Stock Market Predictions Using Deep Neural Networks (LSTM)

Abdul Rehman Khan[1], Sumeria Begum[2] , Mohd Abdul Rahman[3] and Mohammad Rahmat Ali[4]

[123]B.E STUDENT, [4]ASSISTANT PROFESSOR

Department of CSE

ISL Engineering College, Bandlaguda, Hyderabad.

ABSTRACT- Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. The successful prediction of a stock's future price could yield significant profit. There are a bunch of different approaches through which the Problem can be solved as there are many prebuilt algorithms such as: ARIMA, SARIMAX, FB Prophet etc.Through this project, our goal is to compare a regression based Approach for predicting stock prices to a deep learning based approach.

By the end of this project, will draw conclusions on whether the Deep neural network based approach for predicting stock prices can take over the Classic regression based approach. More specifically, we will use Long Short-Term Memory (LSTM) Networks on our time series data to find patterns behind the vague movement of prices of stocks and try to predict future values of the same.

Keywords- *Neural Networks, Recurrent Neural Networks, Long Short Term Memory Networks*

I. INTRODUCTION

After having a look at the growing inflation rate, saving money does not seem to be the best option. Inflation rate in the year 2022 is projected to be 5.7 %. This means that, If we save money in 2022, If value is decreasing at a rate of 5.7 % a year. The stock market is the perfect solution to avoid the effects of inflation for a common man. Here, People can trade in dematerialized Assets and generate side income. Especially after the emergence of many applications that simplify the process of creating Demat accounts and trading in stocks, the only question people are left with is, which stock do we buy? The prediction made by our project will help its users to make an intuition on whether to keep a particular company stock or to sell it. The main objective of this project is to draw a conclusion on which technique is best suited for predicting stock prices among the regression and deep neural networks (LSTM). These two models will be compared based on various metrics like; accuracy, recall score, confusion matrix etc. Our primary goal will be to reach the maximum accuracy without overfitting the model. Secondly, The end result must be more usable by naive users, Minimum input must be taken from the user and a high degree of abstraction must be maintained so as not to overwhelm the user.

II. LITERATURE REVIEW

3.1 Using Neural Networks to Forecast Stock Market Prices,

Author: Ramon Lawrence.

This paper is a survey on the application of neural networks in forecasting stock market prices. With their ability to discover patterns in nonlinear and chaotic systems, neural networks offer the ability to predict market directions more accurately than current techniques. Common market analysis techniques such as technical analysis, fundamental analysis, and regression are discussed and compared with neural network performance. Also, the Efficient Market Hypothesis (EMH) is presented and contrasted with chaos theory and neural networks. Finally, future directions for applying neural networks to the financial markets are discussed.

3.2 Stock Market Prediction Using Hybrid Approach,

Authors: Vivek Rajput, Sarika Bobde.

The objective of this paper is to construct a model to predict stock value movement using the opinion mining and clustering method to predict National Stock Exchange (NSE). It used a domain specific approach to predict the stocks from each domain and took some stock with maximum capitalization. Topics and related opinions of shareholders are automatically extracted from the writings in a message board by utilizing our proposed strategy alongside isolating clusters of comparable sort of stocks from others using clustering algorithms. Proposed methodology will give two output sets i.e. one from sentiment analysis and another from clustering based prediction with respect to some specialized parameters of stock exchange. By examining both the results an efficient prediction is produced. In this paper stocks with maximum capitalization within all the important sectors are taken into consideration for empirical analysis.

3.3 Hybrid ARIMA-BPNN Model for Time Series Prediction of the Chinese Stock Market,

Authors: Li Xiong, Yue Lu.

Stock price prediction is a challenging task owing to the complexity patterns behind time series. Autoregressive integrated moving average (ARIMA) model and back propagation neural network (BPNN) model are popular linear and nonlinear models for time series forecasting respectively. The integration of two models can effectively capture the linear and nonlinear patterns hidden in a time series and improve forecast accuracy. In this paper, a new hybrid ARIMA-BPNN model containing technical indicators is proposed to forecast four individual stocks consisting of both main board market and growth enterprise market in software and information services sector

3.4 Stock index forecasting based on a hybrid model,

Authors: J.J. Wang, J. Z. Wang, Z. G. Zhang, and S. P Guo.

This paper examines the prediction performance of ARIMA and artificial neural networks model with obtained stock information from New York Stock Exchange. The empirical results obtained reveal the prevalence of neural networks model over ARIMA model. The findings further resolve and clarify contradictory opinions reported in literature over the prevalence of neural networks and ARIMA model and the other way around

3.5 A Hybrid Fuzzy Time Series Model Based on ANFIS and Integrated Nonlinear Feature Selection Method for Forecasting Stock

Authors: Chung-Ho Su, Ching-Hsue Cheng.

Forecasting stock price is a hot issue for stock investors, dealers and brokers. However, it's difficult to find out the best time point to buy or to sell stock, due to many variables that will affect the stock market, and stock dataset is time series data. Therefore, many time series models have been proposed for forecasting stock price, furthermore the previous time series methods still have some problems. Hence, this paper proposes a novel ANFIS (Adaptive Neuro Fuzzy Inference System) time series model based on integrated nonlinear feature selection (INFS) method for stock forecasting.

III. PROBLEM STATEMENT

The problem with trading in stock is that the trader needs to have a certain level of expertise to trade fluently and generate profits. Not every common man is an expert trader by birth , And not everyone has the luxury to take time out to study about the stock market. We need a solution that can help a user in trading stocks without him being an expert in it And help him make intuitions about The steps he might take.

IV. PROPOSED SYSTEM

This paper supports the idea of using deep neural networks based approaches for predicting stock prices. We focus on training a long short term memory network using time series data of previous stock prices.

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn! The LSTM has the ability to remove or add information to the cell state, carefully regulated by structures called gates. Gates are a way to optionally let information through.

Using this property of LSTM networks, the model can optionally ignore less relevant information and Derive robust patterns that can help it predict the future values.

Advantages:

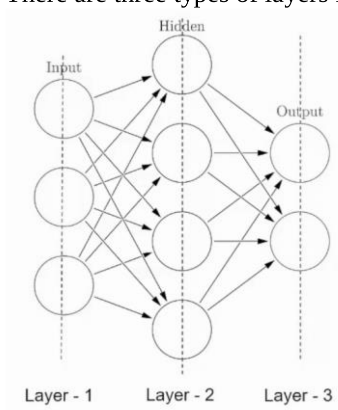
Model can optionally ignore less relevant information

A Public interface will ease the access for users.

V. SYSTEM STUDY

NEURAL NETWORK:

A neural network is made up of vertically stacked components called Layers. Each dotted line in the image represents a layer. There are three types of layers in a NN.



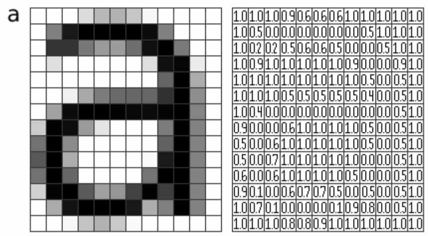
Input Layer– First is the input layer. This layer will accept the data and pass it to the rest of the network.

Hidden Layer– The second type of layer is called the hidden layer. Hidden layers are either one or more in number for a neural network. In the above case, the number is 1. Hidden layers are the ones that are actually responsible for the excellent performance and complexity of neural networks. They perform multiple functions at the same time such as data transformation, automatic feature creation, etc.

Output layer– The last type of layer is the output layer. The output layer holds the result or the output of the problem. Raw images get passed to the input layer and we receive output in the output layer.

CONVOLUTIONAL NEURAL NETWORKS:

A Convolutional Neural Network, also known as CNN or ConvNet, is a class of neural networks that specializes in processing data that has a grid-like topology, such as an image. A digital image is a binary representation of visual data. It contains a series of pixels arranged in a grid-like fashion that contains pixel values to denote how bright and what color each pixel should be.



This layer performs a dot product between two matrices, where one matrix is the set of learnable parameters otherwise known as a kernel, and the other matrix is the restricted portion of the receptive field. The kernel is spatially smaller than an image but is more in-depth. This means that, if the image is composed of three (RGB) channels, the kernel height and width will be spatially small, but the depth extends up to all three channels.

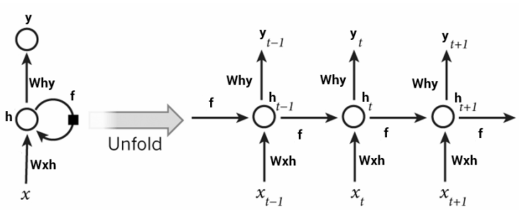
$$W_{out} = \frac{W - F + 2P}{S} + 1$$

Formula for Convolutional layer

RECURRENT NEURAL NETWORKS:

In the conventional feed-forward neural networks, all test cases are considered to be independent. That is when fitting the model for a particular day, there is no consideration for the stock prices on the previous days.

This dependency on time is achieved via Recurrent Neural Networks. A typical RNN looks like:



Here every prediction at time t (h_t) is dependent on all previous predictions and the information learned from them.

Limitations of RNNs:

Recurrent Neural Networks work just fine when we are dealing with short-term dependencies.

The reason behind this is the problem of Vanishing Gradient. We know that for a conventional feed-forward neural network, the weight updating that is applied on a particular layer is a multiple of the learning rate, the error term from the previous layer and the input to that layer. Thus, the error term for a particular layer is somewhere a product of all previous layers' errors. When dealing with activation functions like the sigmoid function, the small values of its derivatives (occurring in the error function) get multiplied multiple times as we move towards the starting layers. As a result of this, the gradient almost vanishes as we move towards the starting layers, and it becomes difficult to train these layers.

Turns out that an RNN transforms the existing information completely by applying a function. Because of this, the entire information is modified, on the whole, i. e. there is no consideration for 'important' information and 'not so important' information.

LSTM (LONG SHORT-TERM MEMORY) NETWORKS:

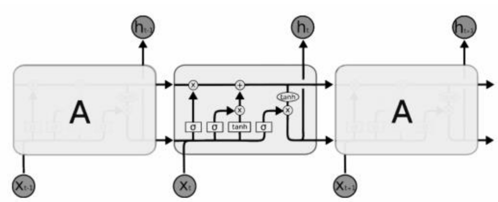
LSTMs on the other hand, make small modifications to the information by multiplications and additions. With LSTMs, the information flows through a mechanism known as cell states. This way, LSTMs can selectively remember or forget things. The information at a particular cell state has three different dependencies.

These dependencies can be generalized to any problem as:

- The previous cell state (i.e. the information that was present in the memory after the previous time step)
- The previous hidden state (i.e. this is the same as the output of the previous cell)
- The input at the current time step (i.e. the new information that is being fed in at that moment)

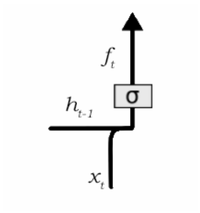
A typical LSTM network is comprised of different memory blocks called cells

(the rectangles that we see in the image). There are two states that are being transferred to the next cell; the cell state and the hidden state. The memory blocks are responsible for remembering things and manipulations to this memory are done through three major mechanisms, called gates. Each of them is being discussed below.



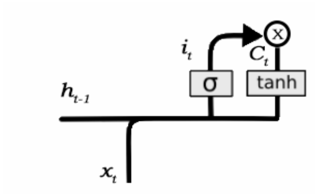
Source

A **forget gate** is responsible for removing information from the cell state. The information that is no longer required for the LSTM to understand things or the information that is of less importance is removed via multiplication of a filter. This is required for optimizing the performance of the LSTM network. This gate takes in two inputs; h_t-1 and x_t.



Basically, the sigmoid function is responsible for deciding which values to keep and which to discard. If a '0' is output for a particular value in the cell state, it means that the forget gate wants the cell state to forget that piece of information completely. Similarly, a '1' means that the forget gate wants to remember that entire piece of information. This vector output from the sigmoid function is multiplied to the cell state.

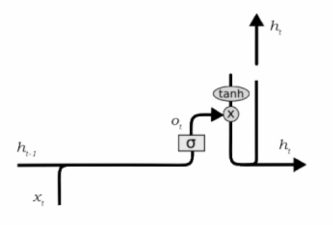
The **input gate** is responsible for the addition of information to the cell state.



This addition of information is basically a three-step process as seen from the diagram above.

1. Regulating what values need to be added to the cell state by involving a sigmoid function. This is basically very similar to the forget gate and acts as a filter for all the information from h_t-1 and x_t.
2. Creating a vector containing all possible values that can be added (as perceived from h_t-1 and x_t) to the cell state. This is done using the tanh function, which outputs values from -1 to +1.
3. Multiplying the value of the regulatory filter (the sigmoid gate) to the created vector (the tanh function) and then adding this useful information to the cell state via addition operation.

This job of selecting useful information from the current cell state and showing it out as an output is done via the **output gate.** Here is its structure:



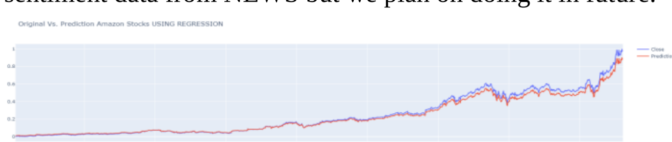
The functioning of an output gate can again be broken down to three steps:

1. Creating a vector after applying tanh function to the cell state, thereby scaling the values to the range -1 to +1.
2. Making a filter using the values of h_t-1 and x_t, such that it can regulate the values that need to be output from the vector created above. This filter again employs a sigmoid function.
3. Multiplying the value of this regulatory filter to the vector created in step 1, and sending it out as an output and also to the hidden state of the next cell.

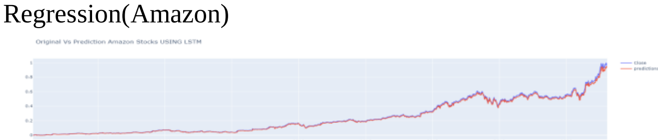
VI. CONCLUSION

This paper explores the possibility of using LSTM Neural Networks for time series data analysis. Time series problems are mostly viewed as Machine learning problems. ARIMA, SARIMAX, FBprophet uses Convolutional Neural Networks for the same. Though they are quite efficient, they still suffer from the Vanishing Gradient problem. In this paper, we perform the time series analysis using Ridge Regression Technique and LSTM Neural Networks and Compare the Performance. We can see a significant improvement in the accuracy of predictions when using LSTM as we consider the previous information while using it.

We can further improve the performance by considering sentiment data from NEWS but we plan on doing it in future.



Output Fig 1: Prediction of closing stock prices using Ridge Regression(Amazon)



Output Fig 2: Prediction of closing stock prices using LSTM Networks(Amazon)

VII. REFERENCES

[1] Stock Price Prediction Using LSTM on Indian Share Market by Achyut Ghosh, Soumik Bose1, GiridharMaji, Narayan C. Debnath, Soumya Sen

[2] S. Selvin, R. Vinayakumar, E. A. Gopalkrishnan, V. K. Menon and K. P. Soman - Stock price prediction using LSTM, RNN and CNN-sliding window model - 2017.

[3] Murtaza Roondiwala, Harshal Patel, Shraddha Varma, “Predicting Stock Prices Using LSTM” in Undergraduate Engineering Students, Department of Information Technology, Mumbai University, 2015.

[4] Xiongwen Pang, Yanqiang Zhou, Pan Wang, Weiwei Lin, “An innovative neural network approach for stock market prediction”, 2018

[5] Ishita Parmar, Navanshu Agarwal, Sheirsh Saxena, Ridam Arora, Shikhin Gupta, Himanshu Dhiman, Lokesh Chouhan Department of Computer Science and Engineering National Institute of Technology, Hamirpur – 177005, INDIA - Stock Market Prediction Using Machine Learning.

[6] Pranav Bhat Electronics and Telecommunication Department, Maharashtra Institute of Technology, Pune. Savitribai Phule Pune University - A Machine Learning Model for Stock Market Prediction.

[7] Anurag Sinha Department of computer science, Student, Amity University Jharkhand Ranchi, Jharkhand (India), 834001 - Stock Market Prediction Using Machine Learning.

[8] V Kranthi Sai Reddy Student, ECM, Sreenidhi Institute of Science and Technology, Hyderabad, India - Stock Market Prediction Using Machine Learning.

[9] Asset Durmagambetov currently works at the mathematics, CNTFI- 'The Riemann HypothesisMillennium Prize Problems' - stock market predictions.

[10] Mariam Moukalled Wassim El-Hajj Mohamad Jaber Computer Science Department American University of Beirut - Automated Stock Price Prediction Using Machine Learning.