Stock Market Prediction Using AI

Nishant Ranjan Verma, MSc. Big Data Analytics and AI ,Under the supervision of Dr. Shagufta Henna,
Letterkenny Institute of Technology, Letterkenny, CO. Donegal

***Abstract*—Artificial Intelligence is the agent of change for the 21st Century. The algorithms under it's belt can execute and complete tasks which are even challenging for the experts of the fields. The field of Finance is no different. Stock Market has been the centre of many research for past many years. It is both complex and challenging for the researchers around the globe. The single reason being the accuracy. In here we will try to make use of such algorithms and tools from the domain of machine learning and deep learning and try to measure the outcome of the stocks. We will use stacked Long Short-Term Memory (LSTM), which is a type of Artificial Recurrent Neural Networks. Also I will try to present a comparison with other ML algo to seek if we could have done it without the LSTM.**

***Index Terms*—LSTM, ANN, RNN, FE(Feature Engineering), Deep Learning(DL), Machine Learning(ML), Root Mean Square Error (RMSE)**

=

I. Introduction

PRedicting stock cost or money related business sectors has been perhaps the greatest test to the AI community. Be that as it may, none of these methods or blend of procedures has been effective enough. To a great extent past the capacity of conventional AI research which has for the most part engaged on creating savvy frameworks that should copy human insight. By its tendency the stock market is generally unpredictable (non-linear) and unstable. Simultaneously, the change in the financial exchange is very serious. In like manner, the information is noisy. It is critical to keep the risk low when making a market prediction. The main objective of this project is to see and find if AI with the help of algorithm help us in predicting the stock market price. We will primarily use LSTM with other algorithm compared as we progress.

[1]

A. RNN

RNN are first of its kind algorithms that can memorize past inputs to memory, after big set of data is provided to it. It is a generalization from feed forward network which comprise of internal memory. It is recurring by nature since it executes similar function for each input of data whereas the outcome of the present input depends on the previous computation. Once the outcome is received it, it is then copied and reverted back into recurrent networks. Inorder to make a decision it takes view from the present i/p and o/p which it has observed from past i/p. In NN all i/p are independent from one another. But in case of RNN they are related. These loops make them quite tedious and typical but after study we can say they are just a bit different then normal NN. A RNN is basically then a multiple sets of similar network where each one is sending message to previous.

[1]https://github.com/nishantrv/Project

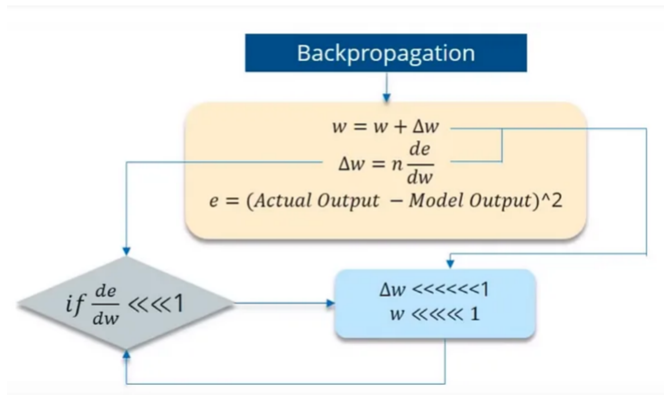

Fig. 1. Vanishing Gradient

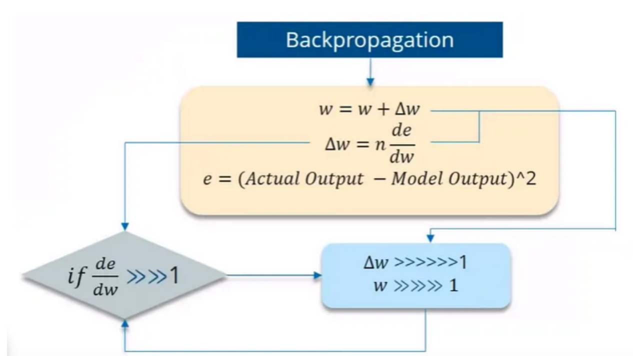

Fig. 2. Exploding Gradient

B. Vanishing Gradient (VG)

Data passes from NN through i/p neurons to o/p neurons whereas the errors are re-checked and sent back to network in order to update weights. While training, cost function(e) evaluates and compares the your o/p to the required o/p. In case partial derivation of error is lower to 1 then after the

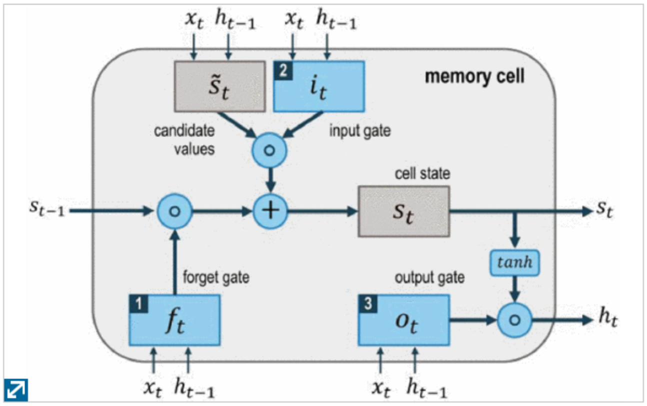

Fig. 3. LSTM Network

$$
\begin{aligned}
f_t &= \sigma(W_f.\,[h_{t-1},\ x_t] + b_f) && (4)\\
i_t &= \sigma(W_i.\,[h_{t-1},\ x_t] + b_i) && (5)\\
c_t &= \tanh(W_c.\,[h_{t-1},\ x_t] + b_c) && (6)\\
o_t &= sigma(W_o[h_{t-1},\ x_t] + b_o) && (7)\\
h_t &= sigma * \tanh(c_t) && (8)
\end{aligned}
$$

Fig. 4. LSTM Equation

multiplication with learning rate, whose outcome is already less, will only generate a significant change in comparison to the iteration. Therefore, for VG the more we proceed in the network the lower our gradient becomes and thus harder it is to train weights, which in a way gives a domino effect for every other weight through the network.

C. Exploding Gradient(EG)

We mention EG when algorithm aligns outrageous high importance to weights without no reason. EG are issues in which massive error gradients combine and results in giving serious big updates to NN model weights while training which causes the model to being unstable and unable to learn from training data. But luckily we can easily solve this by truncating or squashing the gradients.

D. LSTM

LSTM is kind of RNN, gained significance in time series queries. They are made by input, hidden and outpur layer. The hidden layer consists of memory cells having three gates that base to it's cell state. Input Gate, Out Gate and Forget are the three gates. Unlike RNN, LSTM donot have encounters with vanishing gradient issue that is a critical component, since past information being stored in NN will impact the future information. The updated cell values relies on the output of the gates. Equation:

Where xt and ht are respectively input,output vectors, ft is vector forget gate, ct is the vector of the cell state, Ot is input gate vector, Ot is the output gate vector, and W,b is matrix and vector parameter.

II. Literature Review

Sirignano and Cont [1] used a DL model train on universal feature collection of financial market. The DS consisted of selling, purchasing records in transactions, dissolution for around 1000 NASDAQ stocks from purchase logs in stock exchange. The NN was based on 3 layers i.e. LSTM units and ReLUs from feed forward, combined with optimization by algorithm stochastic gradient descent (SGD). This model generalized and covered stocks apart from the already present in training data. Though there were advantages of it, the training cost was still costly. Also because of tedious programming of the DL algorithm, it was not certain if the unwanted features came into being while feeding data to it. Authors realized that it would a good way to perform feature selection instead of proceeding to training the model and found a better way to minimize the computational complexity.

McNally [2] used the RNN, LSTM for predicting price of Bitcoins by using Boruta Algorithm for optimizing the FE portion. Besides that made use of Bayesian Optimization for selecting LSTM params. Bitcoin data was from 19th August 2013 to 19th July 2016. They invested on multiple optimization and improved results of DL algorithms. The main issue of its work was over

tting. This study of it has some common elements as to stock market price prediction. Unknown elements and noises present in price DS are major obstactle to complete it in a right manner. The most interesting portion of this paper is FE and optimization.

Fischer and Krauss in [3] has used LSTM in

nancial market prediction. Their dataset was of SnP 500 index. Had gathered months last list for SnP from Dec 1989 to Sep 2015, then formed binary matrix. They have RMSprop as optimizer. The major focus of research was researcheres implemented current deep learning technique to execute p redictions. They had minimal knowl- edge of the financial sector but they believed in LSTM. Though LSTM was better then DNN and logistic regression.

Bao, Yue and Rao [4] did one splendid research in the field of DL. They said that predictive framework comprises of 3 portions. 1st they applied wavelet transformation (WT) to the financial data. After this they passed de-noised data through stacked autoendcoders in order to produce sensible features and then implemented LSTM for forcasting. They make use of daily observations of 6 stocks and evaluated the profitablity of models by buying future stocks. They depicted that in terms of accuracy and prediction, their model provided better results then other models of same types.

Piramuthu[6] researched a complete testing of feature selection approaches for data mining approaches. He made use of datasets which comprised of credit approval, web traffic, kiang data, tam, loan default and concluded how different feature selection customized the decision tree (DT) outcomes. The key aspect of his paper was the results after comparing both probabilistic distance based and oter inter class feature selection ways. Add to it using different datasets, his research was strengthened. Though it was entirely based on DT.

Weng[7] concentrated on stock price prediction by using ensemble techniques of 4 of the ML algos. The dataset was comprised of 5 sets of data. They obtained it from 3 open sourced APIs and R package named TTR. Model used was NNRE, Random Forest, AdaBoost,SVRE. A complete study of ensemble techniques designed specifically for prediction. After that they applied 8 technical validators and evaluated them based on 5 datasets gathered. The main highlight of this paper is that it has created platform for intrested investor with R, in which the users don't need to provide data rather data is fetched through an API, directly online. In regards to research, they evaluated and judged prediction from 1 to 10 days. Also it can't do it more then 2 weeks and less then 1 day. The limitation of it would be that it only worked on US stocks.

III. Dataset

The proposed model's data collection will be accomplished by using Apple (CSV) 20 year stock data imported through pandas-datareader's Tiingo library. Tiingo is a tracing platform that provides complete end to end cost/prices on equities, ETFs and MFs. Once the data is loaded then it will split accordingly into test and train data. It will comprise of following columns namely: symbol, date, close, high, low, open, volume, adjClose, adjHigh, adjLow, adjOpen, adjVolume, divCash, splitFactor.[5] The data collected will then be used for forecasting the stock market.

IV. Methodology

A. Baseline Model

I tried implementing the model for stock price prediction using Linear Regression method and inferred that RMSE value is higher, which clearly showed that linear regression has performed poorly.

B. Main Model

Collection of Stock Data: Pandas-datareader's Tiingo library through an API Pre-processing of the Data: Create a method consisting of the dataset and times-stamps values. Training Data Set and Test Data Set Evaluation. Creation of Stacked LSTM Model: Reshaping of previous results. Import libraries like Sequential, Dense and LSTM from tensorow and create Stacked LSTM Model. Compile it with Mean Square Error. Fit the train and test datset (Use Epoch) Prediction of Test Data Predict Outcome: Transform back to original form and calculate RMSE. Will plot the test data and train data graph. Final Prediction of Future 30 days and Plotting it's outcome: Write the logic for setting up machine to predict data for next 30 days on the basis of it.[5]

V. Comparison Model

This will include the Baseline model's Linear Regression and I will load the dataset file Tata.csv. After importing the basic libraries will set up the figure size and then move on to normalizing the data. Then set the index as date values and sort to create a separate dataset. Proceed on with create multiple features as required. Split into train and validation. Then I will apply the other algorithms one after another: Auto ARIMA: Import auto arima from pyramid.arima. Place the model on the

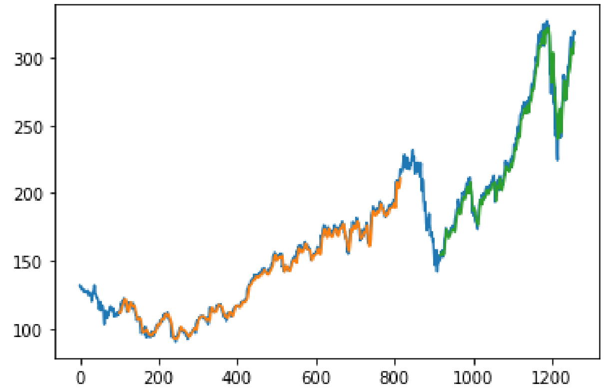

Fig. 5. Performance - Train Plots and Test Plots

train and test data. After that Fit the training and predict using periods = 248. Find the performance through the RMSE value and Plot. PROPHET: From fbprohet, I will import Prophet. Create a newdataframe as per the requirements. Prepare the data. Train and Validation establishment. Fit the model. Make the predictions. After this proceed on to the RMSE and Plot. LSTM: Same as earlier on the main model.

VI. Evaluation Metric

After doing the prediction for x-train and x-test data, I applied scaler inverse trans- form in order to calculate the performance metric i.e. RMSE. RMSE helps us in determining if the prediction on x-train and x-test have provided relevant t. Lower the value of it the better prediction.

VII. Results and Analysis

The study allows as to make use of the stacked LSTM in order to predict stock market.The model is trained, then evaluated by measuring RMSE of the model. 65 percentage of the data is for training while 35 percentage is testing and validation. After training and obtaining the results, I have achieved some accuracy. Also I was able to predict the future 30 days and then plot the output. While making the comparison model, I deduced that for Linear Regression have higher RMSE value. The auto ARIMA model has given has better results in comparison to LR but still not close to the real ones. As per the plot the trend has been taken into consideration but not give idea about the seasonality. Prophet model have been successful in capturing the trend and seasonality from past data. And performs well on time series dataset. As the above dataset has major focus on seasonality therefore, it is not much helpful. But the LSTM model has multiple elements which proved helpful like changes in number of LSTM Layers, adding dropout value or increasing the number of epochs.

VIII. Future Works

Following can be the future directions:- a. Enriching our dataset. Adding more volumes of data to see if the prediction hold b. Exploring advanced models which has seem to have gained stride in the recent years in order to better my results.

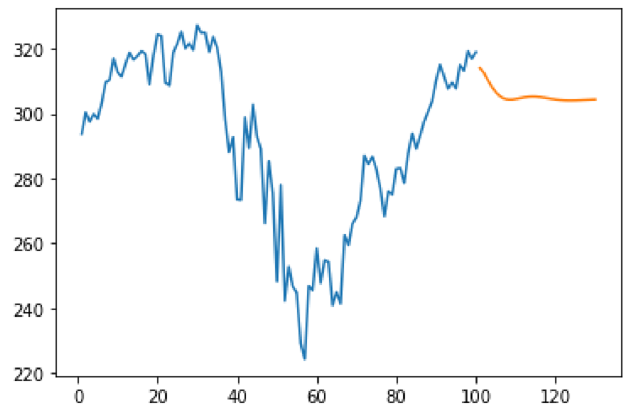

Fig. 6. Result (1) - 30 days prediction

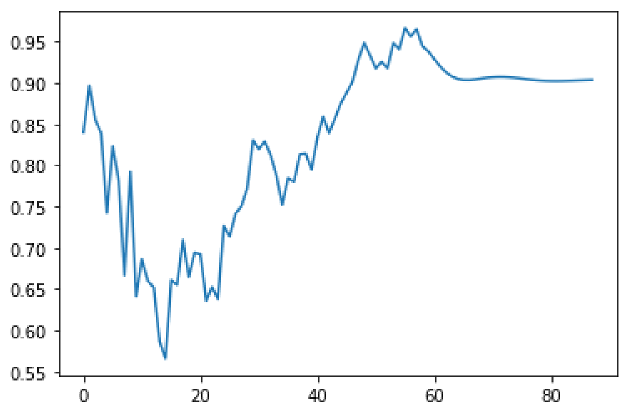

Fig. 7. Result (2) - 30 days prediction

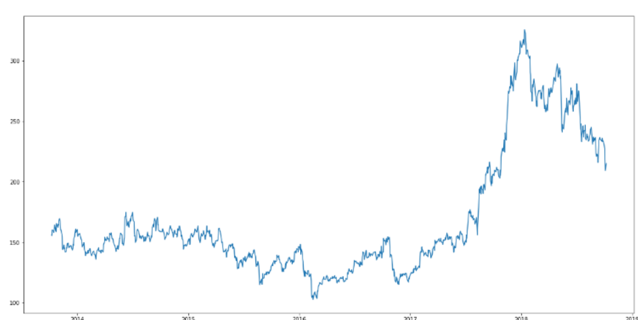

Fig. 8. Comparison Plot

References

[1] Sirignano J, Cont R. Universal features of price formation in nancial markets: perspectives from deep learning. Ssrn. 2018.

[2] McNally S, Roche J, Caton S. Predicting the price of bitcoin using machine learning. In: Proceedings—26th Euromicro international conference on parallel, distributed, and network-based processing.

[3] Bao W, Yue J, Rao Y (2017) A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PLoS ONE 12(7): e0180944. https://doi.org/10.1371/journal.pone.0180944

[4] Fischer T, Krauss C. Deep learning with long short-term memory networks for financial market predictions. Eur J Oper Res. 2018;270(2):654–69. https://doi.org/10.1016/j.ejor.2017.11.054.

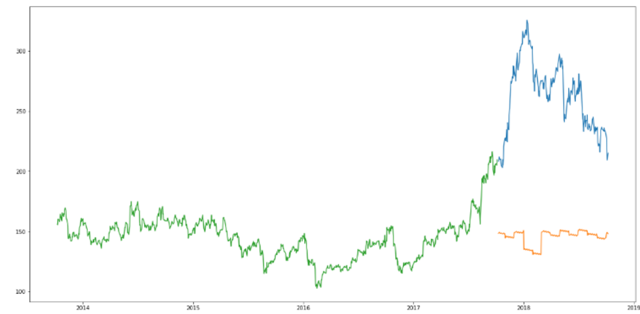

Fig. 9. Linear Regression - Comparison

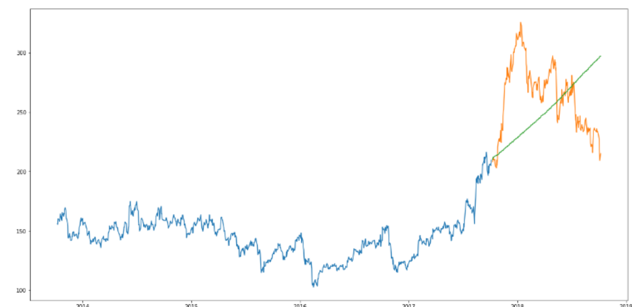

Fig. 10. Auto ARIMA - Comparison

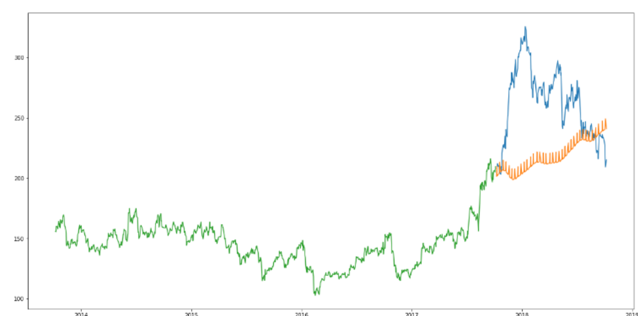

Fig. 11. PROPHET - Comparison

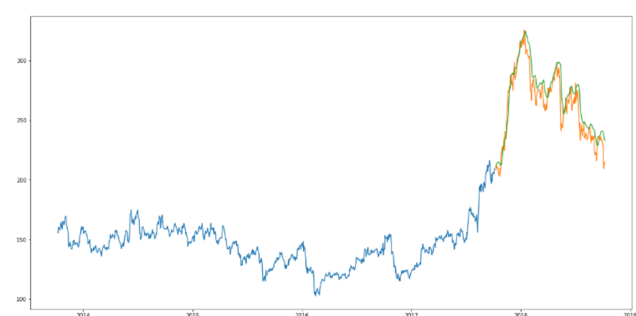

Fig. 12. LSTM - Comparison(fig12)

[5] Krishna Naik https://github.com/krishnaik06/Stock-MArket-Forecasting.

[6] Piramuthu S. Evaluating feature selection methods for learning in data mining applications. Eur J Oper Res. 2004;156(2):483–94.

[7] Weng B, Lu L, Wang X, Megahed FM, Martinez W. Predicting short-term stock prices