) put forward a group of LSTMs (5544 networks) for stock forecasting which are evaluated by the weighting method and create 462 group model! Data for learning has come from 44 US stock markets. The result of testing shows, the group model outperforms all the alternatives for prediction, which are assessed in this paper. (Liang et al., 2019) came up with a method in signal denoising named MOCWT with a new approach for taking the noise of financial data out. Then, they used LSTM and based on the result and prediction accuracy became better. (Du et al., 2019) used LSTM for apple stock forecast and reached absolute error of 0.155 and 0.033 for the univariate and multivariate input, respectively.(Karmiani et al., 2019; Pawar et al., 2019) compared tradition machine learning methods (SVM, RF, FFNN) with LSTM in terms of the power of prediction and architectures in their researches. (Hiew et al., 2019) put up an opinion index based on financial news with the BERT method (developed by Google) on three stock markets. Then, they choose VAR and LSTM for catching the probable nonlinearity of news effects on exchange return. LSTM at the end gave lower MSE in comparison with VAR. (Hushani, 2019) reported between ARIMA, VAR, LSTM, and NARX[4] , it is NARX which can give prediction with high accuracy in the short-term period but failed in a long period. Although VAR forms a good trend line (which is needed for trading), a profitable strategy should be a combination of ML and technical analysis. (Manurung et al., 2018) compared ARIMA and LSTM for prediction accuracy and found, 1-year training of the LSTM model (backed by their data), in comparison with increasing the time of training to 3 or 5, did not necessarily give the better result. In the best scenario, LSTM forecasted with 94% accuracy, while ARIMA reached 56%. (Karakoyun and Cibikdiken, 2018; McNally et al., 2018) compared LSTM and ARIMA for predicting Bitcoin. The first paper reported the accuracy of 52% and RMSE of 8% as the result of the ML approach. Then, they run the selected method in two different hardware (GPU and CPU) which as expected GPU (the newer tech) resulted better. The second one, reported MAPE of 11.86% for ARIMA and 1.46% for LSTM. Thus, both papers show LSTM can give better prediction than ARIMA with Bitcoin data. (Rebane et al., 2018) found seq2seq[5] could give better bitcoin prediction versus ARIMA. (Huynh et al., 2017) proposed an extended version of LSTM and GRU called BGRU for exchange forecast. It used financial news for 7 years (Bloomberg (447,145 news) and Reuters (106,521 news)) and with stock data tried to forecast. The method in comparison with LSTM and GRU, was

[4] Nonlinear autoregressive Exogenous

[5] It is a kind of ML model in TensorFlow library.

effective and could improve the accuracy of prediction. (Selvin et al., 2017) applied a sliding window approach for daily closing price prediction by using LSTM, RNN, and CNN in their paper. (H. Li et al., 2018) as well, with the help of attention-based multi-input could improve the prediction of accuracy out of data in the research. (S. Yu and Li, 2018) found LSTM illustrated better results than GARCH to predict the stock volatility out of their data. (Skehin et al., 2018; Nagahisarchoghaei et al., 2018; Nagahi et al., 2018) presented a combined model of ARIMA, LSTM and wavelet methods to predict FAANG[6], and based on the result, accuracy of prediction improved. (H. Y. Kim and Won, 2018) found a combined of LSTM and GARCH, gave the lowest error for accuracy in comparison with other models in their research. (Hiransha et al., 2018) compared the MLP, RNN, CNN, and LSTM in terms of accuracy in the forecast for national stock exchange of India and New York stock exchange. The models were trained for a company and CNN outperformed other models. While the model was trained with NSE data, it could predict NYSE. It might have been due to the same inner dynamics in both markets. Then the result compared with the forecast model backed with ARIMA and again made a better answer. (Baughman et al., 2018) used LSTM for prediction of amazon stock and compared the result with ARIMA. They could reduce the error of training by 95%. (Shah et al., 2018; Minami, 2018) presented LSTM-RNN outperformed for making longer-term predictions. Minami proposed a sequential learning model with corporate behavioral event information and macroeconomic indices that use LTSM-RNN methods. (Naik and Mohan, 2019) as well found a combination of RNN and LSTM worked better than each of which, individually. However, (Pai and Lin, 2005) found a combination of forecasting models that work well individually, does not all the time produce the best results. (Nandakumar et al., 2018) reported for small stock price prediction, LSTM outperformed RNN. (Fischer and Krauss, 2018) found deep neural network, logistic regression classifier and random forest solutions do

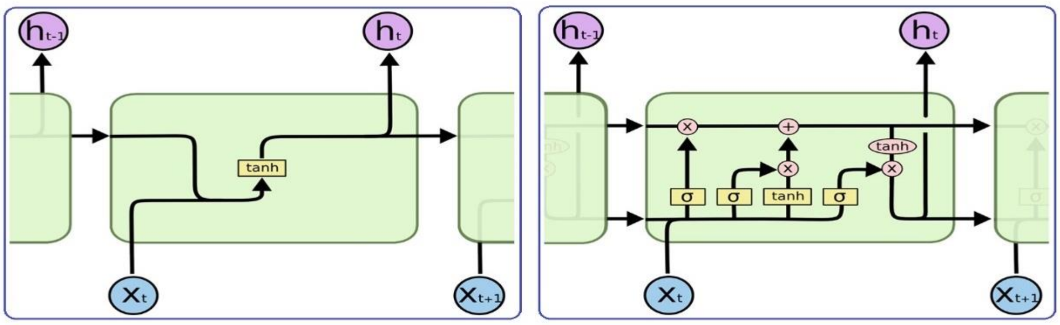
[6] Facebook, Amazon, Apple, Netflix, and Google.

not present the accuracy of prediction as well as LSTM. (Shao et al., 2017) used the K-means algorithm for clustering the data and then set LSTM for each cluster to predict. They repeated the training model after each prediction with the target of making the distance of the clustering center and predicted data, shorter. Their resulted shew, model had a more accurate prediction. (J. Li et al., 2017) combined LSTM with investor sentiment and market factors for making a stock prediction model. The proposed model gave 87.86% of accuracy. (Zhuge et al., 2017) put forward a combined model of emotional analysis model and LSTM for forecasting the open stock price. They first, used an emotion classifier back by naive Bayesian for analyzing the data. Then, combined the emotional data (the export of experiment) with actual data and gave it to LSTM for training. Their result shew prediction accuracy improved by their method. (Bao et al., 2017) believe that based on their data (coming from six markets), their combined model which was made of wavelet transforms and stacked autoencoders models in LSTM performed better accuracy in prediction. (Choi, 2018; G. P. Zhang, 2003) applied a hybrid model of ARIMA-LSTM and ARIMA-ANN respectively. Results in both works indicated hybrid is the way to make forecasting accuracy better. (X. Pang et al., 2018; X. W. Pang et al., 2018) both illustrated LSTM with an embedded layer that reached better accuracy of prediction. (Di Persio and Honchar, 2017) found LSTM in comparison with basic RNNs and Gated Recurrent Unit (GRU) gave better results. (Ariyo et al., 2014) found results of ARIMA for forecasting in the short-term is reliable.

3. Methodology

3.1. Long Short-Term Memory

It is a type of deep learning approach with memory and capable of learning long-term dependencies and sequential data. A Cell as memory part and "regulators" (composed of input, output and foregate gate) are the common architecture of this network. LSTM is designed for applications where the input is an ordered sequence. It is very much like an RNN but with more complicated innards. The great thing about RNN is that it remembers everything. It has got very long-range dependencies that can capture the entire



sequence, but that's also a downside! In LSTM we let RNN selectively choose whatever members and what it forgets. It effectively decides how much the network is going to carry data from previous time-step and how much is supposed to be used in current time-step. Like all neural networks, the node performs a calculation using the inputs and returns an output value. LSTM reuses the output from a previous step as an input for the next step, while in RNN, the output is used along with the next element as the inputs for the next step. A typical inner neuron of RNN and LSTM are as below:

Diagram 1: An inner neuron of an RNN (left) and LSTM (right) (Olah, 2015)

As Diagram.1 shows the RNN takes and combines the previous hidden state as input.1 with current time-step vector (X_t) as input.2. Then passes the result through some non-linearity ($tanh$) for squishing values between [-1 1] to regulate the network. Then it produces a new hidden state (h_t) and goes to the next time-step (X_{t+1}). By repeating the process, it may make a prediction model at the end of layers. (depends on what we want)

LSTM has more components on the inside. This is built on the idea of gates. Gates decide if data is relevant to keep or should be forgotten during training. All the gates have sigmoid (σ) activation to constitute smooth curves in the range 0 to 1 and the model remains differentiable. Data closer to 0, should be forgotten and near to 1, keeps to update cell state. Diagram.1 shows three symbols of sigma (σ) which are responsible for forgetting, input and output gate, respectively. Sigmoid either let information through or stops the information from going through and this decision depends on the input that it's getting. At first gate, it decides whether to remember the previous state based on a function of the previous hidden state (h_{t-1}) and the current vector input (x_t). The equation is as below. LSTM compact equations with a forget gate, first presented in 1997. (Hochreiter and Schmidhuber, 1997)

$$f_t = \sigma\left(W_f[h_{t-1}; x_t] + b_f\right)$$

The other decision LSTM needs to make is how much it is going to contribute to its representation from the current time step. The equations for that are as below:

$$i_t = \sigma(W_i[h_{t-1}; x_t] + b_i)$$
$$\tilde{C}_t = tanh(W_c[h_{t-1}; x_t] + b_c)$$

This is again a function of both the hidden state (h_{t-1}) and current input (x_t), plus bias ($b_{i\ or\ c}$) and then it passes through a non-linearity. There are two components to this step. The first one (i_t) says how much is it going to contribute and the second one ($\tilde{C}_t$) which is about what you are going to contribute. Function (tanh) distributes gradients. Hence prevents vanishing or exploding. Then, the next step goes to interpolation (like RNN) between the current state and the previous state.

$$C_t = f_t * C_{t-1} + \tilde{C}_t * i_t$$

f_t is about how much previous state is going to contribute and i_t is about how much the current contribution is going to contribute to the new hidden state representation. The next equation is about output (o_t) in each time step and making a hidden state h_t which is a function of cell C_t , not h_{t-1}.

$$o_t = \sigma(W_o[h_{t-1}; x_t] + b_o)$$
$$h_t = tanh(C_t) * o_t$$

An important fact about LSTM is that there is no need to concern about reaching a local optimal solution. The vanishing or fading gradient is a common problem occurs during ANN using gradient-based learning methods. This is accomplished by using a partial derivative of the error function to each parameter in each iteration of the training process, but gradient values (moving to the top of the grid) gradually become small enough that the weight changes are negligible. Because of that, the learning process gets slower, and in more severe cases, it stops the learning process. In feed-forward neural networks (MPL, CNN, …) it happens because of many layers and in Recurrent Neural Networks, it is because of the steps. LSTM can learn context specific temporal dependence. It means units have memory for a short or a long-time period and neurons (within their pipeline) kept a context of memory to allow for tackling sequential and temporal data. It is done by the possibility of passing information from the previous cell without any change, instead of iteratively modification at each time-step or layer. Weights as well are expected to converge to their optimal values. Thus, by adding three gates (input, output, forget) and $\tilde{C}_t$, this network solves the problem of disappearance of gradient and assists in controlling error during back-propagation.

3.2. ARIMA

An algorithm supported by the idea that time-series data in the past, can itself be used to predict in the future. The standardized methodology which is capable of dealing with non-stationary time-series was introduced first in 1970. (Box et al., 1970)

There are three steps in its methodology.

(1) Recognizing the type of model.

(2) Estimate parameters for optimizing.

(3) Checking residual analysis.

At first, stationary time-series needs to be calculated for deciding the proper type of network. If data is non-stationary, we must differentiate it once or twice. Because variance, covariance and mean need to be constant in time-series in this model.

An ARIMA model is when the time-series is differenced at least once (to make it stationary) and AR and MA terms are combined. The general mathematical equation of ARIMA:

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \cdots + \beta_P Y_{t-P} + \varphi_1 \varepsilon_{t-1} + \varphi_2 \varepsilon_{t-2} + \cdots + \varphi_q \varepsilon_{t-q}$$

As it shows, Y_t is made of constant + Linear combination of Y (up to p lags) + Linear Combination of Lagged forecast errors (up to q lags).

The equation shows with ARIMA (p,d,q) in which, "P" is the order of the 'Autoregressive' (AR). It shows the number of lags to be used as a predictor. The mathematical equation of pure AR(p) is as follows:

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \cdots . + \beta_P Y_{t-P} + \varepsilon_1$$

As it shows, Y_t depends only on its lags. So, it is a function of the lags of itself.

"D" in ARIMA (p,d,q), demonstrates the minimum number of differencing applied to the data to make the series, stationary. ARIMA turns to ARMA, provided that the time-series is already stationary. (d = 0)

"Q" in the equation is the order of 'Mean Average' (MA). It shows the number of lagged forecast errors. The mathematical equation of pure MA(q) is as follows:

$$Y_t = \alpha + \varepsilon_t + \varphi_1 \varepsilon_{t-1} + \varphi_2 \varepsilon_{t-2} + \cdots + \varphi_q \varepsilon_{t-q}$$

As it shows, Y_t depends only on the lagged forecast errors.

4. Experimental Work

The models in both ARIMA and LSTM are run by Python and the same hardware, to have an equal situation to judge. Diagram 2 shows train data and Diagram 3, demonstrates the test data (60 days). Historical stock data is from Tehran Securities Exchange[7].

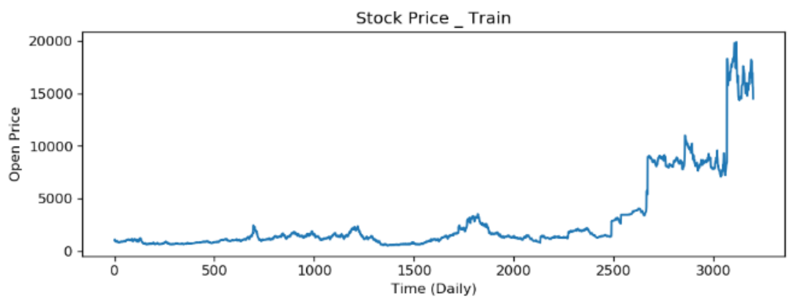


Diagram 2: Train data (day 1 – day 3200)

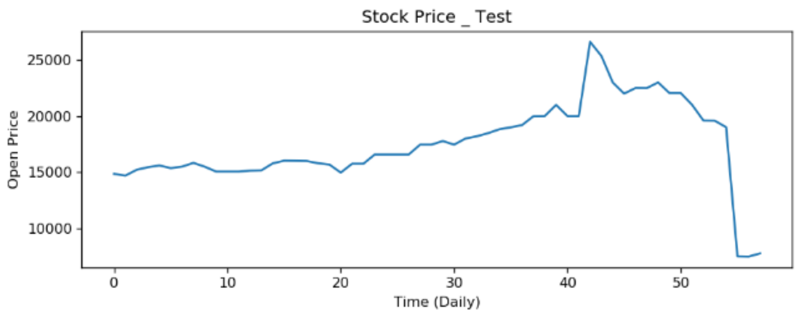


Diagram 3: Test data (day 3201 – 3259)

Diagrams show the fluctuation of data which should be predicted by models.

4.1. ARIMA

Our work is categorized into two parts with the below description.

4.1.1. Finding the order of D, AR, MA

7 http://www.tsetmc.com/Loader.aspx?ParTree=15131F

Augmented Dickey-Fuller (ADF) is the test we used for finding non-stationary in our network. Since the null hypothesis in ADF is non-stationary, the p-value should be less than (0.05) to reject the H_0 and have stationary in time-series. If p-value > 0.05 we go ahead with finding the order of differencing, and If p-value > 0.05, as well as autocorrelations, are positive for 10 or more lags, series needs further differencing. In the case of being neutral for deciding between two orders of differencing, we should go with the order that gives us the least standard deviation. Time-series became stationary after 2 differentiation.

Provided that we are satisfied with stationarity condition, partial autocorrelation function plot (PACF) and autocorrelation function plot (ACF) run to find the model. PACF finds hidden information in the residual which can be modeled by the next lag (AR), while ACF defines how well the present value of the series is related to its past values. (MA)

The model made in this paper is ARIMA (1,2,1) which was supported residuals plot that indicated there were no patterns.

4.1.2. Cross-validation and running model

Although cross-validation is an approach to avoid over/under-fitting, data is small enough to avoid overfitting.[8] (Rashid et al., 2018) Now, we divide the data to assess the power of prediction in ARIMA. Graphs will be reported in the next part.

4.1.3. Data Processing

For any network, we should first import the libraries and data processing related to our work. Here we called NumPy, matplotlib. pylot and panda are the libraries at first. Then we must scale down the data in the same range. This is called featuring scaling, and minmax scaler which is the tool to be called.

The next step goes to creating a data structure. This is probably the most important of creating a model, in which we find the best time steps. It is acquired by repetitive experiments and, 60 was the best result for this experiment. It

[8] Thousands or hundred thousand of data are small data. Millions of data are called as large data.

means we want our network to have a memory of 60 characters and one output. We need to set our data in the form of dependency and independency. By choosing 60 as time steps, data Num.1 to data Num.60 makes the first output. Then, data Num.2 to data Num.61, exports second output and so on. After that, we must reshape data based on RNN.

4.1.4. Building Network

Now we need to call Sequential, LSTM, Dense and Dropout from Keras (backend by TensorFlow or PlaidML[9]). To begin with, we used a sequential layer and then defined LSTM with 50 neurons in each layer. Here we define the hidden layers of the network which is 2. Our dropout for each layer is 20% and the dense layer is just for our output in the last layer. Note that, since we just one output, the unit defines one. For compiling the network, which is the next step, we used "Adam" as an optimizer that works on non-stationary objectives and problems (Kingma and Ba, 2014) and "Mean_Squred_Error" for our loss.

The next step is fitting the model for the training set. Here we run the epochs (number of training times) and batch-size. The batch size explains total number of training that data used. we are not going to give all data to NN at once and instead of that, NN gets the data in several stages and smaller batches. The number of batches equals the number of iterations to complete training. Since 50 in batch-size of our model, the number of iterations for the entire data set is 44. Epoch represents the total number of times; a learning algorithm sees the complete dataset. "Since deep learning uses gradient descent to optimize model, it makes sense to pass the entire dataset through a single network multiple times to update the weights and thus obtaining a better and more accurate prediction model". (Siami-Namini and Namin, 2018)

4.1.5. Making the predictions and visualizing the results

Here was the last part of the model which goes to get the real and predicted stock price and visualizing the results.

[9] If the hardware does not have Nvidia graphic card.

Reports of our studies divided by ARIMA and LSTM are as below:

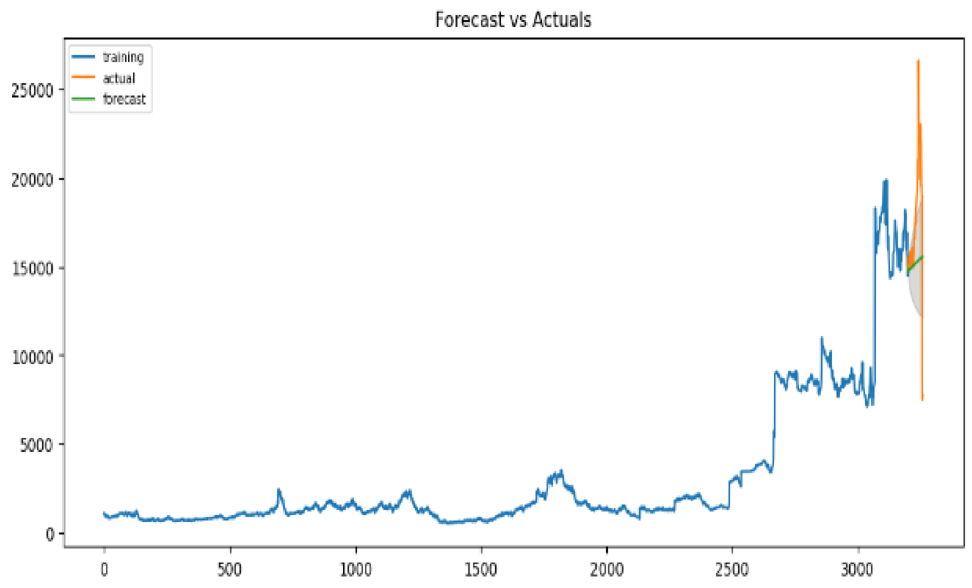


Figure 1: Forecast and Actual data in ARIMA (all data)

5. Experimental Results and Discussion

As you can see, by considering all the data in 10 years, what ARIMA estimates is not out of scope. Prediction in Figure.1 says the total path of fluctuation. It especially works we the prediction does not go more than 2 weeks. But in the long term, it fails in prediction. Figure.2 demonstrates the details of ARIMA prediction for 60 days and in this pierid (based on our data) it is not powerful enough to rely upon on.

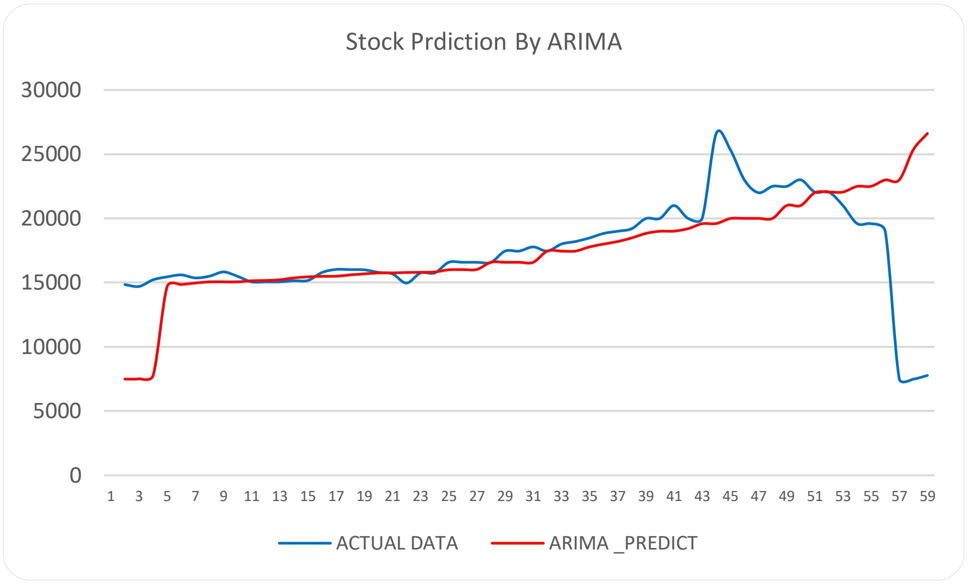


Figure 2: Forecast and Actual data in ARIMA (test data)

Although the prediction accuracy of ARIMA in 60 days equals 18%, by dividing it into three 20-day periods, the error of accuracy equals 9.32%, 3.84%, and 46.95%!

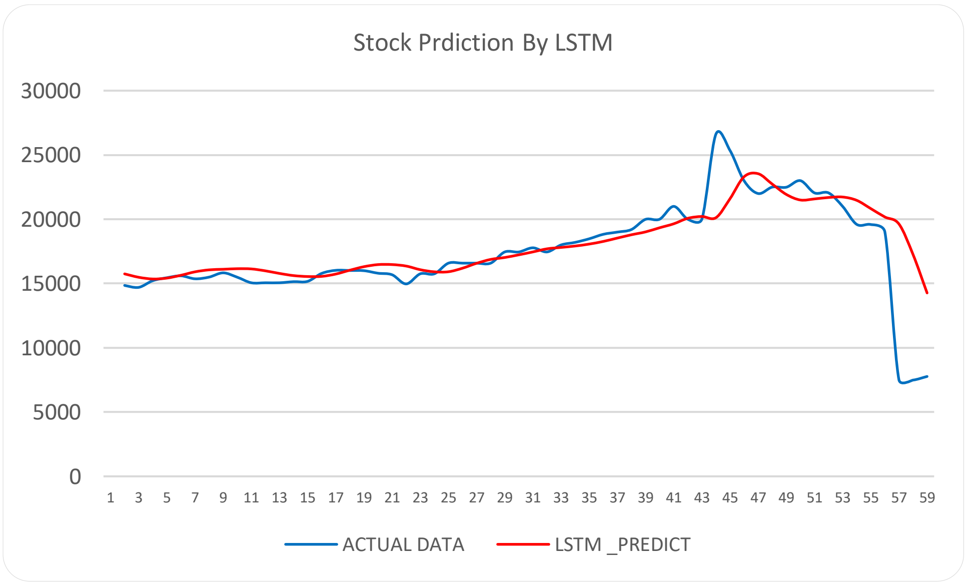


Figure 3: Forecast and Actual data in LSTM (test data)

In LSTM, divided periods for error of accuracy results 3.22%, 2.73% and 25.76% .

As figures 2 and 3 show, although at the beginning of the prediction period, ARIMA was able to manage the trend, as time goes by, the error of forecast increases, while in LSTM (in comparison with ARIMA) we have a better model of prediction in long term.

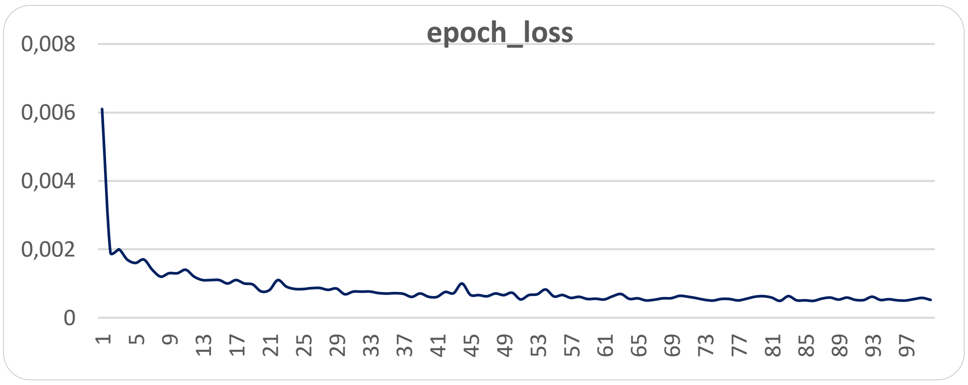


Figure 4: epochs in LSTM

Figure.4 demonstrates the trend of epoch loss in the network. Epochs may not be equal to the number of iterations. The number of best epochs to train a model is not clear. Loss should be decreasing with each epoch; however, it does not mean the higher number of epochs, the better result in loss! As we can see, loss after 25 epochs does not decrease and that's why is the number of epochs we selected for this network, finally. In some research, epoch behavior on prediction reported random. (Siami-Namini and Namin, 2018)

Table below demonstrates the comparison between LSTM and ARIMA in terms of forecast accuracy.

	MAPE	MSE	MAD	TS
LSTM	%10.04	6,334,744	1,181	-22

ARIMA	%19.11	21,215,311	2,238	0

Table 1: Forecast Accuracy

Formulas of methods in table 1, are as below:

$$MAPE\ (Mean\ Absolute\ Percentage\ Error) = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

$$MSE\ (Mean\ Squred\ Error) = \frac{1}{n}\sum_{t=1}^{n}(A_t - F_t)^2$$

$$MAD\ (Mean\ Absolute\ Deviation) = \frac{1}{n}\sum_{t=1}^{n}|A_t - F_t|$$

$$TF\ (Tracking\ Signal) = \frac{\sum_{t=1}^{n}(A_t - F_t)}{MAD}$$

Where A_t and F_t are actual and forecast value.

As it is clear, results in table 1 prove the previous results in this paper and LSTM outperform ARIMA in terms of accuracy of forecasting.

6. Improvements

In this paper, “intraday open price” data was used for predictions, while using other indexes (volume, start, end, etc.) in panel data may improve the performance of the models. Lastly, the hybrid model of panel-var and LSTM or other sophisticated optimization techniques such as extrapolation or other metaheuristic algorithms could report interesting results.

7. Conclusion

Unstable economics system in some developing countries making robust prediction models difficult. With the help of new approaches in RNN, the likelihood of getting a better estimation (without ignoring the nature of data by

Periódico do Núcleo de Estudos e Pesquisas sobre Gênero e Direito
Centro de Ciências Jurídicas - Universidade Federal da Paraíba
V. 9 - Nº 02 - Ano 2020
ISSN | 2179-7137 | http://periodicos.ufpb.br/ojs2/index.php/ged/index

diff and linearization) turns up. In this paper, we applied LSTM for this target and made a forecast model for Tehran stock price. Then compered the accuracy and error of results with ARIMA model. The error of prediction for 20-day triple period in LSTM became 3.22%, 2.73%, and 25.76% while in ARIMA, it resulted 9.32%, 3.84%, and 46.95%! Reports show LSTM outperformed ARIMA in making predictions. Although in the short term both models work well, by increasing the time of prediction, the accuracy of prediction in both models decreases. As well, increasing the number of epochs in LSTM network, after 25, did not have any effect of prediction accuracy.

References:

Mukherjee, C. and White, H. and Wuyts, M. (2013). Econometrics and data analysis for developing countries: Routledge.

Ariyo, A. A. and Adewumi, A. O. and Ayo, C. K. (2014). Stock price prediction using the ARIMA model. Paper presented at the 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation.

Lin, C.-T. and Yeh, H.-Y. (2009). Empirical of the Taiwan stock index option price forecasting model–applied artificial neural network. Applied Economics, 41(15), 1965-1972.

Binner*, J. M. and Bissoondeeal, R. K. and Elger, T. and Gazely, A. M. and Mullineux, A. W. (2005). A comparison of linear forecasting models and neural networks: an application to Euro inflation and Euro Divisia. Applied Economics, 37(6), 665-680.

Wang, C.-C. and Hsu, Y.-S. and Liou, C.-H. (2011). A comparison of ARIMA forecasting and heuristic modelling. Applied Financial Economics, 21(15), 1095-1102.

Hansen, J. V. and Nelson, R. D. (2003). Time-series analysis with neural networks and ARIMA-neural network hybrids. Journal of Experimental & Theoretical Artificial Intelligence, 15(3), 315-330.

Long, W. and Lu, Z. and Cui, L. (2019). Deep learning-based feature engineering for stock price movement prediction. Knowledge-Based Systems, 164, 163-173.

Cao, J. and Li, Z. and Li, J. (2019). Financial time series forecasting model based on CEEMDAN and LSTM. Physica A: Statistical Mechanics and its Applications, 519, 127-139.

Zhang, K. and Zhong, G. and Dong, J. and Wang, S. and Wang, Y. (2019). Stock Market Prediction Based on Generative Adversarial Network. Procedia Computer Science, 147, 400-406.

Kim, T. and Kim, H. Y. (2019). Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data. PloS one, 14(2), e0212320.

Eapen, J. and Bein, D. and Verma, A. (2019). Novel Deep Learning Model with CNN and Bi-Directional LSTM for Improved Stock Market Index Prediction. Paper presented at the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC).

Zhou, X. and Pan, Z. and Hu, G. and Tang, S. and Zhao, C. (2018). Stock market prediction on high-frequency data using generative adversarial nets. Mathematical Problems in Engineering, 2018.

Wang, H. and Guo, Z. and Chen, L. (2019). Financial Forecasting based on LSTM and Text Emotional Features. Paper presented at the 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC).

Yu, S.-S. and Chu, S.-W. and Chan, Y.-K. and Wang, C.-M. (2019). Share Price Trend Prediction Using CRNN with LSTM Structure. Smart Science, 1-9.

Sang, C. and Di Pierro, M. (2019). Improving trading technical analysis with TensorFlow Long Short-Term Memory (LSTM) Neural Network. The Journal of Finance and Data Science, 5(1), 1-11.

Kim, D. and Baek, C. (2019). Factor-augmented HAR model improves realized volatility forecasting. Applied Economics Letters, 1-8.

Chen, S. and Ge, L. (2019). Exploring the attention mechanism in LSTM-based Hong Kong stock price movement prediction. Quantitative Finance, 19(9), 1507-1515.

Borovkova, S. and Tsiamas, I. (2019). An ensemble of LSTM neural networks for high-frequency stock market classification. Journal of Forecasting. Retrieved from https://onlinelibrary.wiley.com/doi/full/10.1002/for.2585

Liang, X. and Ge, Z. and Sun, L. and He, M. and Chen, H. (2019). LSTM with Wavelet Transform Based Data Preprocessing for Stock Price Prediction. Mathematical Problems in Engineering, 2019. Retrieved from http://downloads.hindawi.com/journals/mpe/2019/1340174.pdf

Du, J. and Liu, Q. and Chen, K. and Wang, J. (2019). Forecasting stock prices in two ways based on LSTM neural network. Paper presented at the 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC).

Karmiani, D. and Kazi, R. and Nambisan, A. and Shah, A. and Kamble, V. (2019). Comparison of Predictive Algorithms: Backpropagation, SVM, LSTM and Kalman Filter for Stock Market. Paper presented at the 2019 Amity International Conference on Artificial Intelligence (AICAI).

Pawar, K. and Jalem, R. S. and Tiwari, V. (2019). Stock Market Price Prediction Using LSTM RNN. In Emerging Trends in Expert Applications and Security (pp. 493-503): Springer.

Hiew, J. Z. G. and Huang, X. and Mou, H. and Li, D. and Wu, Q. and Xu, Y. (2019). BERT-based Financial Sentiment Index and LSTM-based Stock Return Predictability. arXiv preprint arXiv:1906.09024. Retrieved from https://arxiv.org/pdf/1906.09024.pdf

Hushani, P. (2019). Using Autoregressive Modelling and Machine Learning for Stock Market Prediction and Trading. Paper presented at the Third International Congress on Information and Communication Technology.

Manurung, A. H. and Budiharto, W. and Prabowo, H. (2018). ALGORITHM AND MODELING OF STOCK PRICES FORECASTING BASED ON LONG SHORT-TERM MEMORY (LSTM). ICIC Express Letters, 12, 1277–1283. Retrieved from http://icicelb.org/ell/contents/2018/12/el-12-12-11.pdf

Karakoyun, E. and Cibikdiken, A. (2018). Comparison of ARIMA Time Series Model and LSTM Deep Learning Algorithm for Bitcoin Price Forecasting. Paper presented at the The 13th Multidisciplinary Academic Conference in Prague 2018 (The 13th MAC 2018).

McNally, S. and Roche, J. and Caton, S. (2018). Predicting the price of Bitcoin using Machine Learning. Paper presented at the 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP).

Rebane, J. and Karlsson, I. and Denic, S. and Papapetrou, P. (2018). Seq2Seq RNNs and ARIMA models for cryptocurrency prediction: A comparative study. SIGKDD Fintech, 18. Retrieved from https://pdfs.semanticscholar.org/c1ec/480f005244a2cfb93f1d3ad15c2d22b864d6.pdf

Huynh, H. D. and Dang, L. M. and Duong, D. (2017). A new model for stock price movements prediction using deep neural network. Paper presented at the Proceedings of the Eighth International Symposium on Information and Communication Technology.

Selvin, S. and Vinayakumar, R. and Gopalakrishnan, E. and Menon, V. K. and Soman, K. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model. Paper presented at the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI).

Li, H. and Shen, Y. and Zhu, Y. (2018). Stock Price Prediction Using Attention-based Multi-Input LSTM. Paper presented at the Asian Conference on Machine Learning.

Yu, S. and Li, Z. (2018). Forecasting stock price index volatility with LSTM deep neural network. In Recent Developments in Data Science and Business Analytics (pp. 265-272): Springer.

Skehin, T. and Crane, M. and Bezbradica, M. (2018). Day ahead forecasting of FAANG stocks using ARIMA, LSTM networks and wavelets.

Kim, H. Y. and Won, C. H. (2018). Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. Expert Systems with Applications, 103, 25-37.

Hiransha, M. and Gopalakrishnan, E. A. and Menon, V. K. and Soman, K. (2018). NSE stock market prediction using deep-learning models. Procedia Computer Science, 132, 1351-1362.

Nagahisarchoghaei, Mohammad and Nagahi, Morteza and Soleimani, Nadia, Impact of Exchange Rate Movements on Indian Firm Performance (2018). International Journal of Finance and Accounting, Vol. 7 No. 4, 2018, pp. 108-121. doi: 10.5923/j.ijfa.20180704.03..

Nagahi, Morteza and Nagahisarchoghaei, Mohammad and Soleimani, Nadia and Jaradat, Raed M., Hedge Strategies of Corporate Houses (February 7, 2018). Journal of Business Administration Research, 7(1), 6..

Baughman, M. and Haas, C. and Wolski, R. and Foster, I. and Chard, K. (2018). Predicting Amazon spot prices with LSTM networks. Paper presented at the Proceedings of the 9th Workshop on Scientific Cloud Computing.

Shah, D. and Campbell, W. and Zulkernine, F. H. (2018). A Comparative Study of LSTM and DNN for Stock Market Forecasting. Paper presented at the 2018 IEEE International Conference on Big Data (Big Data).

Minami, S. (2018). Predicting Equity Price with Corporate Action Events Using LSTM-RNN. Journal of Mathematical Finance, 8(1), 58-63.

Naik, N. and Mohan, B. R. (2019). Study of Stock Return Predictions Using Recurrent Neural Networks with LSTM. Paper presented at the International Conference on Engineering Applications of Neural Networks.

Pai, P.-F. and Lin, C.-S. (2005). A hybrid ARIMA and support vector machines model in stock price forecasting. Omega, 33(6), 497-505.

Nandakumar, R. and R, U. K. and R, V. and Lokeswari, Y. V. (2018). Stock Price Prediction Using Long Short Term Memory. International Research Journal of Engineering and Technology, 05(03). Retrieved from https://www.irjet.net/archives/V5/i3/IRJET-V5I3788.pdf

Fischer, T. and Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2), 654-669.

Shao, X. and Ma, D. and Liu, Y. and Yin, Q. (2017). Short-term forecast of stock price of multi-branch LSTM based on K-means. Paper presented at the 2017 4th International Conference on Systems and Informatics (ICSAI).

Li, J. and Bu, H. and Wu, J. (2017). Sentiment-aware stock market prediction: A deep learning method. Paper presented at the 2017 International Conference on Service Systems and Service Management.

Zhuge, Q. and Xu, L. and Zhang, G. (2017). LSTM Neural Network with Emotional Analysis for Prediction of Stock Price. Engineering Letters, 25(2). Retrieved from https://pdfs.semanticscholar.org/1a08/bcc2e0866e64d7b4d8d7a4c670faa26ab13d.pdf

Bao, W. and Yue, J. and Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PloS one, 12(7), e0180944.

Choi, H. K. (2018). Stock price correlation coefficient prediction with ARIMA-LSTM hybrid model. arXiv preprint arXiv:1808.01560. Retrieved from https://arxiv.org/pdf/1808.01560.pdf

Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing, 50, 159-175.

Pang, X. and Zhou, Y. and Wang, P. and Lin, W. and Chang, V. (2018). An innovative neural network approach for stock market prediction. The Journal of Supercomputing, 1-21.

Pang, X. W. and Zhou, Y. and Wang, P. and Lin, W. and Chang, V. (2018). Stock Market Prediction based on Deep Long Short Term Memory Neural Network. Paper presented at the COMPLEXIS.

Di Persio, L. and Honchar, O. (2017). Recurrent neural networks approach to the financial forecast of Google assets. International journal of Mathematics and Computers in simulation, 11.

Olah, C. (2015). Understanding lstm networks. Retrieved from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

DAVODI, H., SHABANALI, F. H., & KALANTARI, K. (2012). An investigation of technology development barriers in Agricultural Science and Technology Parks of Tehran University.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

Davodi, H., Iravani, H., Fami, H. S., & Ameri, Z. D. (2017). Affecting Factors on Water Resources' Sustainability in case of small holding farmers, Alborz province, Islamic Republic of Iran. Advances in Bioresearch, 8(3).

Box, G. E. and Jenkins, G. M. and Reinsel, G. (1970). Time series analysis: forecasting and control Holden-day San Francisco. BoxTime Series Analysis: Forecasting and Control Holden Day1970.

Rashid, T. A. and Fattah, P. and Awla, D. K. (2018). Using Accuracy Measure for Improving the Training of LSTM with Metaheuristic Algorithms. Procedia Computer Science, 140, 324-333.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Siami-Namini, S. and Namin, A. S. (2018). Forecasting economics and financial time series: Arima vs. lstm. arXiv preprint arXiv:1803.06386. Retrieved from https://arxiv.org/ftp/arxiv/papers/1803/1803.06386.pdf