[[1]](#footnote-1)

Stock Prediction using neural networks

Sneha, Mohdi Habibi, Shreya Hagalahalli Shrinivas, Mahesh Reddy Konatham

Computer Engineering Department

San Jose State University

***Abstract*— The Stock market has been running on volatility and extreme fluctuation due to influences from many external factors. The sentiment of the market, the health of each industry along with the overall economy are some of the big factors that can help predict some of the fluctuations. Although there have been many attempts at stock price predictions in the past, the time series stock market data combined with the aforementioned factors can give a good indication of problems that can be solved with modern machine learning techniques. Traditional applications of deep learning include speech recognition and image classification within which neural networks have proved their capability in learning to decode non-linear mappings between inputs and outputs. Utilizing such modern techniques to create a model that predicts the future of the stock market can result in increased customer confidence in the system, indicating increased stock purchases over time. In this project, we propose to forecast future movements in the stocks by leveraging Neural networks and Deep learning techniques. We try to present a solution that can overcome the naive estimator effect and thus precisely estimating the next gradient change, which is one of the biggest challenges in predicting the stocks.**

***Index Terms*— Stock Prediction, Neural Networks, Deep learning, RNN, CNN, LSTM, RMSE.**

# INTRODUCTION

Stock market historically has been very volatile with many factors having direct impact on the fluctuating nature of it. Many stock trading techniques involve technical analysis and fundamental analysis of each individual company’s stock to understand the future of its value. Not only does a company's health and earnings have weight in deriving its stock value, but also many other external factors such as sentiment of the market and public reaction toward events have direct effect on the volatility.

With the vast amount of data available from the stock market, there has been lots of interest in using Machine Learning models to solve such problems. The idea is to extract knowledge from the previous stock market data to train such models so that they can predict the future price of a certain stock. There are many shortcomings in terms of using Machine Learning models to predict stock such as not taking into consideration the full previous history of a certain stock when prediction is forming. The history context in solving such problems is a crucial component that many Machine Learning models lack in.

Recently, Artificial Intelligence has shown promising results in Image Recognition and Natural Language Processing fields. The new branch of AI which is Deep Neural Networks has been the underlying process for getting better results combining with the powerful processing units out in the market. Recurrent Neural Networks (RNN) are a branch of Deep Learning which can bring the history context into picture. That is why it has been a popular tool to be applied to problems that need previous inputs for better prediction. The vanilla RNN models have a problem with keeping long term memory by design. Long Short-Term Memory is a type of Recurrent Neural Networks that is specialized in carrying over information for a long time to allow neural networks rely on historical context and not only the previous inputs.

This research paper focuses on the previous movement of the stock market to predict the near future volatility without taking into consideration other external factors. The idea behind this approach is how to derive certain knowledge from stock market time series data using deep learning to understand near future positive or negative volatility. The automation of such a process will allow anyone to make quick decisions for making profit from the stock market volatility. Since LSTMs are specialized in keeping the historical context of inputs, it makes it a viable option for stock market prediction.

# Related Work

## Existing Work

Stock market prediction involves dealing with complex, deficient and highly skewed data as stated by Kim et al. [1], and because of the non-linearity problem, it becomes very hard to forecast. Tillakaratne et al. [2] also mentions the imbalance of trading signals in the classification of data in to buy, sell and hold classes. Yu et al. [3] explored the conventional and standard machine learning models for predicting stocks like using genetic algorithms for feature extraction. The features extracted from the least squares support vector machine (LSSVM) learning model with optimal parameters and kernel techniques are used to predict the stocks. Zahedi et al. [4], used principal component analysis (PCA) for feature extraction, dimensionality reduction and with the help of artificial neural networks (ANNs) and various other accounting variables to predict new patterns in stock movements. Saad et al. [5] explained the problem of false alarm arising from a lack of short-term memory. The authors [5] also explored TDNN, RNN, and PNN for prediction to minimize risks and losses. Kwon et al. [6] stated that the hybrid algorithm based on a neurogenetic model, in which the data was generated by technical indicators, outperformed the average buy and hold strategy. Yu et al. in the most recent research [7], stated that integrating deep learning and neural networks helped tremendously in solving the nonlinear complications and thus justifying their results in achieving better accuracy for stock prediction.

The comparison of artificial neural networks with heuristic genetic algorithms for feature extraction is very useful, especially in reducing the dimensional volume and unrelated components for forecasting the stock movement [1]. The problem of an imbalanced data set is addressed in this research paper. The parameters in the neural network are adjusted according to the forward movement pattern and the optimal least square error model [2]. This research paper thoroughly analyzed the usage of the support vector machine model with the technique of minimizing the least square distance of each data point from the imaginary hyperplane sides [3]. Exploiting the Artificial neural networks with the help of principal component analysis seems to be very interesting. The optimal solution for finding the correct number of principal components is calculated by experiments on new data patterns [4]. The idea of minimizing the risk ratio by reducing the amount of loss using algorithms like time delay, recurrent neural networks is very interesting. Forecasting which is built on the analysis of daily closing prices of stocks is a good strategy for maximizing profits [5]. This research paper proposes that we can generate an input feature vector by using technical measures from business analysts. After that this feature vector can be optimized by using a Genetic algorithm that can be further used to reduce the neural network weights using two-dimensional encoding [6]. This research paper explores the concept of long short-term memory (LSTM) and deep learning to achieve better accuracy than traditional machine learning models. The problem of non-linearity and the solution to tackle that is discussed which can help my research on stock prediction [7].

## Baseline Approaches

The baseline chosen for this research paper is an implementation of stock prediction using LSTM [8]. This research paper explores the concept of long short-term memory (LSTM) and deep learning to achieve better accuracy than traditional machine learning models. In this paper the implementation is discussed as such, the approach taken is to predict the stock movement using LSTM with a limited number of dimensions for data. This research paper will focus on the same implementation but with more dimensions to bring a holistic view of the data into account such as volume and total trades. These two factors can result in decisive improvement toward the results since the volume and total trades within a certain amount of time can be an indication of positivity or negativity fluctuation. There are some fundamental investment strategies that can be used as baseline also which are frequently used by stock traders such as holding on to a stock in hope of seeing a near future positive fluctuation in order to make profit. Several other papers used Machine Learning models such as Random Forest and Support Vector Machine which will be used as baseline for this paper. The idea is how to improve the result using LSTM and bringing the historical context into the view for neural networks to obtain improved results in predicting near future stock prices.

# Problem Statement / Project Architecture

## Problem Statement

The Stock market has been running on volatility and extreme fluctuation due to influences from many external factors. The sentiment of the market, the health of each industry along with the overall economy are some of the big factors that can help predict some of the fluctuations. Although there have been many attempts at stock price predictions in the past, the time series stock market data combined with the aforementioned factors can give a good indication of problems that can be solved with modern machine learning techniques. Traditional applications of deep learning include speech recognition and image classification within which neural networks have proved their capability in learning to decode non-linear mappings between inputs and outputs. Utilizing such modern techniques to create a model that predicts the future of the stock market can result in increased customer confidence in the system, indicating increased stock purchases over time. In this project, we propose to forecast future movements in the stocks by leveraging Deep learning techniques using LSTM and GRU. We try to present a solution that can overcome the naive estimator effect and thus precisely estimating the next gradient change, which is one of the biggest challenges in predicting the stocks.

## Architecture

Diagram

Description automatically generated

Fig. 1. High level Architecture

A candlestick data chart is a financial chart that is used to describe price movements. A candlestick represents one day of data. It is very similar to a bar chart with four important chunks of information for that day: open, close, high and low as shown in figure 2. Each candlestick represents a datapoint in our dataset. A snapshot of the dataset is in figure 3. Each sample contains date, ticker, time bar, first trade price, high trade price, low trade price, last trade price, volume weight, volume and total trades attributes. Data preprocessing (data cleaning, data transformation) are performed on the data to make it fit for processing.

Long short-term memory (LSTM) networks are a type of artificial recurrent neural networks that are capable of learning order dependence in sequential prediction problems. In sequential data problems or the time series problems, the data changes occur over time and it becomes necessary to keep track of changes in order to make accurate predictions. For instance, in stock prediction problem, which is a classic time series problem, the prediction of whether the stock is bearish or bullish depends on the information learned over years. Classical machine learning models have no memory per say, which makes them incapable of referring back over time to learn patterns. When there is a significant change, these models must be retrained with new datasets to keep them updated and the process is very costly for huge datasets

Neural networks, especially recurrent neural networks were introduced to solve such time dependent problems. Although RNN’s performed better than the classical models, RNNs could retain memory up to certain levels only. Going back several steps and connecting information was difficult for RNNs. LSTMs in this regard, have a special ability to add or remove information to the cell state, regulated carefully using gates. Gates are composed of sigmoid neural net layer and a pointwise multiplicative operation. LSTMs use tanh function for activation of the states and sigmoid activation function for the node output. The outputs of this layer are zero and one specifying how much information of each cell must be let through. After successive epochs, the trained model is then tested for accuracy. The measure for accuracy is the loss function that trains the model for lower losses and increases the prediction accuracy.

# Data Exploration and Preprocessing

The dataset holds information on various stocks such as Apple (AAPL), Baidu Inc (BIDU), Chipotle Mexican Grill (CMG), Ebay (EBAY) etc. The dataset consists of 20-year samples of all these stocks. This dataset involves several .csv files where each csv file represents data over 20 years for a particular stock. To sum up, the dataset we have at present is for 5000 days (weekday data) x 41 stocks per day samples as shown in fig. 2. Incorporating all the data files into a single csv file to feed the model is a challenging task. Current implementation focuses only on the AAPL and GOOGL stock.

Diagram

Description automatically generated

Fig. 2. Structure of Data files

A sample of the dataset for AAPL stock is shown in fig. 3. The Stock data contains details such as High, low, low and high trade price, amount, and total trade value obtained every few minutes.

**A picture containing text, newspaper

Description automatically generated**

Fig. 3. Sample data for stock AAPL

The price difference for the AAPL stock on a day is shown in fig. 4. From the figure, we can visualize the dynamic nature of the stock and understand the complexity of the problem.

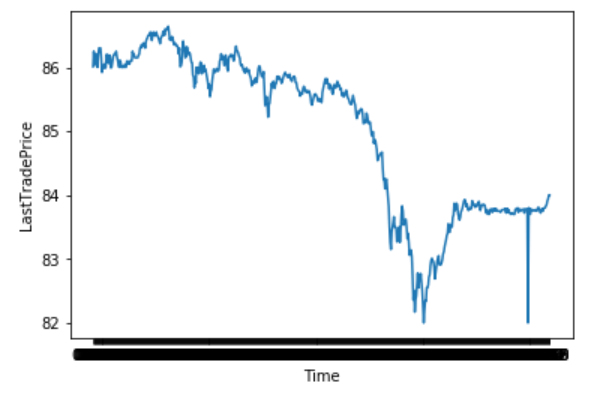


Fig. 4. Visualization of AAPL stock data

Since the stock market data is spread into different directories. Data processing involves combining the time series data together. The combined data is then cleaned to handling any missing values in the data set, handling any duplicate rows, to ensure right data format of the attributes etc. the data is further normalized to have values between 0 and 1 using MinMaxScaler. The processed dataset is divided into 75% training set and 25% test to be applied for the LSTM model. The visualization of the training set and test set for AAPL stock is shown in fig. 5 and fig. 6.

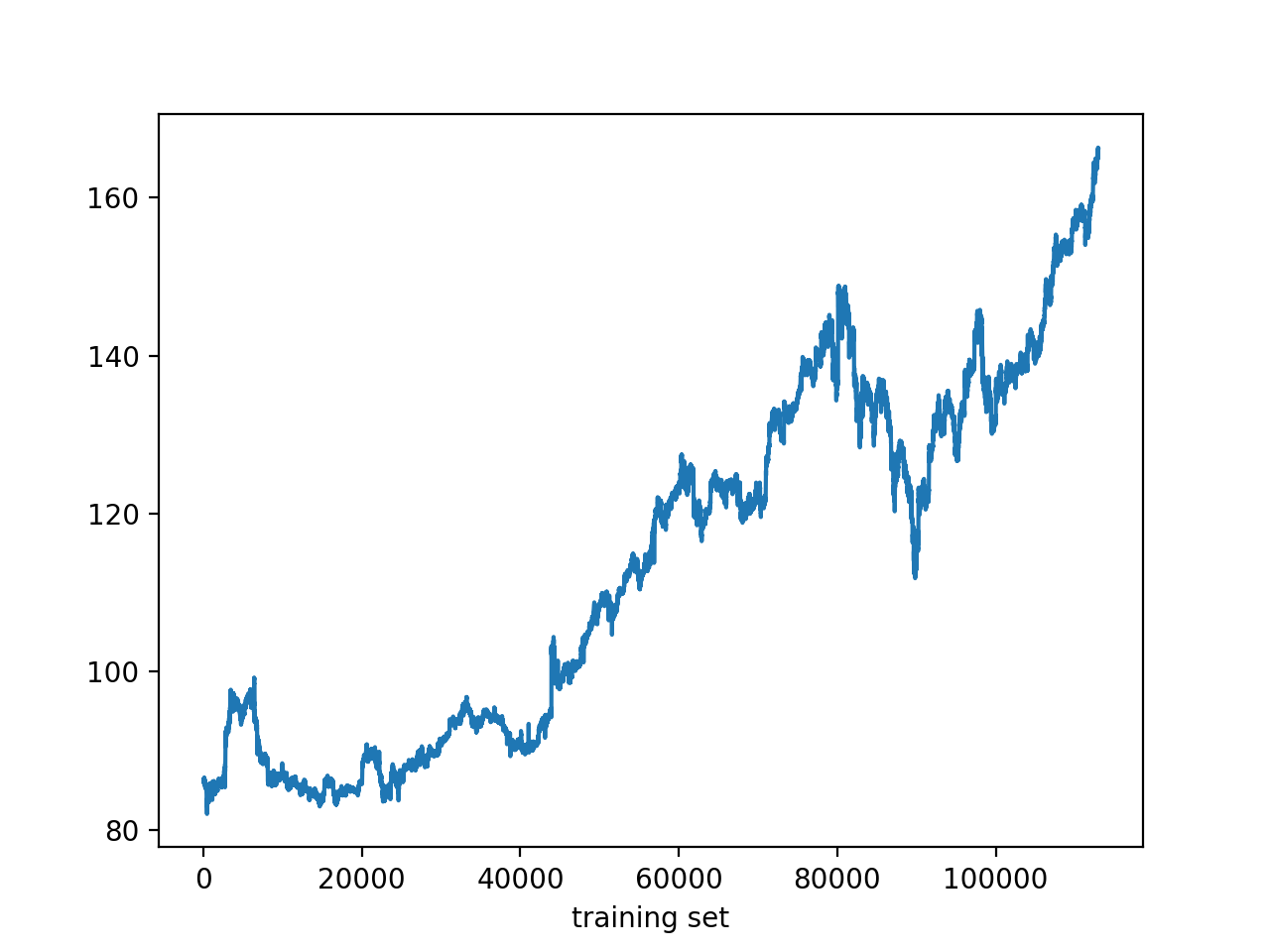


Fig. 5. Visualisation of training set for AAPL stock

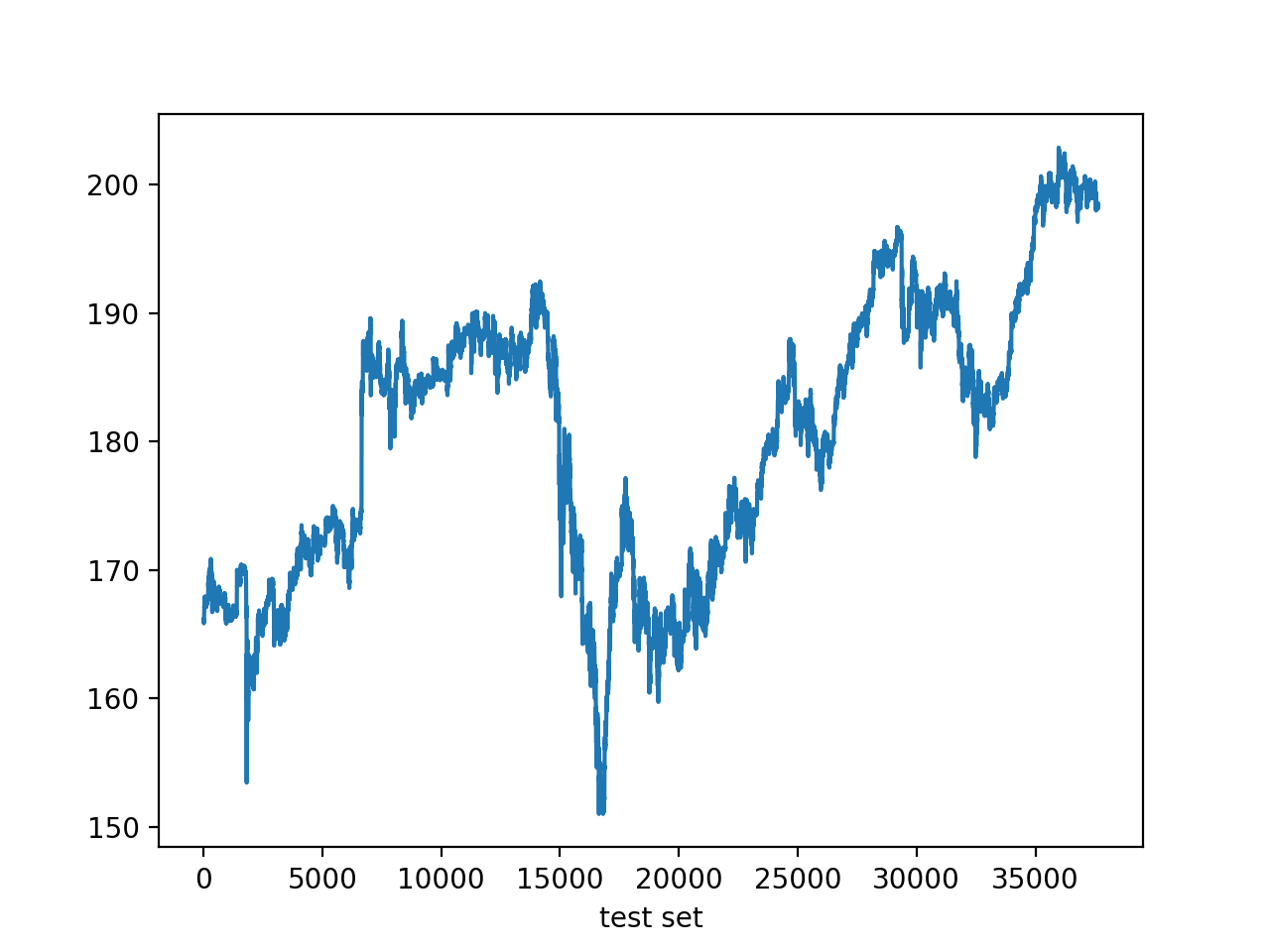


Fig. 6. Visualisation of test set for AAPL stock

## Tableau Visualizations

The stock data is imported in tableau to understand and analyze through different visualizations such as time series graph of stock price over period of different years and bar charts to plot Average closing price of various stocks over all the months of a year. The dashboard in tableau is embedded with a filter option to select the ticketr to see the visualizations for the stocks in which users are interested. Some important plots and charts are shown in fig 7.

Chart, line chart

Description automatically generated Chart, bar chart

Description automatically generated

Fig 7. Visualizations from Tableau

# Methods / System Design

## Long Short-Term Memory (LSTM)

LSTM Neural Networks are a type of Recurrent Neural that is designed to specifically handle long term dependencies. The RNN module is a simple design having a single tanh layer. LSTMS consists of four layers of tanh where each layer communicates in a unique way. The LSTM can remove or add information to a cell state through a structure called gates. The gates are made up of a sigmoid layer. The sigmoid layer can have values ranging between 0 and 1. The sigmoid layer values signify the amount of information to be passed through the gate. A value of one indicates that the information is transferred completely, and a value of zero indicates that there is no flow of information.

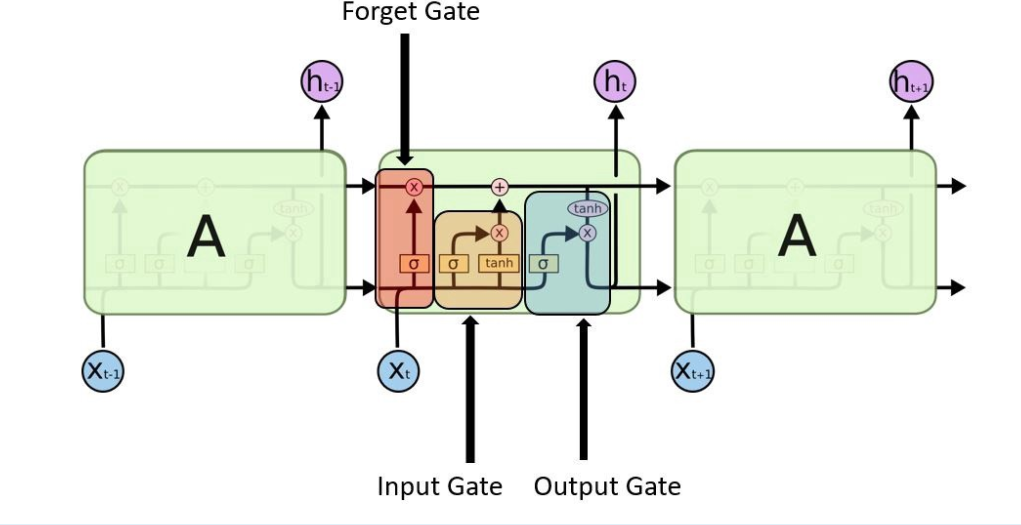
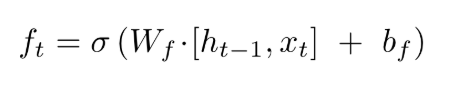


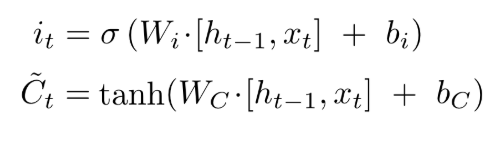
Fig. 8. LSTM Neural Network Architecture

The architecture of LSTM neural network is as shown in fig. 8. It is made up of three gates consisting of input gate, forget gate, and output gate. The design has 4 layers among which three of them are sigmoid layers connected to the three gates and the fourth layer is the tanh layer.

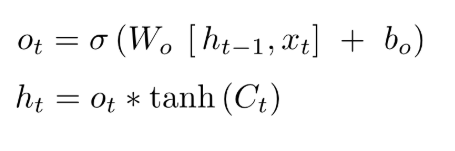
Forget gate layer -The first step in the LSTM model is to remove the information in the cell state c(t-1) with the help of sigmoid layer. Sigmoid layer outputs a value in the range (0,1) which helps in determining the amount of information to be discarded. The input given to sigmoid layer is x(t) and h(t-1) where h(t-1) is the previous output at time t-1 and x(t) is the current input sample at time t



Input gate layer – The next step in the model is to evaluate what new data in the cell state will be processed. The sigmoid layer is given x(t) and h(t-1) as input. The candidate values that can be added to the cell state is determined by the tanh layer. The formula for the input gate layer is given below.



Output layer – The final step is to determine the cell state output information. Sigmoid layer is part of this decision. Sigmoid layer will be used to set the information to be sent as output by taking x(t) and h(t-1) as input. This data is multiplied to only relay this part of the information as the output by the tanh layer. The formula is shown below to determine the output value.



The LSTM model has the ability to distinguish between recent and early samples via this architecture by assigning them various weights. This helps to boost stock price prediction results. The input sample here is considered to be independent by the LSTM model. In stock price prediction, this serves as an advantage. The current stock is linked to previous stock prices, so the model is capable of capturing this dependence to provide a better outcome.

## Gated Recurrent Unit (GRU)

Gated Recurrent Unit is the improvised version of Recurrent Neural Network and has a lot in common with LSTM.

While LSTM used cell state to maintain internal memory, GRU uses the hidden state to transfer information. It has two gates, a reset gate and an update gate.

The activation of the GRU at time t is a linear interpolation between the previous activation and the candidate activation

The candidate activation function is computed similarly to that of the traditional recurrent unit.



The reset gate is computed similarly to the update gate.



Diagram

Description automatically generated

The Update gate does the work of both forget and input gate of an LSTM – responsible for making decisions on what information to throw away and what new information to add.

The Reset gate is another gate used to decide how much past information to forget.

GRU is computationally more efficient considering fewer parameters and need less data to generalize. However, GRU does not have any mechanism to control the degree to which its state or memory content is exposed but exposes the whole state or memory content each time. LSTM can control how much memory content it wants to expose.

Given the similarities and differences in GRU and LSTM, the decision as to which of the models performs better is totally dependent on the dataset and corresponding task. Because stock prediction requires sequence-data, LSTM is a better choice as the accuracy of the prediction is of high importance.

# Evaluation Methodology

## Mean Squared Error (MSE)

 Mean Square Error (MSE)is used to calculate the mean of squares of actual values minus predicted values divided by the number of observations. Mathematically it is defined by the following formula:

A close up of a clock

Description automatically generated

With Minimizing MSE, one can observe that the performance of the model increases. Usually in the training step, the MSE starts to be a higher number but through different epochs, the number should become smaller. Based on our current implementation, the MSE of our model is 1.90. This is a considerably low number, but our implementation needs to be improved and perform the same way for other stocks. As it was discussed, this is a reiterative process, and our goal is to keep the MSE low through different changes we make to the model.

# Results

The processed data is given as input to the LSTM model. The LSTM neural network currently consists of four layers and a 20 % dropout is attained at each layer to prevent any overfitting. Utilized Adam optimizer with mean square error as loss function. Fig. 8 shows the preliminary results obtained. As we can see the model is able to predict the negative movements in stock price clearly. This model will further be fine-tuned to be able to get the predictions for positive movements in stock price. In our current implementation, we predict the closing of a day using the previous 60 days closing. Our approach was different from what we anticipated since our dataset format changed in between. Given the same implementation, the same approach can be applied to more granular data.

Chart, line chart, histogram

Description automatically generated

Fig. 9. AAPL stock price Prediction

Chart, line chart

Description automatically generated

Fig. 10. AAPL stock price Prediction

The result shows promising prediction values for predicting the future of stock prediction. The normal fluctuation in the stock market can be easily visualized in the above diagrams. Since the amount of data is limited in our case, the accuracy of the model is not what we expected but it can be improved if a more granular dataset is used.

## Python flask-based web app

The neural network model built is deployed on AWS (Amazon Web services) EC2 instance using python flask server for backend and HTML, CSS for frontend for the user to interact with. The users can see the analysis page with visualizations and interactive plots from tableau dashboard which includes filters with tickers. The predictions page shows the predictions of GOOGL from both tableau and python. The images of web app are shown in the fig 11.

Graphical user interface, chart

Description automatically generated

Fig 11. Web App Screen Shots

# Further Research

Future work would consist of using more granular data to create a model that is more generic and works with multiple stocks. Our current model works with one stock at a time since the dataset is in time series and the prediction value is the closing price of each stock using the previous 60 days. Using different model optimizers can also be explored to get a higher precision. Due to lack of time, we were unable to compare different models with different amounts of layers and dropouts which can be explored furthermore. In order to test the outcome, different values can be used as input data to validate some assumptions about the stock data. This technique can be applied to find the right amount of data and understand the importance of time in predicting the value of the stock market.

# Conclusions

The stock market data consists of samples recorded every day for the last 20 years. Since the dataset is large and each stock data is spread among multiple directories, the stock market prediction is currently analyzed for stock AAPL and GOOGL. We can see that the LSTM model is able to trace the negative movements precisely. This model will further be optimized to accommodate prediction in the positive movements of stock price. The stock data spread among multiple directories will be combined for the rest of the stocks and the LSTM model will be applied to get predictions on the entire dataset.

References

1. Kim, Kyoung-Jae, and Won Lee. "Stock Market Prediction Using Artificial Neural Networks with Optimal Feature Transformation." *Neural Computing & Applications* 13.3 (2004): 255-60.
2. Tilakaratne, C., M. Mammadov, and S. Morris. "Modified Neural Network Algorithms for Predicting Trading Signals of Stock Market Indices." *Journal of Applied Mathematics & Decision Sciences* 2009 (2009): 1-22.
3. L. Yu, H. Chen, S. Wang and K. K. Lai, "Evolving Least Squares Support Vector Machines for Stock Market Trend Mining," in *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 87-102, Feb. 2009
4. Zahedi, Javad, and Mohammad Mahdi Rounaghi. "Application of Artificial Neural Network Models and Principal Component Analysis Method in Predicting Stock Prices on the Tehran Stock Exchange." *Physica A: Statistical Mechanics and Its Applications* 438 (2015): 178-87.
5. E. W. Saad, D. V. Prokhorov and D. C. Wunsch, "Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks," in *IEEE Transactions on Neural Networks*, vol. 9, no. 6, pp. 1456-1470, Nov. 1998.
6. Y. Kwon and B. Moon, "A Hybrid Neurogenetic Approach for Stock Forecasting," in *IEEE Transactions on Neural Networks*, vol. 18, no. 3, pp. 851-864, May 2007.
7. Yu, Pengfei, and Xuesong Yan. "Stock Price Prediction Based on Deep Neural Networks." *Neural Computing and Applications (2019).*
8. D. M. Q. Nelson, A. C. M. Pereira and R. A. de Oliveira, "Stock market's price movement prediction with LSTM neural networks," 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, 2017, pp. 1419-1426.

1. [↑](#footnote-ref-1)