# **STOCK MARKET ANALYSIS USING STACKED LONG SHORT-TERM MEMORY(S-LSTM)**

**RUPESH KOUSHIK – 200953018**

**JOSH JAISON -200953033**

**RAJIV BASKARAN – 200953027**

**SUHAS GOWDA - 200953014**

**INDEX PAGE**

|  |  |
| --- | --- |
| Front page | **0** |
| Index | **1** |
| Abstract | **2** |
| Introduction | **3-4** |
| Background Information | **4** |
| Methodology(design) | **5-9** |
| Implementation | **9-11** |
| Results | **11-14** |
| Conclusion | **14** |
| References | **14-15** |

# **Abstract**

# One of the most intricate machine learning problems is the share value prediction. Stock market prediction is an activity in which investors need fast and accurate information to make effective decisions. Moreover, the behavior of stock prices is uncertain and hard to predict. For these reasons, stock price prediction is an important process and a challenging one. This leads to the research of finding the most effective prediction model that generates the most accurate prediction with the lowest error percentage. The goal of the project is to build a website for trading in the Stock Market using Stacked Long short-term memory. Using keras library we train the model to predict the cost of a stock or index an implementation of various arbitration strategy to profit from these changes in prices.

# **Introduction**

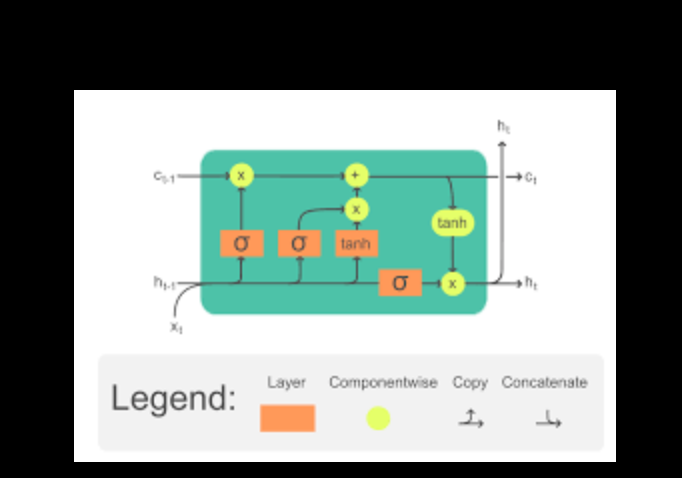
The stock market is known for its volatility, randomness, and unpredictability. It is a chaotic place with an unbelievably huge continuously changing stream of data which makes predicting and acting on those predictions to make a profit very hard. It is actually one of the most challenging tasks in times series forecasting. This research’s main goal is to study and apply deep learning techniques to the stock market in order to predict stock behaviour and thus act on those predictions to avoid investment risk and generate profit. The goal is to be achieved by using transfer learning in order to take advantage of pre-built neural networks models. Predictions are then tested against actual historical stock price data.

In this project, we used the Stacked Long Short-Term Memory (LSTM) approach to forecast future stock values using data from the stock market. We trained the model to forecast the price of a stock or index using the TensorFlow library, and we implemented multiple arbitration strategies to profit from these price movements. Using the different python libraries like pandas, NumPy, Matplotlib, Seaborn as well as Jupyter Notebooks we could obtain stock information, visualize various elements of it, and analyses a stock's risk based on its prior performance history. We have also looked at the stock's volatility rather than its absolute value when analyzing its risk. By using the libraries like datareader , yfinance we obtain the datasets for our stock market analysis. We analyze the stock for companies like Google, Amazon, Microsoft, Apple.

**Background Information**

We have used yahoo-finance api in this project where it offers an excellent range of market data on stocks, bonds, currencies and cryptocurrencies. It also offers market news, reports and analysis and additionally options and fundamentals data

We used streamlit app framework which is an open source app framework in Python language. It **helps us create web apps for data science and machine learning in a short time**. It is compatible with major Python libraries such as scikit-learn, Keras, PyTorch, SymPy(latex), NumPy, pandas, Matplotlib etc.



**Methodology(design)**

A. Long Short Term Memory model (LSTM)

LSTM, which stands for Long Short Term Memory, is a type of neural network which is particularly useful in the case of time series forecasting. According to an article by Srivastava on LSTM’s and essentials of deep learning, an LSTM network is the most effective solution to time series analysis and thus stock market prediction. With the recent breakthroughs that have been happening in data science, it is found that for almost all of these sequence prediction problems, long short Term Memory networks have been observed as the most effective solution. LSTMs have an edge over conventional feed-forward neural networks and Recurrent Neural Networks in many ways. This is because of their property of selectively remembering patterns for long durations of time.

In the case of a basic neural network, in order to add a new information, it transforms the existing information completely by applying a sigmoid function. Because of this, the entire information is modified as a whole. Thus, there is no consideration for ‘important’ information and ‘not so important’ information. 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.

We have looked at the stock's volatility rather than its absolute value when analysing its risk.

We utilised seaborn to generate a histogram and a KDE plot on the same graphic.

We can now see that there should be a linear relationship between the daily return values of two equities if they have a perfect correlation with one another.

Repeating this comparative study for every conceivable combination of stocks in our technology stock ticker list is made incredibly simple by Seaborn and Pandas. To generate this plot automatically, use sns.pair plot().

We may also do a correlation plot to gain precise numerical figures for the correlation between the stocks' daily return values. Analysing the closing prices, we can notice an intriguing link between Microsoft and Apple.

We can see here, both quantitatively and visually, that Microsoft and Amazon had the highest connection of daily stock return, as we had expected from our PairPlot. It is also intriguing to note the strong correlation between all technological companies.

There are several methods for quantifying risk,

Comparing the expected return with the standard deviation of the daily returns is one of the simplest ways to use the data we have acquired on daily percentage returns.

**Implementation:**

Proposed Algorithm is given below:

Step 1: Importing the Libraries

Step 2: Getting to Visualising the Stock Market Prediction Data

Step 3: Check for Null Values by printing the DataFrame Shape

Step 4: Setting the Target Variable and Selecting the Features

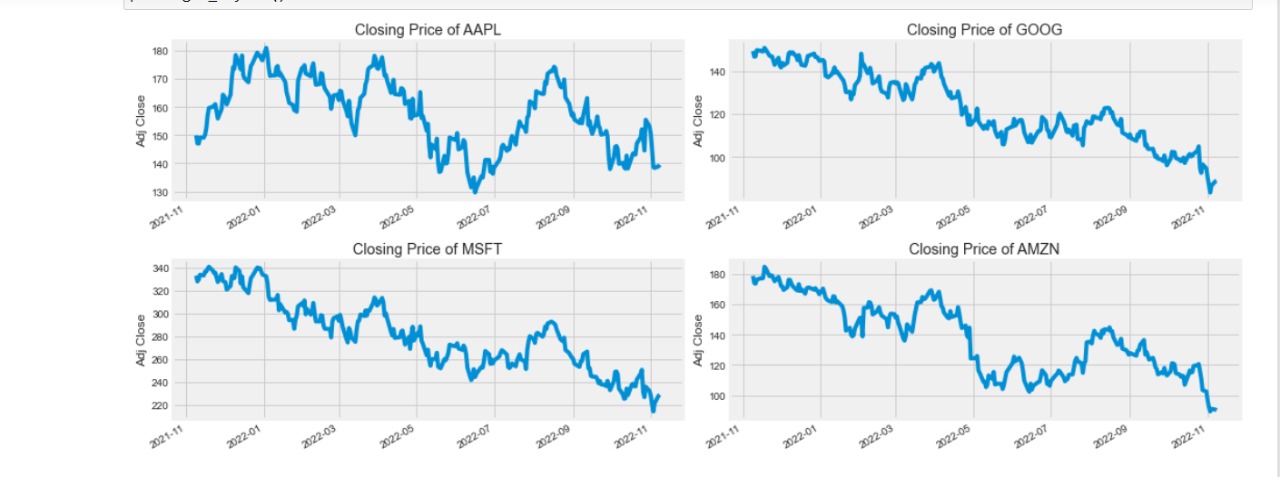
Step 5: Creating a Training Set and a Test Set for Stock Market Prediction

Step 6: Building the LSTM Model for Stock Market Prediction

Step 7: Training the Stock Market Prediction Model

Step 8: LSTM Prediction

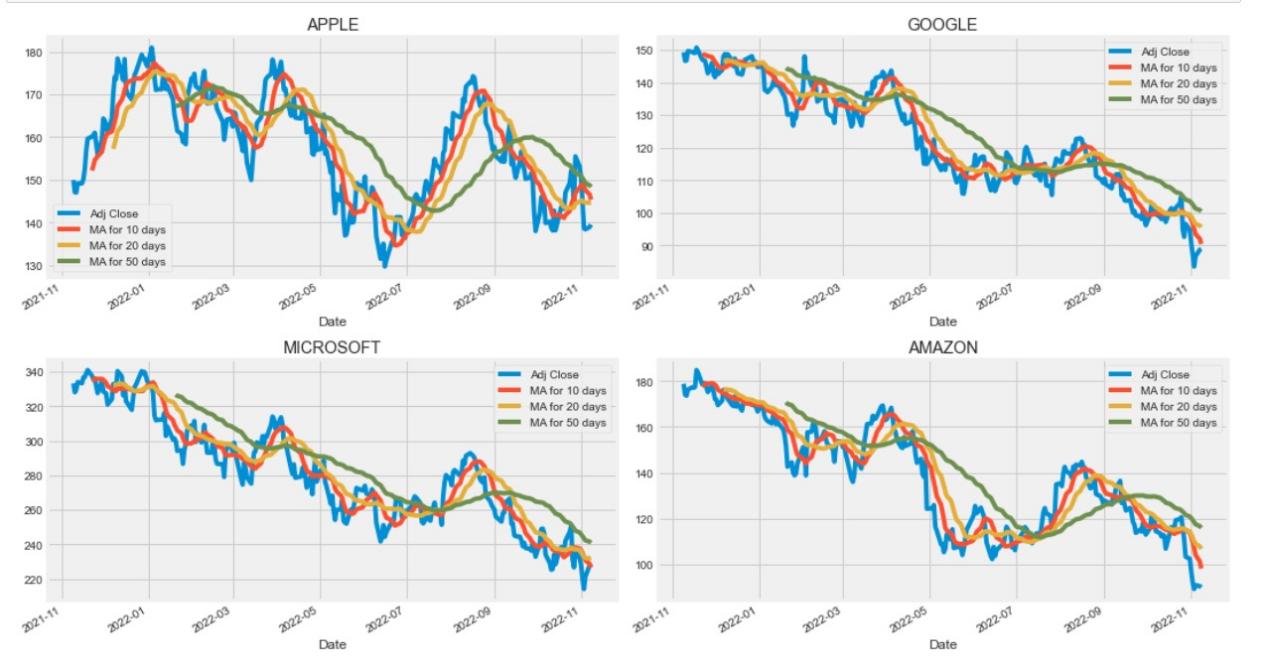
Step 9: Comparing Predicted vs True Adjusted Close Value – LSTM

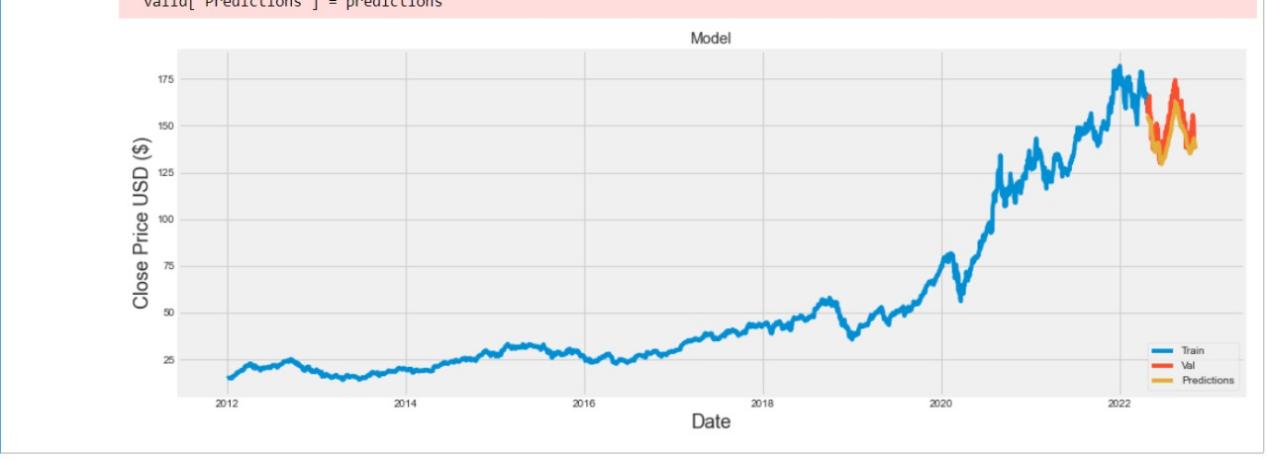


**Results**

This research aimed to test LSTM networks in forecasting the value of the apple and tsla

on the data from 2013 to the end of 2020 on data with the following frequencies: daily, 1 h and 15 min. Based on the forecasts from LSTM models we generated buy and sell investment signals, used them in algorithmic investment strategies and created equity lines for our investment. For this purpose we used various combination of LSTM models optimized on in-sample period and tested on out-of-sample period with rolling window approach. We paid special attention to data preprocessing in the input layer, to avoiding overfitting in the estimation and optimization process, and we assured correct selection of hyperparameters at the beginning of our tests. We introduced our authorship loss function with better utilizes the forecasting ability of LSTM model in algorithmic investment strategies. Then we performed the sensitivity analysis of the main parameters and hyperparameters. In the final step, we combined the signals from various frequencies into one ensemble model





**Conclusion**

The popularity of stock market trading is growing rapidly,which is encouraging researchers to find out new methods for the prediction using new techniques. The forecasting technique is not only helping the researchers but it also helps investors and any person dealing with the stock market. In order to help predict the stock indices, a forecasting model with good accuracy is required.

**References**

* <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00512-z>
* <https://www.youtube.com/>
* https://medium.com/