Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 170 (2020) 1168–1173

Procedia Computer Science

www.elsevier.com/locate/procedia

International Workshop on Statistical Methods and Artificial Intelligence (IWSMAI 2020)
April 6-9, 2020, Warsaw, Poland

# Stock Market Prediction Using LSTM Recurrent Neural Network

Adil MOGHAR[a]* ,Mhamed HAMICHE[b]

[a]University Abdelmalek Essaadi, Morocco
[b]University Abdelmalek Essaadi, Morocco

**Abstract**

It has never been easy to invest in a set of assets, the abnormally of financial market does not allow simple models to predict future asset values with higher accuracy. Machine learning, which consist of making computers perform tasks that normally requiring human intelligence is currently the dominant trend in scientific research. This article aims to build a model using Recurrent Neural Networks (RNN) and especially Long-Short Term Memory model (LSTM) to predict future stock market values. The main objective of this paper is to see in which precision a Machine learning algorithm can predict and how much the epochs can improve our model.



*Keywords:* Recurrent Neural Network; Long Short-Term Memory; Stock Market; forecasting; prediction;

## 1. Introduction

Several studies have been the subject of using machine learning in the quantitative financial, predicting prices of managing and constricting entire portfolio of assets, as well as, investment process, and many other operations can be covered by machine learning algorithms. In general machine learning is a term used for all algorithm's methods using computers to reveal patterns based only on data and not using any programming instructions. For quantitative finance and specially assets selections several models supply a large number of methods that can be used with machine learning to forecast future assets value. This type of models offers a mechanism that combine weak sources

* Adil MOGHAR Tel.: +212617545599
*E-mail address:* adilmoghar@gmail.com


10.1016/j.procs.2020.03.049

of information and make it a strange tool that can be used efficiently. Recently, the combination of statistics and learning models have polished several machine learning algorithms, such as acritical neural networks, gradient boosted regression trees, support vector machines and, random forecast. These algorithms can reveal complex patterns characterized by non-linearity as well as some relations that are difficult to detect with linear algorithms. These algorithms also prove more effectiveness and multi collinearity than the linear regressions ones. A large number of studies is currently active on the subject of machine learning methods used in finance, some studies used tree-based models to predict portfolio returns [4], others used deep learning in the production of future values of financial assets [9] [1]. Also, some authors overviewed the forecasting of returns using of ADaBoost algorithm [10]. Others proceeds to forecast stock returns using unique decision-making model for day trading investments on the stock market the model developed by the authors use the support vector machine (SVM) method, and the mean-variance (MV) method for portfolio selection [6]. Another paper conversed deep learning models for smart indexing [3]. Also, some study has covered a large number of trends and Applications of Machine Learning in Quantitative Finance [2], the literature review covered by this paper consist of return forecasting portfolio construction, ethics, fraud detection, decision making, language processing and sentiment analysis. These models don't depend one long term memory (passed sequences of data), in this regard a class of machine learning algorithms based on Recurrent Neural Network prove to be very useful in financial market price prediction and forecasting. A paper has compares the accuracy of autoregressive integrated moving average ARIMA and LSTM, as illustrative techniques when forecasting time series data. These techniques were executed on a set of financial data and the results showed that LSTM was far more superior to ARIMA [8]. Our paper aim to use ML algorithm based on LSTM RNN to forecast the adjusted closing prices for a portfolio of assets, the main objective here is to obtain the most accurate trained algorithm, to predict future values for our portfolio.

## 2. Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is one of many types of Recurrent Neural Network RNN, it's also capable of catching data from past stages and use it for future predictions [7]. In general, an Artificial Neural Network (ANN) consists of three layers:

1) input layer,
2) Hidden layers,
3) output layer.

In a NN that only contains one hidden layer the number of nodes in the input layer always depend on the dimension of the data, the nodes of the input layer connect to the hidden layer via links called 'synapses'. The relation between every two nodes from (input to the hidden layer), has a coefficient called weight, which is the decision maker for signals. The process of learning is naturally a continues adjustment of weights, after completing the process of learning, the Artificial NN will have optimal weights for each synapses.

The hidden layer nodes apply a sigmoid or tangent hyperbolic (tanh) function on the sum of weights coming from the input layer which is called the activation function, this transformation will generate values, with a minimized error rate between the train and test data using the SoftMax function.

The values obtained after this transformation constitute the output layer of our NN, these value may not be the best output, in this case a back propagation process will be applied to target the optimal value of error, the back propagation process connect the output layer to the hidden layer, sending a signal conforming the best weight with the optimal error for the number of epochs decided. This process will be repeated trying to improve our predictions and minimize the prediction error.

After completing this process, the model will be trained. The classes of NN that predict future value base on passed sequence of observations is called Recurrent Neural Network (RNN) this type of NN make use of earlier stages to learn of data and forecast futures trends.

The earlier stages of data should be remembered to predict and guess future values, in this case the hidden layer act like a stock for the past information from the sequential data. The term recurrent is used to describe the process of using elements of earlier sequences to forecast future data.

RNN can't store long time memory, so the use of the Long Short-Term Memory (LSTM) based on "memory line" proved to be very useful in forecasting cases with long time data. In a LSTM the memorization of earlier stages

can be performed trough gates with along memory line incorporated. The following diagram-1 describe the composition of LSTM nodes.

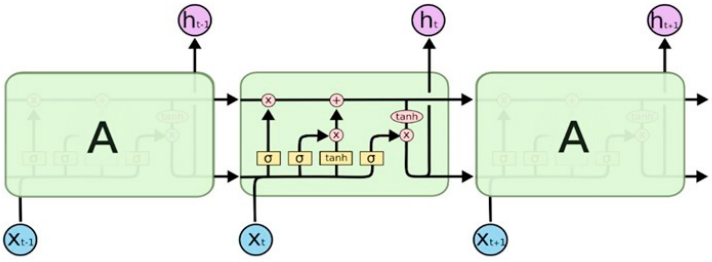



Figure 1. The internal structure of an LSTM [5].

The ability of memorizing sequence of data makes the LSTM a special kind of RNNs. Every LSTM node most be consisting of a set of cells responsible of storing passed data streams, the upper line in each cell links the models as transport line handing over data from the past to the present ones, the independency of cells helps the model dispose filter of add values of a cell to another. In the end the sigmoidal neural network layer composing the gates drive the cell to an optimal value by disposing or letting data pass through. Each sigmoid layer has a binary value (0 or 1) with 0 “let nothing pass through”; and 1 “let everything pass through.” The goal here is to control the state of each cell, the gates are controlled as follow:

- Forget Gate outputs a number between 0 and 1, where 1 illustration “completely keep this”; whereas, 0 indicates “completely ignore this.”

- Memory Gate chooses which new data will be stored in the cell. First, a sigmoid layer “input door layer” chooses which values will be changed. Next, a *tanh* layer makes a vector of new candidate values that could be added to the state.

- Output Gate decides what will be the output of each cell. The output value will be based on the cell state along with the filtered and freshest added data.

## 3. Methodology and data

The data in this paper consist of the daily opening prices of two stocks in the New York Stock Exchange NYSE (GOOGL and NKE) extracted from yahoo finance, for GOOGL our data series cover the period going from 8/19/2004 to 12/19/2019 and for NKE the data cover the period from 1/4/2010 to 12/19/2019.

To build our model we are going to use the LSTM RNN, our model uses 80% of data for training and the other 20% of data for testing. For training we use mean squared error to optimize our model. Also, we used different Epochs for training data (12 epochs, 25 epochs, 50 epochs and 100 epochs) our model will be structured as follow:

Table 1: the LSTM model summery

| Layer (type) | Output Shape | Parameters |
|---|---|---|
| lstm_1 (LSTM) | (None, 50, 96) | 37632 |
| dropout_1 (Dropout) | (None, 50, 96) | 0 |
| lstm_2 (LSTM) | (None, 50, 96) | 74112 |
| dropout_2 (Dropout) | (None, 50, 96) | 0 |
| lstm_3 (LSTM) | (None, 50, 96) | 74112 |
| dropout_3 (Dropout) | (None, 50, 96) | 0 |
| lstm_4 (LSTM) | (None, 96) | 74112 |
| dropout_4 (Dropout) | (None, 96) | 0 |
| dense_1 (Dense) | (None, 1) | 97 |
| Total number of parameters : 260,065<br>Trainable parameters : 260,065<br>Non-trainable parameters :0 | | |

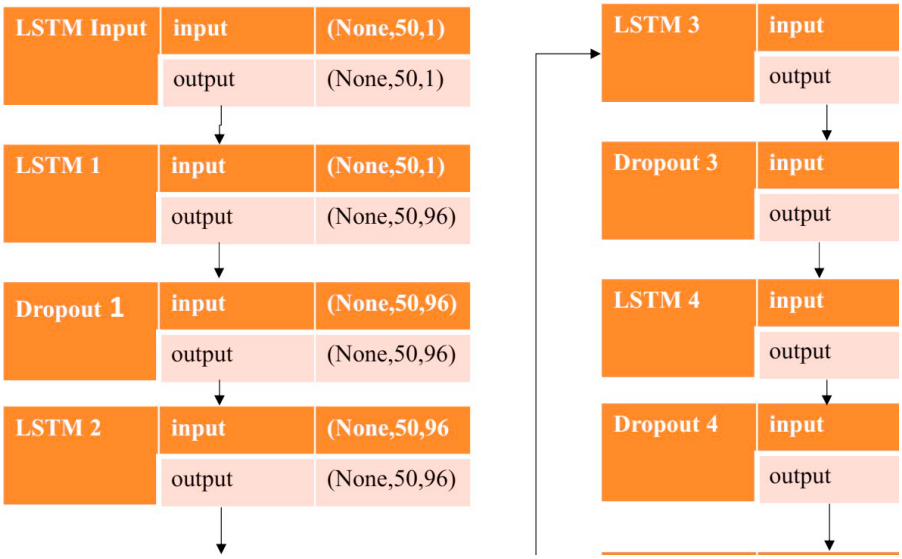



Figure 2: the LSTM model structure

## 4. Result and discussion:

After training our NN the result of our testing has shown different results, the number of epochs as well as the length of the data have both significant impact on the result of testing. For example, if we changed the dataset for NKE giving it a time set going from 12/2/1980 to 12/19/2019 the results will appear as follow:

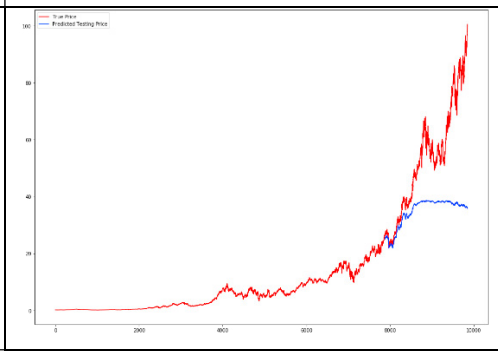
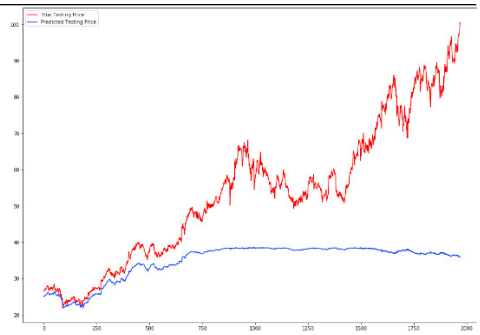
| Epochs | NKE | NKE zomed view |
|---|---|---|
| 25 Epochs |  |  |

Figure 3: result of training for the NKE stocks with different dataset time

After observing our data, we can see that at first the data was less volatile and have lower values, in the figure, the red lines represent the real market value and the blue lines represent the predicted price value, after the NKE start peeking bigger values, the asset become more volatile, then the nature of this asset changed. In our case is better to avoid this type of change. Our model has lost trace of opening prices around 600 to 700 day of testing which conform the change in data nature. The result for our dataset for different number of epochs is giving by the following Figure:

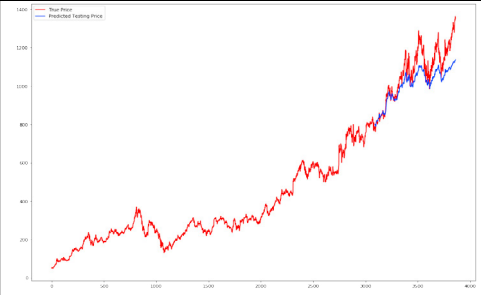
| Epochs | GOOGL | NKE |
|---|---|---|
| 12 Epochs |  |  |
| 25 Epochs |  |  |
| 50 Epochs |  |  |
| 100 Epochs |  |  |

Figure 4 : Result of training for NKE and GOOGL stocks with different number of epochs

For different data set we can observe that training with less data and more epochs can improve our testing result and at the same time allow us to have beater forecasting and prediction values. The following table shows the precision of our training and testing for all the epochs for both NKE and GOOGL asset price.

Table 2: the value of loss for GOOGL and NKE for deferrent number of epochs

| | GOOGL | | NKE | |
|---|---|---|---|---|
| | processing Time / sec | Loss | processing Time / sec | Loss |
| 12 epochs | 264 | 0.0011 | 132 | 0.0019 |
| 25 epochs | 550 | 0.001 | 275 | 0.0016 |
| 50 epochs | 1100 | 6.57E-04 | 550 | 0.001 |
| 100 epochs | 2200 | 4.97E-04 | 1100 | 8.74E-04 |

The table 2 above confirms that the precision of our forecasting increases if we add more epochs of training for our model.

## 5. Conclusion

This paper proposes RNN based on LSTM built to forecast future values for both GOOGL and NKE assets, the result of our model has shown some promising result. The testing result conform that our model is capable of tracing the evolution of opening prices for both assets. For our future work we will try to find the best sets for bout data length and number of training epochs that beater suit our assets and maximize our predictions accuracy.

## References

[1] Batres-Estrada, B. (2015). Deep learning for multivariate financial time series.

[2] Emerson, S., Kennedy, R., O'Shea, L., & O'Brien, J. (2019, May). Trends and Applications of Machine Learning in Quantitative Finance. In 8th International Conference on Economics and Finance Research (ICEFR 2019).

[3] Heaton, J. B., Polson, N. G., & Witte, J. H. (2017). Deep learning for finance: deep portfolios. Applied Stochastic Models in Business and Industry, 33(1), 3-12.

[4] Moritz, B., & Zimmermann, T. (2016). Tree-based conditional portfolio sorts: The relation between past and future stock returns. Available at SSRN 2740751.

[5] Olah, C. (2015). Understanding lstm networks–colah's blog. Colah. github. io.

[6] Paiva, F. D., Cardoso, R. T. N., Hanaoka, G. P., & Duarte, W. M. (2018). Decision-Making for Financial Trading: A Fusion Approach of Machine Learning and Portfolio Selection. Expert Systems with Applications.

[7] Patterson J., 2017. Deep Learning: A Practitioner's Approach, O'Reilly Media.

[8] Siami-Namini, S., & Namin, A. S. (2018). Forecasting economics and financial time series: Arima vs. lstm. arXiv preprint arXiv:1803.06386.

[9] Takeuchi, L., & Lee, Y. Y. A. (2013). Applying deep learning to enhance momentum trading strategies in stocks. In Technical Report. Stanford University.

[10] Wang, S., and Y. Luo. 2012. "Signal Processing: The Rise of the Machines." Deutsche Bank Quantitative Strategy (5 June).