Stock Market Data Analysis and Stock Price Prediction

A Comparative Analysis using Neural Networks

Submitted To:

Dr. Somesh K. Mathur
Professor, IIT Kanpur
skmathur@iitk.ac.in

Submitted By:

Abhishek Yadav(190043)(J3)
Deepalok Kaushik(190261)(J3)
Neeraj Pratap Singh(190542)(J2)
Nitin Patil(190572)(J3)
Group(19) Leader-Prajwal Arya(190613)(J4)
Rishabh Arora(190698)(J1)
Rishabh Chaudhary(190700)(J2)
Rohan Kumar(190715)(J2)
Rohit Sharma(190719)(J1)
Sandipan Mitra(190755)(J4)

Abstract:

As a company's stock price is an indicator of its value in public, it is quite interesting to analyze and predict the stock market. There are many factors outside and within the company that affect the stocks' prices, so it is one of the most unpredictable trends. Researchers use many linear and non-linear models to predict the stock market. Though the development of a precise prediction tool is still challenging, much effort has been made in this field using advanced techniques. The most popular method is by using intelligent deep learning architectures.

This paper will present an improved model based on Artificial Neural Networks to successfully predict the stock price and compare it with the actual price. For this purpose, we are using multi-dimensional data of five companies listed on the National Stock Exchange (NSE) of India from various sectors for the last six months. To train the network faster, we reduce the multi-dimensionality of the data by using **Principal Component Analysis (PCA).**

Introduction:

We know that stock market prediction is one of the toughest tasks for everyone. The basic of stock market prediction is to predict the future stock using the market statistics of the past few years. However, not much technique or combination of techniques has been successful enough. In our research work, we will use the **Artificial Neural Network (ANN)**, as it is the most powerful tool to predict and analyze data. Our biological brain is the main concept for neural networks. It is perfect for finding out the unknown relation among different data variables and recognizing complex patterns.

Literature Review:

Stock prices are very sensitive to macroeconomic variables. Public news announcements or moments of no public information release also considerably affect fluctuations in stock prices. During observing patterns, several studies have revealed that stock returns exhibit reversal and drift over different intervals. Several models are planned to capture the predictability of stock returns. Popular models are generally too restrictive, and they fail to perform empirical tests. Most of the studies that use ANN to predict movements in stock prices use high-frequency data, meaning hourly or daily data. Martinez et al. analyze intraday price movements and predict

the best times to trade and make profits in a day. Some other studies try to predict the direction of the movements rather than how much the stock moves, where the output is usually a categorical variable.

Objective:

Our objective is to propose an approach towards stock prediction using the neural network technique, test our model on various companies to predict their stock prices, and analyze and compare the result observed with the stocks' actual prices.
ML models can be used to make predictions. We can also use such models to make stock market predictions. The ML model that we found suitable for this purpose was the Artificial Neural Network (ANN), which exploits parallel computing to gain intelligence from input data as a basis for predicting future values.

Hypothesis:

The **Random Walk Hypothesis** is a hypothesis in financial economics which states that "we cannot predict stock market prices since it evolves according to a random walk." According to this theory, without assuming additional risk, it is impossible to outperform the market. On the other hand, the **Efficient-Market Hypothesis** (**EMH**) is a financial theory stating that "asset prices reflect all available information about the stock."
In our paper, we formally took the null hypothesis as, "the past asset price of the stock does not affect the present asset price." And as the alternate hypothesis, we took that "the past asset price of the stock affects the present asset price."
Therefore:
Null Hypothesis: The past asset price of the stock does not affect the current asset price.
Alternate Hypothesis: The past asset price of the stock affects the current asset price.

Model and Methodology:

1. Data extraction from NSE

We have extracted the data from the Centre for Monitoring Indian economies (CMIE) prowess dx, which is the largest database of the financial performance of Indian companies. We have extracted the data of five companies from the past six months,

which includes variables like *opening price, closing price, high price, trade quantity, average rate, share delivery, etc.*

co_code	company_name	co_stkdate	nse_opening_price	nse_high_price	nse_low_price	nse_closing_price	nse_traded_qty	nse_avg_rate	nse_delivery_qty
256066	AXIS BANK LTD.	01-07-2020	410	437.35	406.65	433.25	53380997	423.3	11277321
256066	AXIS BANK LTD.	02-07-2020	441	441.5	421.3	423.2	50362840	427.8	8019429
256066	AXIS BANK LTD.	03-07-2020	430	431.6	423.65	428.45	31044470	427.75	5973108
256066	AXIS BANK LTD.	06-07-2020	434.15	437.6	431.1	434	18425359	434.51	3120874
256066	AXIS BANK LTD.	07-07-2020	433.5	449	426.2	447.35	37991906	438.58	6686878
256066	AXIS BANK LTD.	08-07-2020	450	461.4	441	444.15	45878605	450.01	6452488
256066	AXIS BANK LTD.	09-07-2020	447.85	457	446.05	453.75	29162682	452.36	6984633
256066	AXIS BANK LTD.	10-07-2020	449.9	451.45	437	439.6	25676461	443.12	5559514
256066	AXIS BANK LTD.	13-07-2020	443.95	446.4	432.75	440.05	21326996	439.02	3355967
256066	AXIS BANK LTD.	14-07-2020	434.9	435.8	415.4	417.7	25686686	422.06	6357440

2. Principal Component Analysis

Principal Component Analysis (PCA) is a dimensionality reduction method that is used to minimize the dimension of large datasets. This is done by transforming several variables into a smaller one which yet contains a lot of facts in the large set.

We have several variables in our dataset. For example, we take the data set variables as opening price (OP), closing price (CP), high price (HP), and low price (LP). We reduce them into two principal components, PC_1 and PC_2. This data will be used as an input for neural networks.

3. Model selection

ANN(Artificial Neural Network)

The Artificial Neural Network is capable of learning any nonlinear function. Therefore, these networks are also known as Universal Function Approximators. The main reason for Universal Function approximation is the Activation function; it introduces nonlinear properties to the network. This helps the network learn any complex relationship between input and output.ANN can be used to solve problems related to **Tabular data, Image data, Text data.**

RNN(Recurrent Neural Network)

RNN is meant to process sequences of data. They are used in forecast models and language models. The most common kind of recurrent layers is called LSTM (Long Short Term Memory) and GRU (Gated Recurrent Units). RNNs share the parameters across different time steps. This is popularly known as Parameter Sharing. The RNN is used for **Time Series data, Text data, Audio data, and NLP(Natural language processing).**

CNN(Convolutional Neural Network)

CNN is explicitly designed for **computer vision**. The name comes from convolutional layers: they are different from standard layers of canonical ANNs, and they have been invented to receive and process pixel data. They are used for **image processing etc**.

	ANN	RNN	CNN
Data	Tabular Data	Sequence Data	Image Data
Recurrent Connections	No	Yes	No
Parameter Sharing	No	Yes	Yes
Spatial Relationship	No	No	Yes
Using Complexity	Easy	Medium	Hard

Therefore we chose ANN, which is easier to implement and can be used to approximate the result of our problem with minimum error.

Activation Function

The activation function used should be nonlinear because they allow backpropagation because they have a derivative function is related to the inputs.We will use **ReLU** or **Leaky ReLU** as our activation function as it is less susceptible to vanishing gradient problem i.e *the gradients become very less for large values* which is convenient for our problem because our data set will have a wide range of values.

4. Training of model

After applying PCA, we get reduced data, and this is used as the input vector for the Artificial neural network(ANN). The data available to us is divided into two parts: the first part is for training and the second is for testing.

The model consists of neurons in the input layer, hidden layer, and output layer. We specify these parameters according to our data set. There are some weights and biases associated with the networks between the nodes; these values are changed to fit our model with the data.

In the beginning, we assign random values to the hyperparameters like weights and biases for the network. In the feed-forward step, input patterns are propagated through the network one by one, and actual outputs are calculated. Comparing actual and target outputs, according to the BPNN algorithm, the magnitude of error is determined. Weight updates take place through the backpropagation to minimize the error in the cost function. When all the data is passed through the model and

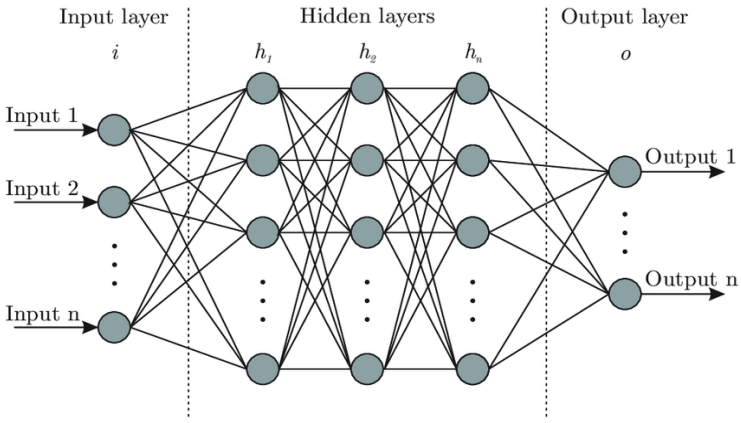
hyperparameters are set accordingly, this is called an epoch. We perform multiple epochs to maximize our model's efficiency, but not so much to cause overfitting.

$$\Rightarrow Updated\,Weight = weight(old) + [(learning\,rate) * (output\,error) * output(neuro$$
$$output(neurons\,i + 1) * (1 - Output(neurons\,i + 1))]$$

Hidden layer neurons generate the final output, which is compared with the actual output data, and calculate an error signal. Unless estimation reaches a satisfactory level measured based on the error threshold, as specified before epochs, it is continued. In this work, five different companies are trained individually, and their results are analyzed.

5. Testing

The trained model is then tested with the remaining data. The resulting data is then plotted against the actual data. Error is calculated. We try to minimize this error by choosing the number of epochs etc. We can also divide the whole data into three parts and use one part to validate our model parameters, i.e., to prevent overfitting. We will set the number of hidden layers, the number of neurons in each layer, the number of epochs, and the learning rate(hyperparameters) by the hit and trial method. We will do error analysis for different values and then adjust these values to minimize the error and to avoid overfitting and underfitting of our model. Our range for the number of neurons will be between the size of the input and the output.



Tentative Conclusions:

In our suggested system, we will develop a model to anticipate a particular firm's stock rate by training their preceding data in a neural network. First, we use Principal Component Analysis (PCA) to reduce the data dimension, which helps train our system faster. We can get the most influential features of data after the reduction of the data dimension. Backpropagation neural network, which is one of the finest neural network methods, will be used, which can reduce the error between the actual output and the desired output by using gradient descent. Since it will be quite

strenuous to predict with 100% accuracy, the performance is not expected to be satisfactory every time. This encourages the use of better models, different specifications, and different processing of the data to predict more accurately future stock movements.

References:

[I] Indian stock market prediction using artificial neural networks on tick data -*.By Dharmaraja Selvamuthu, Vineet Kumar & Abhishek Mishra(2019)*

[II] Principal Component Analysis, *By Ian T. Jolliffe*

[III] (Stock Market Data Analysis and Future Stock Prediction using Neural Network - *By Tapashi Gosswami, S.K. Saha and Mahmudul Hasan (*Base paper) https://www.researchgate.net/publication/328418664_Stock_Market_Data_Analysis_and_Future_Stock_Prediction_using_Neural_Network

[IV] NSE Stock Market Prediction Using Deep-Learning Models - *By Hiransha Ma, Gopalakrishnan E.Ab, Vijay Krishna Menonab, Soman K.P,* Volume 132, 2018, Pages 1351-1362

[V] Optimal Neural Network Architecture for Stock Market Forecasting - By Gitansh Khirbat, Rahul Gupta and Sanjay Singh (CSNT) *(2013)*

[VI] Martinez, L. C., da Hora, D. N., Palotti, J. R. d. M., Meira, W., and Pappa, G. L. (2009). From an artificial neural network to a stock market day-trading system: A case study on the bm&f bovespa. In Neural Networks, 2009. IJCNN 2009. International Joint Conference on, pages 2006–2013. IEEE.