Stock Market Behaviour Prediction using Stacked LSTM Networks*

Samuel Olusegun Ojo*, Pius Adewale Owolawi†, Maredi Mphahlele‡ and Juliana Adeola Adisa§
Department of Computer Systems Engineering,
Tshwane University of Technology
Pretoria, South Africa
Email: *samuelojo88@gmail.com, †owolawipa@tut.ac.za, ‡mphahlelemi@tut.ac.za, §adeolaa33@gmail.com

***Abstract*—Predicting the behavior of the stock market has been an area that has attracted the interest of many researchers particularly in the field of Machine Learning and time series analysis. The ability to analyse an entity such as the stock market that appears to lack synchronicity yet seemingly influenced by historic events remains an open issue in research. This study adopts a stacked Long Short Term Memory network model for predicting stock market behabiour. The data used is composed of historic stock market data from the American Stock Exchange, NASDAQ Composite (ÎXIC). Results obtained show that by making use of a stacked Long Stort Term Memory network model, future stock market behavior can be predicted.**

***Index Terms*—stock market prediction, time series, LSTM**

I. Introduction

The stock market can be described as a highly volatile entity, with great sensitivity to various types of parameters. Much research has been conducted in the way of stock market prediction, with various models associated with time series analysis being adopted for this purpose. In economics, theories such as the efficient market hypothesis and the random walk theory challenge the thought that future stock price can be predicted using historic stock price data [1] [2]. With the availability of historic and current stock market data, researchers have attempted to predict the future state of the stock market through models that can analyse historic stock market data. Typically, this historic data consists of opening, closing, high, and low prices, and volume. Using various time series analysis models, daily data can be analysed and the future state of the stock market can be predicted.

Machine Learning (ML) techniques have gained popularity more recently for time series analysis and stock market prediction, due to their ability to predict future states given large historical datasets [3]. Time series analysis is divided into two types:

- Univariate Analysis - The analysis of a time series which has a single input parameter, over equally spaced time intervals.
- Multivariate Analysis - The analysis of a time series with multiple input variables which vary over time.

Stock market analysis can be considered as Multivariate due to the multiple parameters, which vary over time, involved in its analysis. These parameters include, but are not limited to, time, opening price, closing price, among others. For an investor/trader these parameters have an impact on their decision making.

This study looks at various approaches to time series analysis that have gained popularity over the years, and reviews literature focussing on stock market behavior prediction. A model for stock market behavior prediction is presented making use of stacked Long Short Term Memory networks along with the results obtained after this model was trained on NASDAQ Composite (ÎXIC) data.

II. Background

Time series forecasting is not a new area of research. A number of techniques have gained popularity within the sphere of time series analysis and forecasting, particularly for stock market prediction. Some of these include:

A. AutoRegressive Integrated Moving Average

AutoRegressive Integrated Moving Average (ARIMA) was developed from the Box Jenkins method and aims to predict the future value of a variable based on the previous values of the same variable. ARIMA has the best performance with linear and largely stationary time series data, and generally requires at least 50 historical data events to work effectively [4]. With the ARIMA model, the forecasted value of a variable is deduced through a combination of historic values and can be represented by the equation [5]:

$$y_t = \theta_0 + \phi_1 y_{t-1} + \varepsilon_t - \theta_1 \varepsilon_{t-1} \tag{1}$$

where y_t is the future value, ε_t represents a random error at time t, ϕ_t and θ_t are coefficients, p represents an integer known as autoregressive polynomial, and q too is an interger known as a moving average polynomial [5].

B. Support Vector Machines

The Support Vector Machine (SVM) algorithm has also had success in time series forecasting. It is a supervised learning technique which classifies training samples in a hyperplane by separating samples into one class or the other [6]. In a 2-D plane, this would be a line which assigns training samples to one of two classes, on either side of the line. This divider is known as a descision boundary. The decision boundary can be depicted by the following equation [7]:

$$w.x + b = 0 \tag{2}$$

where x represents a poin on the descision boundary, w is a n-dimentional vector at 90^o to the descision boundary, and b is a constant. SVMs are also popular for their ability of avoid the problem of overfitting [8].

C. Multilayer Perceptron

Multilayer Perceptron (MLP) is a type of Artificial Neural Network (ANN), made up of at least 3 layers namely, an input layer, a hidden layer, and an output layer. It makes use of supervised learning through backpropagation for training [9]. Training of MLPs is achieved through a self correcting approach in which the result of each hidden layer (known as a weight) is backpropagated and these weights updated iteratively to improve the prediction prerformance of the network. The most common means of evaluating the performance of an ANN is by calulating the Mean Squared Error (MSE) and Mean Absolute Diviation (MAD). Figure one illustrates a MLP with 3 layers.

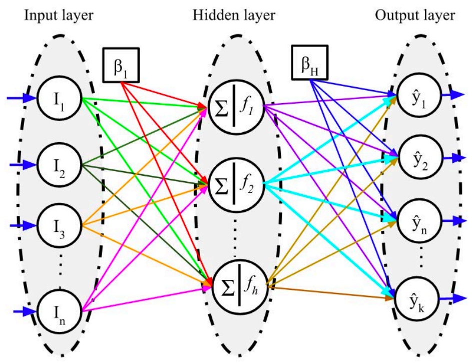

Figure 1. MLP Neural Network [10]

D. Recurrent Neural Networks

A Recurrent Neural Networks (RNNs) is a type of ANN which make use of internal hidden neurons. A longstanding problem with RNNs is the vanishing gradient problem in which data is lost during the learning process as the internal state of an RNN is updated, recurssivley. The update of a RNN's state can be represented by the following equation:

$$h_t = \sigma(Wx_t + Uh_{t-1} + b) \tag{3}$$

where N represents the number of neurons in the RNN, $x_t \in \mathbb{R}^M$ is the input state and $h_t \in \mathbb{R}^N$ is the hidden state at a time t. $W \in \mathbb{R}^{NxM}$ is the weight of the current input, $U \in \mathbb{R}^{NxN}$ is the weight of the recurrent input, and $b \in \mathbb{R}^N$ is the bias. σ is an activation function. Figure 2 illustrates the structure of a RNN.

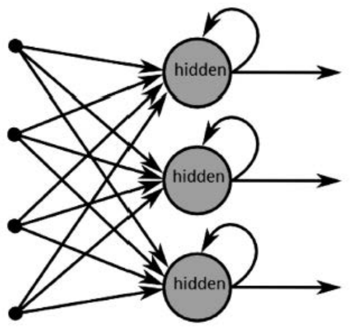

Figure 2. Recurrent Neural Network [11]

E. Long Short Term Memory Networks

Long Short Term Memory (LSTM) network is a type of RNN which has gained popularity in time series / sequential analysis. They consist of an input layer, a hidden layer, and an output layer. The hidden layer of a LSTM network contains memory cells which in turn contains three gates which are responsible for updates to it's cell state. These three gates are: an input gate, and output gate, and a forget gate. Unlike RNNs, LSTM networks do not suffer from the vanishing gradient problem which is an important factor in stock price prediction as previous information being processed in the neural network will affect the future/subsequent information [12]. Figure three illustrates the structure of a LSTM network with its 3 gates. The output of the gates determine the values of the updated

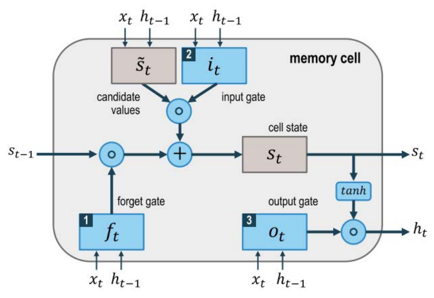

Figure 3. Long Short Term Memory Network [13]

cell state. This is represented by the following equations [14]:

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{4}$$

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{5}$$

$$c_t = \tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{6}$$

$$o_t = sigma(W_o[h_{t-1}, x_t] + b_o) \tag{7}$$

$$h_t = sigma * \tanh(c_t) \tag{8}$$

where x_t and h_t are input and output vectors respectively, f_t is a vector representing the forget gate, c_t represents the cell state vector, i_t is the vector of the input gate, o_t is the vector of the output gate, and W, b represent the parameter matrix and vector.

F. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have been used in applications relating to image processing and clssification, text detection, object tracking etc [15]. A basic CNN has 3 layers, namely, a convolutional layer, a pooling layer, and a fully connected layer. In stock price prediction, CNNs have been used to learn from graphical representations of stock price data such as cadlestick charts [16] [17]. Figure four shows the general structure of a CNN with its various layers.

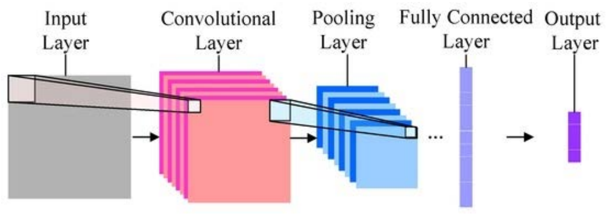

Figure 4. Convolutional Neural Network [18]

CNNs analyse pixel information through a matrix known as a filter (or kernel) which slides through blocks of the input layer in strides to form the convoluted layer. Each resulting pixel of the Convoluted layer is calculated by the following equation, where C_k is the kth pixel of the convoluted layer, x is the pixel value corresponding to C_k, W is the coefficient vector, and b is the bias [18].

$$C_k = f(x * W + b) \tag{9}$$

The kth value of the pooling layer is calculated as follows [18]:

$$P_k = f(\beta * down(C) + \alpha) \tag{10}$$

where P_k is the kth value in the pooling layer matrix, C is the value vector from the convoluted layer, β is the coefficient, and α is the bias.

The Pooling Layer is responsible for down-sampling an image. A popular approach to pooling is known as Max Pooling, where the maximum value is calculated for a block in the matrix. The Max Pooling value can be calculated as follows [18]:

$$down(C) = \max\{C_{s,l} | |s| \leq \frac{m}{2}, |l| \leq \frac{m}{2}, s, l \in z^+\} \tag{11}$$

where $C_{s,l}$ is the pixel value at C of the matrix, and m is the size for subsampling.

III. Related Work

In [4], the use of ARIMA for stock price prediction using historic data from the Pakistan company OGDCL was presented. The daily closing prices were used as observations from 23 January 2004 to 19 November 2018, amounting to 3632 data items. The results showed that for short term forecasting, ARIMA was effective, having an error estimate of 0.63 in monetary units.

In a study by [5], ARIMA and SVM are combined in a hybrid model for stock price forecasting. By combining ARIMA and SVM, the study aimed to combat the shortcomings of ARIMA in it's inability to effectively analyse non-linear and non-stationary data. In analysing the model, the stock data of 10 companies was examined and the model results were compared against other models which make use of either only ARIMA or SVM. The results show that the hybrid model outperformed the other models for predicting future stock market data.

The authors in [12] adopted an ensemble model of LSTM networks in predicting stock market data and compares the results obtained to that of a multilayer LSTM network. The proposed model approches the ensemble learning process by dividing the test data into a number of smaller data sets using the Bagging algorithm. The smaller datasets are then trained on a number of LSTM networks. The Bagging algoritm is then used to obtain a prediction result from the multiple test results. Data from the Shanghai Composite Index and Shenzen Composite Index, among others, were used. Results of the study showed that the amsemble LSTM network model had an increased accuracy of 11.7%, a precission increase of 8.5%, and a recall rate increase of 10%. The ensemble LSTM network model had a greater predictive accuracy than the multilayer LSTM network.

IV. Methodology

The dataset used in this study consists of daily stock data for the NASDAQ Composite (ÎXIC) obtained through Yahoo Finance. The data is for the period 01 January 2009 to 19 July 2019. Figure five shows a scaled plot of date versus the closing stock price of NASDAQ Composite (ÎXIC) for the period.

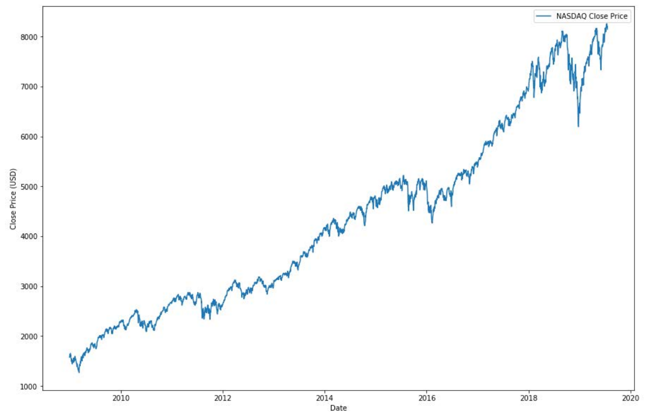

Figure 5. NASDAQ Composite (ÎXIC) 01 January 2009 - 19 July 2019

A stacked LSTM model is trained on the data, and evaluated by measuring the Mean Squared Error (MSE) and Mean Absolute Diviation (MAD) of the model. 90% of the data is used as training while the remaining 10% is used as testing and validation.

The following equations are used to evaluate the model:

$$MSE = \frac{1}{n}\sum_k (r_k - y_k)^2 \quad (12)$$

$$MAD = \frac{1}{n}\sum_k |r_k - y_k| \quad (13)$$

n is the number of observations, y_k is the output of the kth observation, and r_k is the kth output of the model [19].

In developing the models, Python 3 is used as the programming language of choice with libraries that include pandas, numpy, and sklearn. The Integrated Development Environment used is Spyder. The model was developed and run on a MacBook Pro with a 2.2GHz Intel Core i7 processor, and 16 GB 1600 MHz DDR3 memory.

Stacked LSTM networks provide a deeper model for learning and are composed of multiple hidden layers of LSTMs. These multiple hidden layers act as a Deep Recurrent Neural Network (DRNN). Figure six shows the stacked LASTM model adopted by this study, with an input, 2 LSTM layers, and an output.

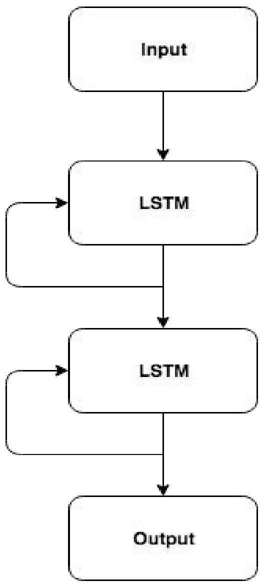

Figure 6. Stacked LSTM model. Adapted from [20]

In this study, a stacked LSTM model is used for prediction with 100 neurons for the first LSTM layer, 50 neurons for the second LSTM layer, and 200 epochs.

An important aspect of training a LSTM network is the iterative updating of weights using the training data. To achieve this, the Adaptive Movement Estimation (Adam) algorithm is used for stochastic gradient descent [21].

V. Results

Figure seven shows the plot of the cost function for the validation data and cost function for the training data, for 200 epochs.

The actual and predicted behavior of the closing price for the NASDAQ Composite (ÎXIC) is depicted in Figure eight below, showing the stacked LSTM model's ability to closely predict the stock market behavior based on historic data it was trained on.

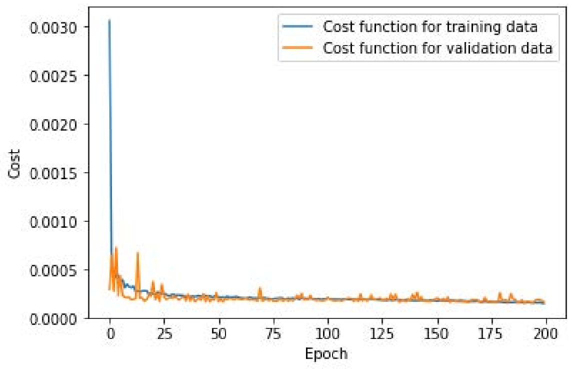

Figure 7. Cost function for validation and training data

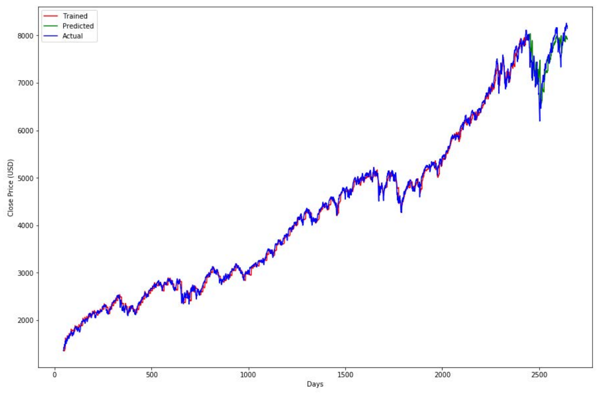

Figure 8. Actual and Predicted NASDAQ Composite (ÎXIC) Behaviour

Results obtained showed the model produced an accuracy of 53.6% in predicting future stock market behavior.

Table one shows the values obtained for MSE and MAD for the stacked LSTM model.

TABLE I. Mean Squared Error and Mean Absolute Diviation

Measurement	**Value**
MSE	0.0022
MAD	0.0495

VI. Conclusion and Future Work

This study proposes the use of a stacked LSTM network model for predicting stock market behavior, using data from NASDAQ Composite (ÎXIC) . The model was trained and the results obtained show that the model was able to predict stock market behavior with some accuracy. An open issue remains in that the volatility of the stock market cannot be mitigated using only historic data, but factors of the present also need to be analysed including current news in the world of politics and economics that could affect the behavior of investors and ipso facto the behavior of stock markets.

Acknowledgment

The authours of this study thank Alexandre Xavier for his contribution in the form of a stock market prediction tutorial (Predicting stock prices with LSTM), and code that formed a part of the methodology and experimentation for this study.

References

[1] V. Atanasov, C. Pirinsky, and Q. Wang, "The efficient market hypothesis and investor behavior," 2018.

[2] S. Rehman, I. U. Chhapra, M. Kashif, and R. Rehan, "Are stock prices a random walk? an empirical evidence of asian stock markets," *ETIKONOMI*, vol. 17, no. 2, pp. 237–252, 2018.

[3] S. Agrawal, D. Thakkar, D. Soni, K. Bhimani, and C. Patel, "Stock market prediction using machine learning techniques," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 1099–1103, 04 2019.

[4] M. Afeef, A. Ihsan, and H. Zada, "Forecasting stock prices through univariate arima modeling," 2018.

[5] P.-F. Pai and C.-S. Lin, "A hybrid arima and support vector machines model in stock price forecasting," *Omega*, vol. 33, no. 6, pp. 497–505, 2005.

[6] D. Karmiani, R. Kazi, A. Nambisan, A. Shah, and V. Kamble, "Comparison of predictive algorithms: Backpropagation, svm, lstm and kalman filter for stock market," in *2019 Amity International Conference on Artificial Intelligence (AICAI)*. IEEE, 2019, pp. 228–234.

[7] E. Ahmadi, M. Jasemi, L. Monplaisir, M. A. Nabavi, A. Mahmoodi, and P. A. Jam, "New efficient hybrid candlestick technical analysis model for stock market timing on the basis of the support vector machine and heuristic algorithms of imperialist competition and genetic," *Expert Systems with Applications*, vol. 94, pp. 21–31, 2018.

[8] S. Sharma and B. Kaushik, "Quantitative analysis of stock market prediction for accurate investment decisions in future," *Journal of Artificial Intelligence*, vol. 11, pp. 48–54, 2018.

[9] S. Jain, M. Kain, and N. Singh, "Prediction for stock marketing using machine learning," *Bhagwan Parshuram Institute of Technology*, p. 1, 2018.

[10] A. A. Heidari, H. Faris, I. Aljarah, and S. Mirjalili, "An efficient hybrid multilayer perceptron neural network with grasshopper optimization," *Soft Computing*, pp. 1–18, 2018.

[11] W. De Mulder, S. Bethard, and M.-F. Moens, "A survey on the application of recurrent neural networks to statistical language modeling," *Computer Speech & Language*, vol. 30, no. 1, pp. 61–98, 2015.

[12] Q. Xie, G. Cheng, X. Xu, and Z. Zhao, "Research based on stock predicting model of neural networks ensemble learning," in *MATEC Web of Conferences*, vol. 232. EDP Sciences, 2018, p. 02029.

[13] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.

[14] S. Selvin, R. Vinayakumar, E. Gopalakrishnan, V. K. Menon, and K. Soman, "Stock price prediction using lstm, rnn and cnn-sliding window model," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2017, pp. 1643–1647.

[15] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

[16] T. Kim and H. Y. Kim, "Forecasting stock prices with a feature fusion lstm-cnn model using different representations of the same data," *PloS one*, vol. 14, no. 2, p. e0212320, 2019.

[17] R. M. I. Kusuma, T.-T. Ho, W.-C. Kao, Y.-Y. Ou, and K.-L. Hua, "Using deep learning neural networks and candlestick chart representation to predict stock market," *arXiv preprint arXiv:1903.12258*, 2019.

[18] M. Peng, C. Wang, T. Chen, and G. Liu, "Nirfacenet: A convolutional neural network for near-infrared face identification," *Information*, vol. 7, no. 4, p. 61, 2016.

[19] E. Guresen, G. Kayakutlu, and T. U. Daim, "Using artificial neural network models in stock market index prediction," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10 389–10 397, 2011.

[20] U. Singh, S. Chauhan, A. Krishnamachari, and L. Vig, "Ensemble of deep long short term memory networks for labelling origin of replication sequences," in *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2015, pp. 1–7.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.