LSTMSA: A Novel Approach for Stock Market Prediction Using LSTM and Sentiment Analysis

Abhijoy Sarkar[1]
Artificial Intelligence
IIT, Ropar, India
abhijoy.sar@gmail.com

Abhaya Kumar Sahoo[2]
School of Computer Engineering
KIIT, Deemed to be University, India
abhayakumarsahoo2012@gmail.com

Sourav Sah[3]
School of Computer Engineering
KIIT, Deemed to be University,India
sourav29sah@gmail.com

Chittaranjan Pradhan[4]
School of Computer Engineering
KIIT, Deemed to be University, India
chitaprakash@gmail.com

Abstract—**Stock market prediction is one of the most popular use cases for machine learning models. A general model that can predict the rise and fall of stocks is an arduous task as there maybe multifarious factors that can affect stock prices. This paper attempts to create a model by emulating the approach traders, investors and analysts take to evaluate stock investment strategy. A conjunction of both technical analyses using available numerical data about stocks and fundamental analysis using news headlines are attempted to understand and predict market behavior for the Google stock. For this purpose, sentiment analysis is used to understand news data regarding the stock along with existing time series data as input for an LSTM neural network. It is observed that such an approach yields a more intuitive and accurate yet generalized model that can be used for prediction of the stock market.**

Keywords—LSTM, Neural Network, Sentiment Analysis, Stock Market

I. INTRODUCTION

The stock market is a very volatile environment. The capability to predict the returns accurately for different stocks can minimize risk and maximize earning potentials for traders, investors and brokers alike. Prediction can enable smart purchase of stocks at their lowest possible price and the selling the highest possible price at a time window decided by the trader [2,7,14]. A mathematical model would thus demystify the entire process of stock evaluation if done the right way.

This paper aims to taken into account the correlation of news headlines in influencing the fluctuations in stock prices. It performs sentiment analysis on this data to aid a fundamental analysis of the market. Further, for technical analysis, 60 days worth of stock price data is used as a context at every time step to train the model in a supervised way. All of this data (from 1st January 2014 to 31stDecember 2018) is fed into an LSTM network to get a prediction for the month of January 2019. This approach is compared to a naive approach of only using LSTM without sentiment analysis of news headlines [11,16]. This is accompanied by describing the methodology undertaken for the approach and discussion of the corresponding results. The paper will conclude by drawing associations between the results of this research and prior research. Contributions can be summarized as follows:

(i) Studies how news data can comprehensively influence fluctuations in the stock market.

(ii) Provides an optimal data structure that can be used for recurrent neural networks. This comprises the news data along with stock price data. Data structures suitable for the purpose have been run against several tests to present the optimal data structure.

(iii) Takes long term contextual time series data to make predictions.

(iv) Emphasizes importance of trend prediction over absolute prediction attempts for the stock market.

The remaining parts of the paper are well organized as follows: the section 2 explores related work. This section stresses on why it is not possible to accurately predict the absolute value of stock prices and the implied limitations in making a prediction model for the same. Section 3 highlights the proposed solution to the problem which is implemented in section 4. Section 4.1 and Section4.2 are concerned with conducting the fundamental analysis phase for the experiment, while section 4.3 aims to draw upon its results and perform technical analysis using LSTM based neural networks. Finally, section 5 discusses the lessons learned from the performance study and review show introduction of intuition in the model affects accuracy. Section 6 shows the conclusion with future work.

II. RELATED WORK

A. Second Order Caos Event and the Brownian Motion

In financial engineering, the Brownian motion Osborne (1959) for stock market suggests that the future variations of the stock prices are independent of the past behaviors of the market. The change in stock market price is also classified as a second order chaos event Thomas (2002). It means that building an absolute prediction system for stock market is not viable as the stock market can react to the predictions made upon it [1,2,4]. A paradox is thus introduced and the prediction model may behave very differently with respect to the actual market as the market acts upon the predictions made in order to maximize its decision outcome[9,13].

B. Naive Approaches

The reasons for failure of past approaches to predict the stock market can be classified as follows:-

Attempting absolute prediction of stock price

As mentioned in section 2.1- definitive prediction is not possible and hence all approaches that seek to predict actual stock prices do not perform generically well for a given test set Siew (2012). What can be predicted however is the trends in the stock market and prediction models should hence aim to predict the same.

Exclusive Technical and Fundamental Analysis

Technical analysis involves using charts, numerical data, and other statistics of a company in the past to predict its future stock price. On the other hand fundamental analysis involves using following market trends, news headlines, blog posts, qualitative analysis of a stock's worth to predict its future price Deshmukh (2016); Kalyani et al. (2016); Bollena et al.(2011); Zhang et al. (2011). Prior work has not yet tried to combine both these approaches to yield a better result.

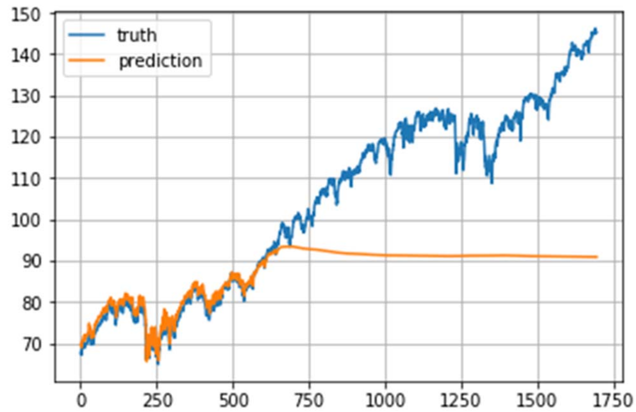

Fig.1. RNN to predict absolute values instead of relative change rates.

In the figure 1, the point where the prediction line bends to achieve some constant value is where the test set values are used for prediction. As evident from the graph, the model performs poorly on the test set.

Non consideration of long term dependencies

Trends in the stock market show a correlation to a previous sequence of prices of the stock over a time period. Naive approaches may aim to predict only the close prices of the stock based on a particular day's other market price values such as open price only. In such a scenario, analyzing the change in prices over duration of time steps can critically improve the accuracy of the model[3,12].

III. Proposed Work

In this paper, a supervised learning methodology is employed to train a recurrent neural network. For this purpose, we use 5 years of worth of Google stock price, i.e., from 1st January 2014 to 31st December 2018. Only opening and closing prices are used for the model. For each day in the mentioned duration, we collect the headlines of 10 relevant news articles regarding the Google stock. In order to use this data we need to convert it to a numerical form. We calculate a polarity score for the 10 news headlines per day and append this data to the dataset consisting of stock prices. An LSTM's ability to harness long term dependencies is utilized to take the prepared dataset as input. For any batch of training, the input takes in a window of 60 previous days or time steps to learn the corresponding closing price for the day. Hyper parameters are appropriately tuned for the training purpose as described in section 4.3.3. The trained LSTM model is then used to make generalized predictions about Google stock prices from 1st January to 31st January2019.

IV. Experimemt and Result Analysis

A. Data Preparation

The stock prices form a sequential time series of length n, defined as [p0,q0], [p1,q1], ..., [pn−1,qn−1] in which pi is open price on day i and qi is the close price on day i, given $0 \le i \le N$. The corresponding input Xi for any given i^{th} day shall consist of values from [pi−60, qi−60] to [pi, qi]. The input Xi is given as:-

$$Xi = ([pi-60,qi-60],[pi-59,qi-59],...,[pi,qi]) \qquad (1)$$

Applying Equation (1) to all possible values of i for Xi a data structure can be created for feeding into the LSTM model.

$$X60 = ([p0, q0],[p1,q1],...,[p59,q59])$$
$$X61 = ([p1, q1],[p2,q2],...,[p60,q60]) \qquad (2)$$

$$Xn = ([pn-60, qn-60],[pn-59,qn-59],...,[pn, qn])$$

The input vector X forms a 3D data structure of dimension n−60×60×2.The data structure thus formed consists of 60 time steps for every data point to be used for training.

Since a supervised learning model is being deployed, creation of another vector yi is done, which shall consist of the close price for the i^{th} day. Hence, size of vector y is given as n − 60.

B. Sentiment analysis on news data

i) News data preprocessing

Event Registry API is used to extract the top 10 articles relevant to the Google stock on a particular day over the time period of 1st January 2014 to 31st December 2018. Only alphabetic characters are retained in the news headlines. All characters are converted to lowercase inputs to overcome double identical words with a different capitalization. Following this, stop words are removed and stemming is performed on the remaining headlines Bird and Klein (2009).

ii) Polarity incorporation to dataset

After the preprocessing is performed as in section 4.2.1, the data is ready for sentiment analysis Bird and Klein (2009). A polarity score is calculated for the 10articles for each day and is appended to the Google stock market dataset. It consists of open and close prices from 1st January 2014 to 31st December 2018. This appended polarity data forms the fundamental analysis component for the prediction model. The dataset in Equation (1), takes the form: -

$$Xi= ([pi-60,qi-60],[pi-59,qi-59],...,[pi, qi],[\mu i, \mu i]) \quad (3)$$

μ_i, in Equation (3), refers to the polarity score of 10 news headlines relevant to the Google stock. The value μ_i is repeated to conform to the data structure dimensions in equation (2) to get a final dimension of$n-60\times61\times2$after the addition of polarities to the dataset.

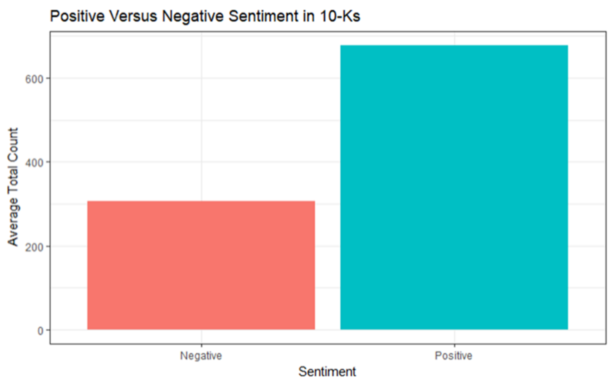

Fig.2. Sentiment overview of the dataset

It can be noted from figure 2, that the news headlines of the firm naturally tend toward positivism, with the amount of positive polarities being approximately double the amount of negative polarities calculated for each day Deshmukh(2016); Kalyani et al.(2016).

C. Long Short Term Memory (LSTM)

Recurrent Neural Network (RNN)

Stock market prices form sequential data, i.e., every value can be mapped to a specific instance in time and can conjointly work to predict data for a future time step. Their current neural network (RNN) Chung (2015) has self-loops in its hidden layer(s), to enables RNN to use the previous state of the hidden neuron(s) to learn the current state given the new input. However, this RNN architecture in itself cannot serve the purpose for an optimal model due to the vanishing gradient problem, which is beyond the scope of this paper[5,6,10].

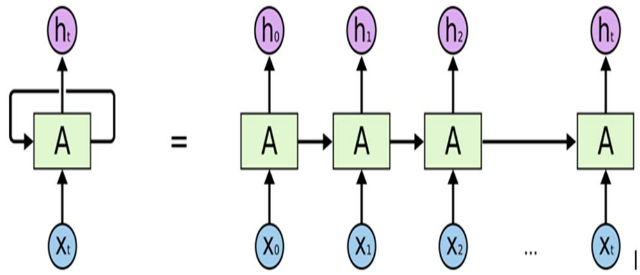

Fig.3. An unrolled recurrent neural network

In the figure 3, X_t forms the input at time step t and h_t is the calculated hypotheses at each time step t. A denotes the hidden layer(s) of neurons for the network. (3)

LSTM Architecture

LSTM Schmid huber (1997) uses the concept of a memory cell, that consists of many more computational components apart from the conventional artificial neurons in the hidden layer of the network. In these memory cells, states are passed along through time and eliminate the problem of the vanishing gradient. LSTM cell help effectively use states recall long term memories and input in the current time step, to make sequence based predictions. Long short-term memory (LSTM) cell is a specially designed working unit that helps an RNN better memorizes the long-term context[11,16].

There may be a distant gap between the inputs used by the recurrent neural network (RNN) for predicting outputs. This makes tracing and calculation of back propagation computation difficult. In order to keep track of the states that influence output, RNN can be represented in an unrolled version which shall consist of a fixed number of time steps used for calculation. The model is then trained on this finite approximation of the RNN. This can be implemented by feeding inputs of length equivalent to the number of time steps being considered as in equation (1) and performing a backward pass after every such input block.

Model Construction

Parameters defined in Table 1 are used for creating the LSTM model.

Optimizer: Using an optimizer greatly affects how fast an algorithm converges to the minimum value. It performs much better than the traditional gradient descent algorithm and avoids getting stuck in the local minima the of cost function. This paper uses the Adam optimizer Kingma and Ba (2014) which is mostly recommended for building RNN architectures or other sequential models. The Adam optimizer combines the power of two other optimizers: gradient descent with momentum Qian (1999) and RMSprop Hinton.

Let denote the gradient at time step.

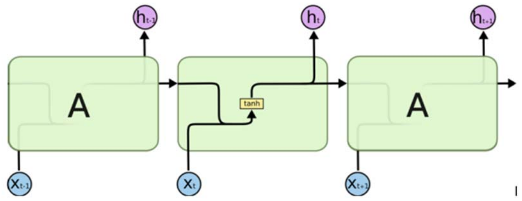

Fig.4. The repeating module in a standard RNN containing a single layer.

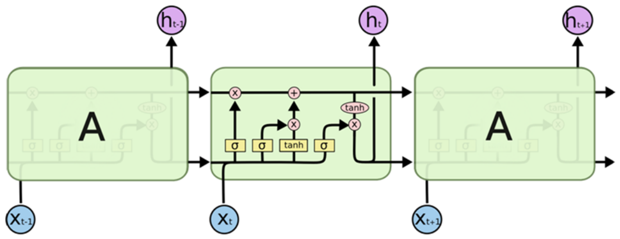

Fig.5. The repeating module in an LSTM contains four layers along with the input, forget and output gate

Table 1. LSTM Model Initialization Parameters

Parameter	Value	Description
lstm_size	50	Number of Units in one LSTM layer
num_layers	4	Number of stacked LSTM layers
keep_prob	0.8	Percentage of cell units to keep in the dropout operation
init_learning_rate	0.001	The learning rate to start with
learning_rate_decav	0.0	Decay ration in later learning epochs
max_epoch	100	Total number of epochs in training
input_size	1	Size of one training data point
batch_size	32	Number of data points to use in one mini-batch

Model Construction

Parameters defined in Table 1 are used for creating the LSTM model.

OPTIMIZER

Using an optimizer greatly affects how fast an algorithm converges to the minimum value. It performs much better than the traditional gradient descent algorithm and avoids getting stuck in the local minima the of cost function. This paper uses the Adam optimizer Kingma and Ba (2014) which is mostly recommended for building RNN architectures or other sequential models. The Adam optimizer combines the power of two other optimizers: gradient descent with momentum Qian (1999) and RMSprop Hinton[5,8].

Let g_t denote the gradient at time step t.

Then $g_{t,i}$ is the partial derivative of the cost function with respect to the parameter θ_i at time step t:

$$g_{t,i} = \nabla_{\theta_t} J(\theta_{t,i}) \qquad (4)$$

Adaptive Moment Estimation, or Adam, is a method that computes the learning rates that may change dynamically for each parameter by considering the exponentially weighted moving average of squared gradients in past time steps and the exponentially weighted moving average of past gradients. This can be represented as

$$v_t = \beta v_{t-1} + (1-\beta) g^2_t \qquad (5.a)$$

$$m_t = \beta m_{t-1} + (1-\beta) g_t \qquad (5.b)$$

The v_t and m_t can be considered as the estimates of the first moment (mean) and second moment (variance) of the gradients respectively; hence getting the name Adaptive Moment Estimation. A bias correction is performed for the to solve the problem of exponentially weighted moving averages in Equation (5.a) and Equation (5.b) starting towards 0, using the following estimates:

$$\widehat{V}_t = \frac{v_t}{1-\beta^{t2}} \qquad (6.a)$$

$$\widehat{m_t} = \frac{m_t}{1-\beta^{t1}} \qquad (6.b)$$

Equation (6.a) and Equation (6.b), where m_t is the exponentially decaying average of past gradients and v_t is the past squared gradients, are then used to update the parameters, which yields the Adam update rule

$$\theta_{t+1} = \theta_t - \int_{v^t}^{\alpha} \widehat{m_t} \qquad (7)$$

Where α is the learning rate and ε is added so that the denominator term in Equation (7) is not 0. The values for the Adam optimizer used for the LSTM model are:

Following is a summary of the benefits of the Adam optimizer used for this paper:

1. The learning rate is calculated for every weight and after every iteration.
2. The learning does not diminish exponentially.
3. The gradient update uses the moments of the distribution of weights, allowing for a faster descent towards minima and lower noise during parameter optimization.

Regularization

Training the model for generalized predictions and to avoid over fitting of data points, a regularization term must be added to the hypothesis. This causes a penalty for larger weight values to render a general fitting curve. This paper attempts to use the Tikhonov regularization Hlaváckov-Schindler (2008),which aims to solve the minimization problem[6].

Dropouts

Another method explored in this paper for the purpose of preventing over fitting considers randomly dropping a number neurons based on some predefined probability[15]. The LSTM model in this paper attempts to set a dropout of 20% for its purpose. This forces the model to not be overly dependent on any particular of neuron or groups of neurons, and considers every single one of them for generalized predictions.

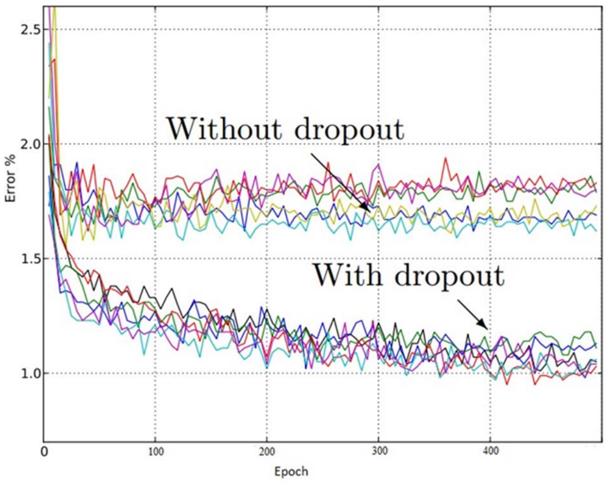

Fig. 6. Validation Error % vs Epoch.

From figure 6 , it can be seen that adding a dropout during training significantly generalizes the model and reduces error during validation.

Result Analysis

The LSTM model predictions when plotted against close prices without taking news headlines into consideration as in Figure 2, shows a general trend. Though accurate in terms direction depicted for the stock price, i.e., upwards through time, it fails to wholly mimic the shape of the trend to be effective for any short term decision-making.

This problem however is solved in Figure3, which takes the sentiment analysis of news headlines into consideration for training the LSTM model. The shape of the predicted prices is seen to conform to the real prices. The accuracy is to be measured graphically in for the purpose of this paper. Use of error calculation methods such as mean square will not be able to represent the results as accurately since absolute predictions are being intentionally avoided for reasons provided in section2.2.1.

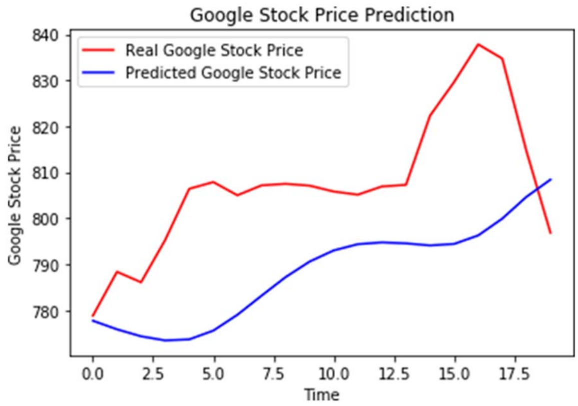

Fig. 7. Applying RNN on dataset without polarity data from sentiment analysis of news headlines.

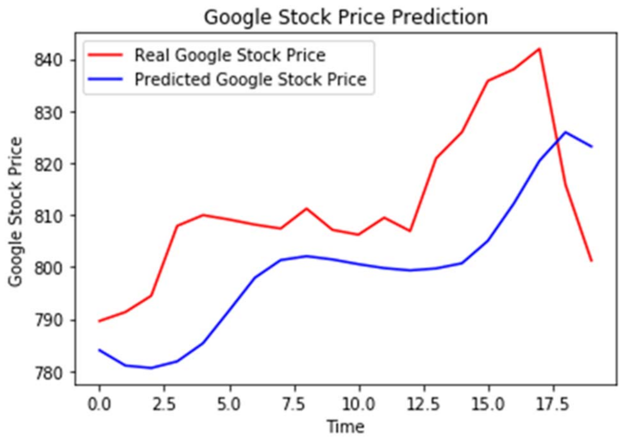

Fig. 8. Applying RNN on dataset with polarity data from sentiment analysis of news headlines.

V. CONCLUSION

Item-based collaborative filtering recommendation system is better to use when users are far greater than the number of items. A single generic market model is yet to be developed which can predict all market nuances and fluctuations. In fact, the lack of a tautological panacea would be a much-desired outcome to avoid problems mentioned in section2.1,witha generalized rule that can assist all investors make better decisions without necessarily influencing the market itself in a substantial way. Anamalgam of technical and fundamental analysis for stock market utilizes the aptitude of an arbitrary professional trader bout reading management, understanding cash flows, and for timing the market with short-term.

Simple sentiment analysis has the capability of explaining 40% of the variance in market returns and validates the use of alternative data for serious consideration in investing. Hence, numerical analysis using these news headlines along with a recurrent neural network architecture capable of evaluating time series data with long term dependencies makes for one of the most optimal and intuitive approaches to market prediction.

VI. REFERENCES

[1] Bird, S. & Klein, E. L. E. (2009). Natural Language Processing with Python. O'Reilly Media Inc.

[2] Bollena, J., Maoa, H., & Zengb, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, 2.

[3] Chung, J. (2015). Gated feedback recurrent neural networks.

[4] Deshmukh, D. K. R. R. (2016). Sentiment analysis of news headlines for stock price prediction.

[5] Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12.

[6] Hlaváčková-Schindler, K. (2008, September). Tikhonov regularization parameter in reproducing kernel Hilbert spaces with respect to the sensitivity of the solution. In International Conference on Artificial Neural Networks (pp. 215-224). Springer, Berlin, Heidelberg.

[7] Kalyani, J., Bharathi, H. N., & Jyothi, R. (2016). Stock trend prediction using news sentiment analysis. *CoRR*, 1607(1958).

[8] Kingma, D. & Ba, J. (2014). "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980.

[9] Osborne, M. F. M. (1959). Brownian Motion in the Stock Market. *Operations Research*, 7(2), 145–173

[10] Qian, N. (1999). On the momentum term in gradient descent learning algorithms, volume 12.

[11] Schmidhuber, S. H. (1997). Long short-term memory.

[12] Siew, H. L. (2012). Regression techniques for the prediction of stock price trend. In *2012 International Conference , 10-12 Sept. 2012, Langkawi, Malaysia*, volume on Statistics in Science, Business and Engineering (ICSSBE). IEEE.

[13] Thomas, C. M. (2002). Chaos Theory versus the Efficient Market Hypothesis in Financial Markets.

[14] Zhang, X., Fuehres, H., & Gloor, P. A. (2011). Predicting Stock Market Indicators Through Twitter "I hope it is not as bad. *Social and Behavioral Sciences*, 26.

[15] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), 1929-1958.

[16] Chen, C., & Xia, L.(2015). Recurrent neural network and long short-term memory.