CSE 510: Applied Deep Learning

Project 2: Time-Series Prediction

Report

Team Members:

Name:	Nitin Kulkarni	Gautam Suryawanshi
UB Person Number:	50337029	50337017
UBIT Name:	nitinvis	gautamja

Part I: Preparing the dataset for training

1. **Choose the dataset.**
   We chose the Walmart stock dataset from 01/01/2000 to 12/31/2020 from Yahoo Finance.

2. **Extract and describe the main statistics about the dataset and provide visual representation of the dataset.**
   The dataset contains 5031 entries with 7 features.

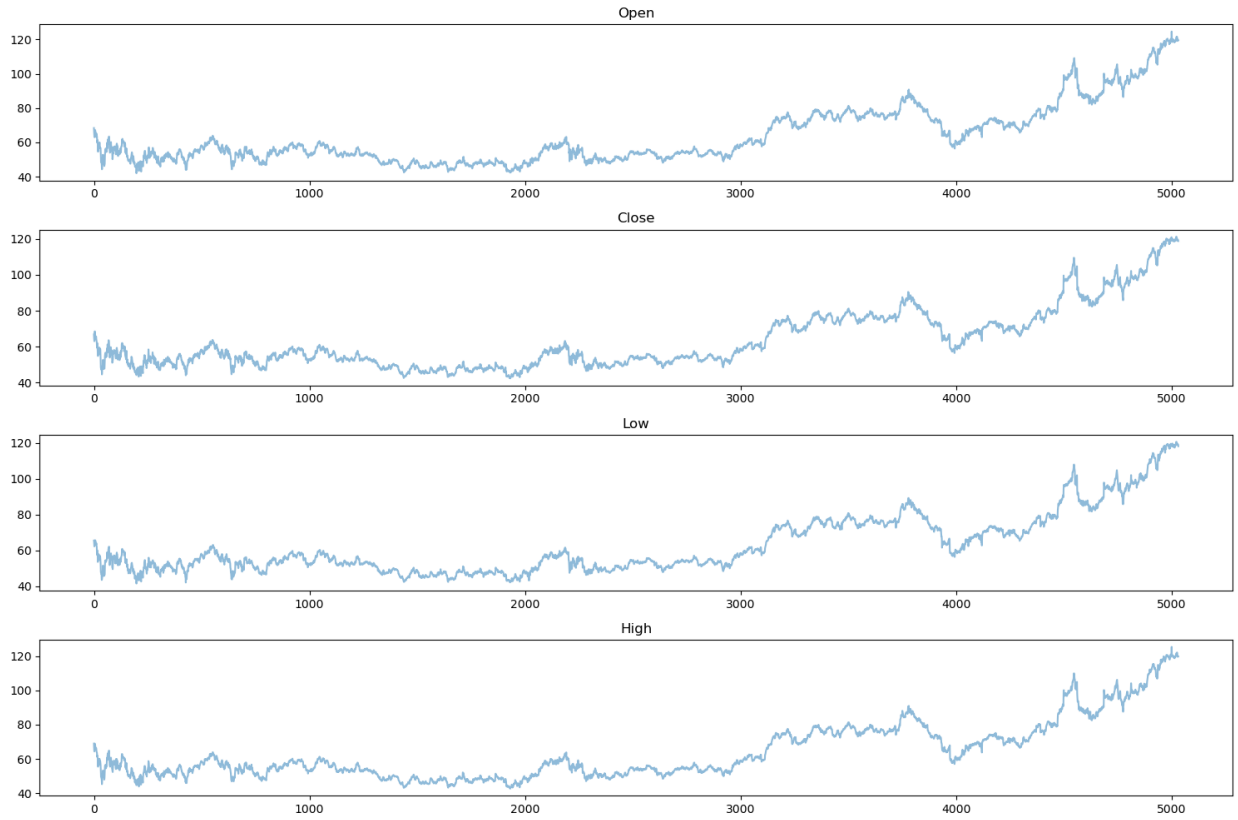
   The features and their descriptions are given in the table below:

Feature	Data Type	Description
Date	Object	It represents the date. It's in the format YYYY-MM-DD.
Open	Float64	It represents the price at which the stock opened for trading.
High	Float64	It represents the highest price the stock reached during that day's trading period.
Low	Float64	It represents the lowest price the stock reached during that day's trading period.
Close	Float64	It represents the price at which the stock closed for trading.
Adj Close	Float64	Adjusted closing price amends a stock's closing price to accurately reflect that stock's value after accounting for any corporate actions.
Volume	Int64	It represents the total number of shares that are actually traded (bought and sold) during the trading day.

The dataset description is as follows:

	Open	High	Low	Close	Adj Close	Volume
Count	5031	5031	5031	5031	5031	5031
Mean	63.352913	63.910212	62.807056	63.362401	51.747292	1.106437e+07
Standard Deviation	16.866083	16.911829	16.828010	16.870848	20.336000	6.846935e+06
Minimum	42.000000	42.680000	41.437500	42.270000	29.720531	2.031400e+06
25%	51.115000	51.750000	50.565000	51.174999	36.933237	6.698600e+06
50%	56.500000	57.110001	55.759998	56.419998	41.810890	9.040200e+06
75%	73.800003	74.165001	73.350003	73.770001	64.352356	1.330850e+07
Maximum	124.599998	125.379997	120.699997	121.279999	120.216911	9.678680e+07

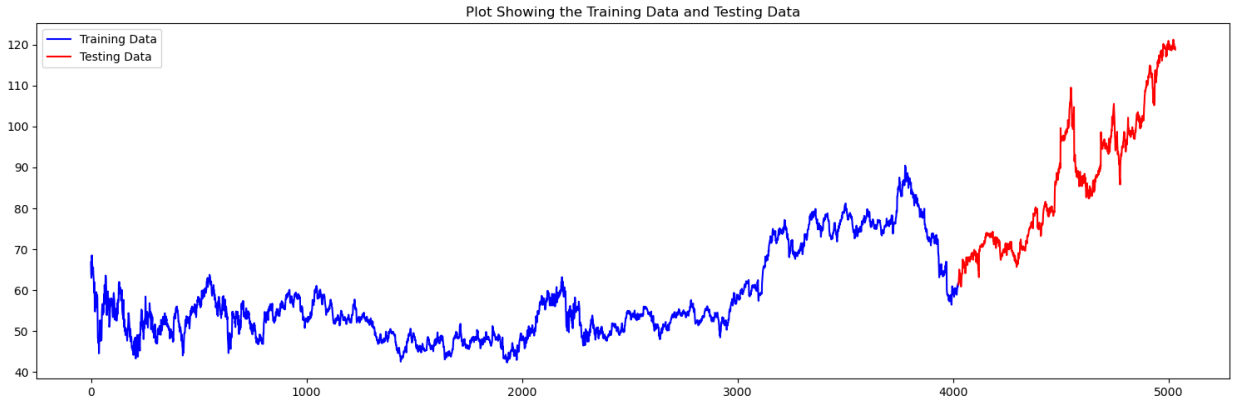
The following figure shows the plot of the stock features 'Open', 'Close', 'Low', 'High' over time:



3. Preprocess the dataset for training (e.g. cleaning and filling the missing variables, split between training/testing/validation)

The dataset was clean. There were no missing entries. I split the dataset into training data and testing data in the ratio 80:20. The first 80% of the data is used to predict the last 20% of the data.

The following figure shows the training data from which we will try to predict the testing data:



Part II: Classical Time Series Forecasting Methods

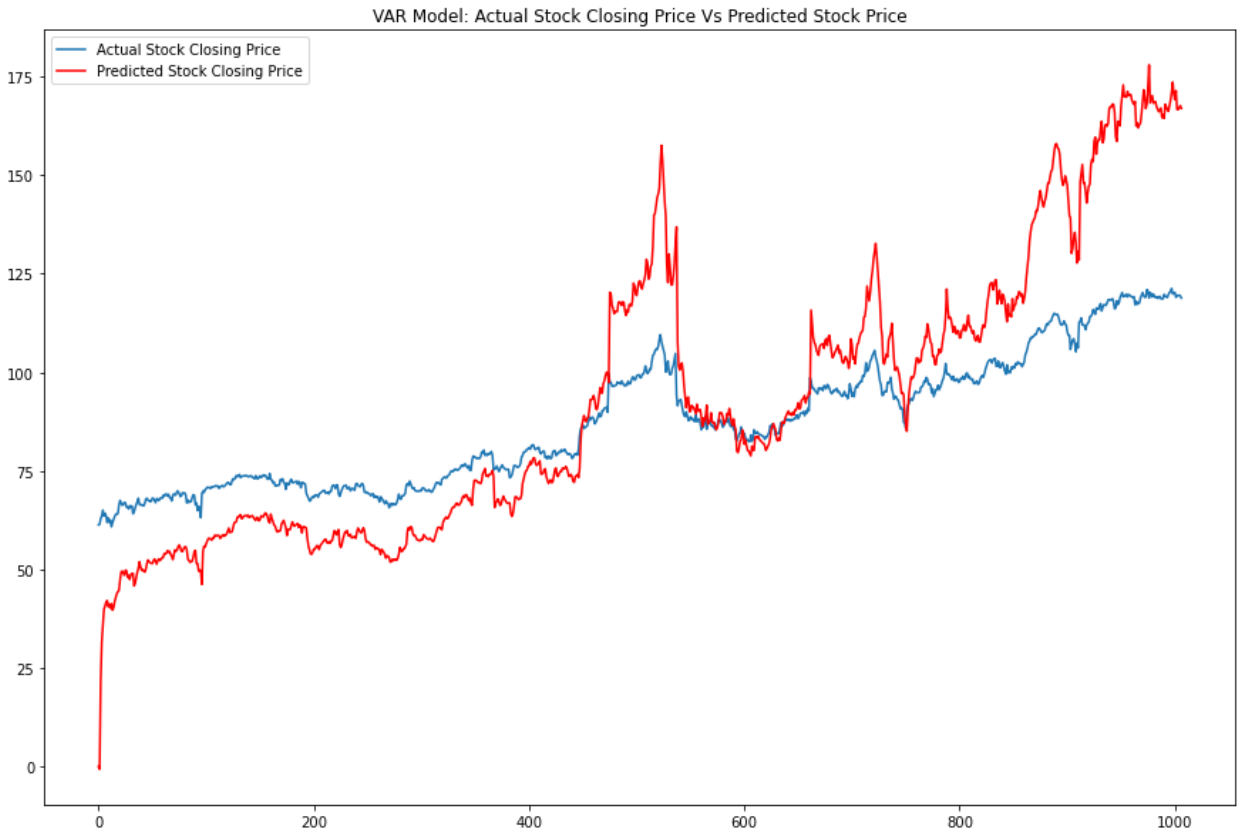
1. **Choose the features and targets in the dataset.**
   We chose the following features and targets for the three different setups.
   Features: Date, Open, High, Low, Adj Close, Volume.
   Target: Close

2. **Apply statistical algorithms (min 3 algorithms) to forecast the values. Possible algorithms include: ARIMA, VAR, SARIMAX, etc.**

   ARIMA Model: For the ARIMA Model we chose to predict the stock closing price based on past stock closing price. We have referenced the ARIMA model code based on the demo shown by the Professor in class. The model seems to be accurate and produces a Mean Squared Error of **1.137516253457746**.

Auto Regression Model: For the Auto Regression Model we predicted the stock closing price based on the past values. Auto Regression model requires the data to be stationary so we made it stationary by performing first differencing. The model seems to be accurate and produces a Mean Squared Error of **0.10257225205346769**.

Vector Auto Regression Model: Vector Auto Regression is a multivariate prediction model. We used the features **Open** and **Close** to make the predictions and extracted the stock closing price predictions. The model didn't perform as well as the ARIMA or AR models and produced a Mean Square Error of **416.5022697772365**.



3. Provide the comparison of the results of different statistical models you have used. This can be in the form of graph representation and your reasoning about the results.

We can see from the results that the AR model performs the best, followed by the ARIMA model and finally the VAR model.

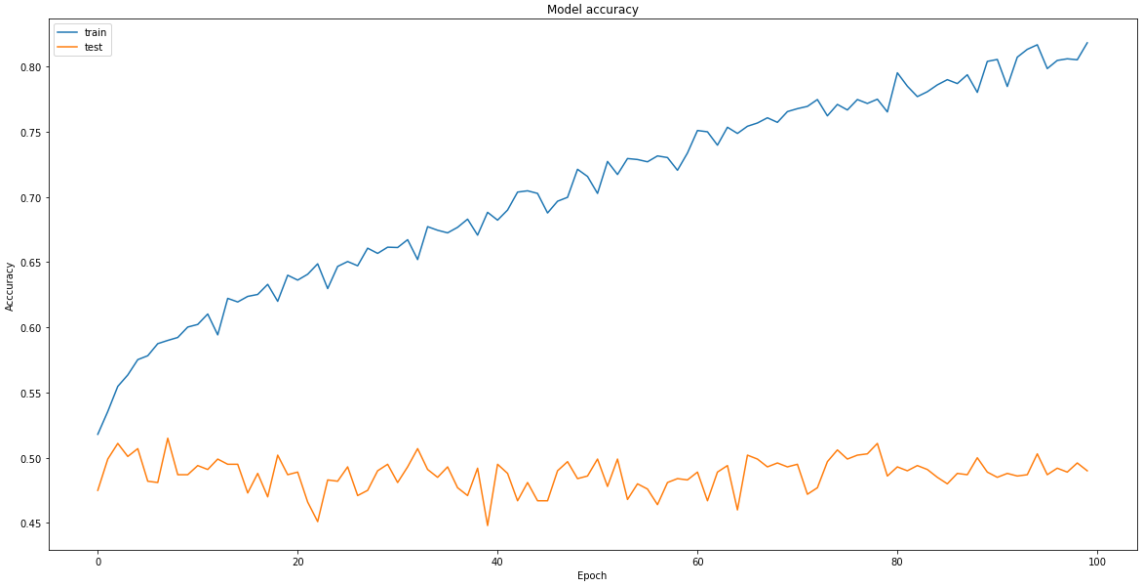
	ARIMA	**AR**	**VAR**
MSE	1.137516253457746	0.10257225205346769	416.5022697772365

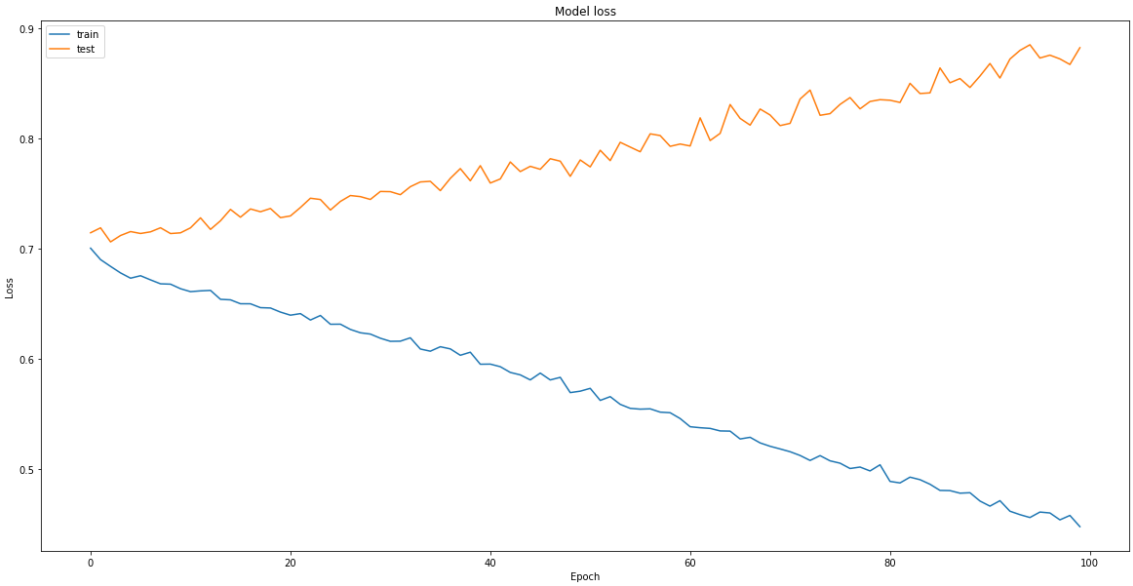
Part III: Deep Learning Time Series Forecasting Methods

1. **Apply MLP to predict the value. Show the results on 3 different MLP setups (#layers, activation functions, learning rate, layers structures, etc.)**

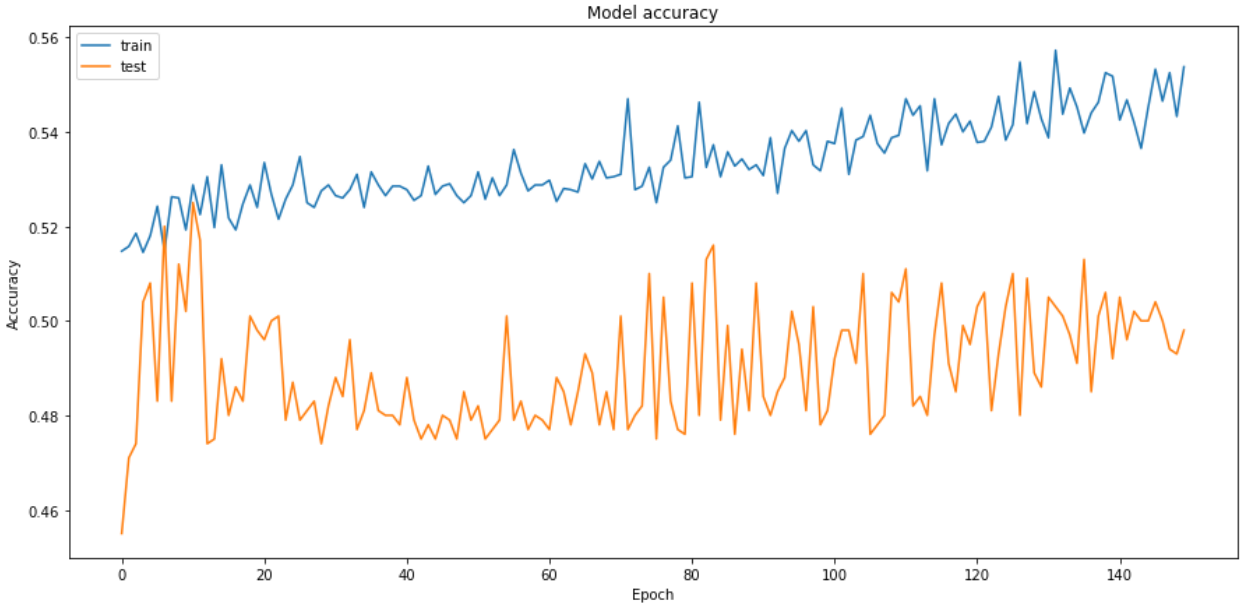
   For MLP we have referenced the code based on the demo given by the Professor in class.

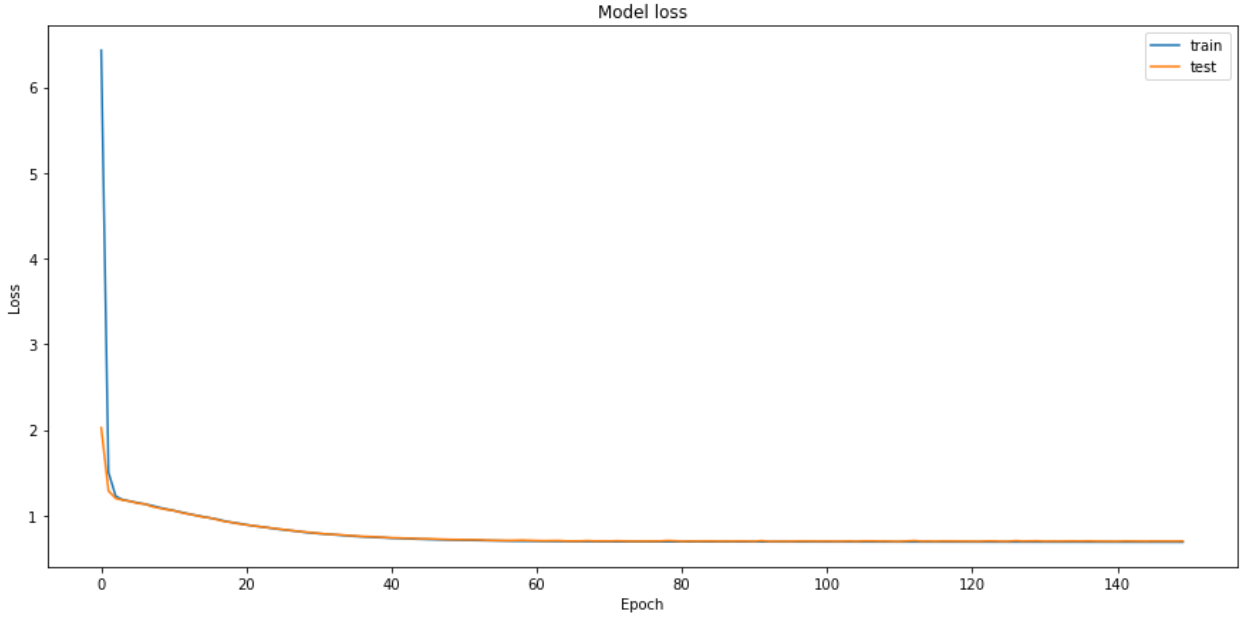
   Setup 1: For the first setup we went with a simple Neural Network with 1 layer containing 256 nodes, we didn't add any Batch Normalization or regularizers to see what a basic network would do. We can see that although the training accuracy increases to around 80% the test accuracy remains around 50% and though the training loss decreases the test loss increases. This model is clearly overfitting.



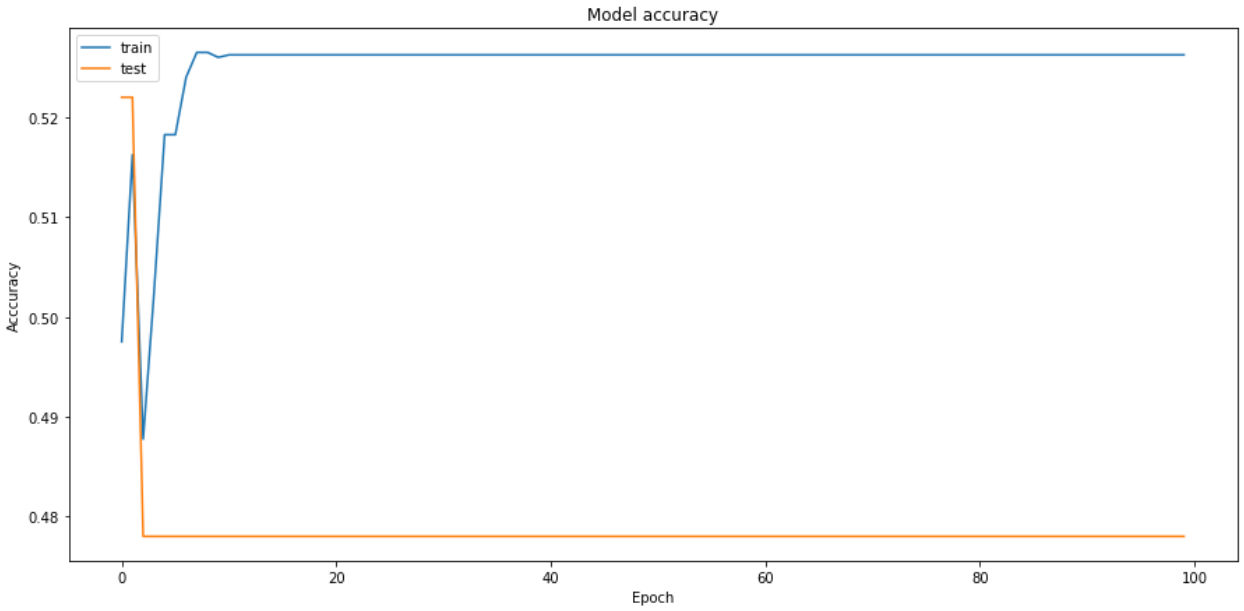


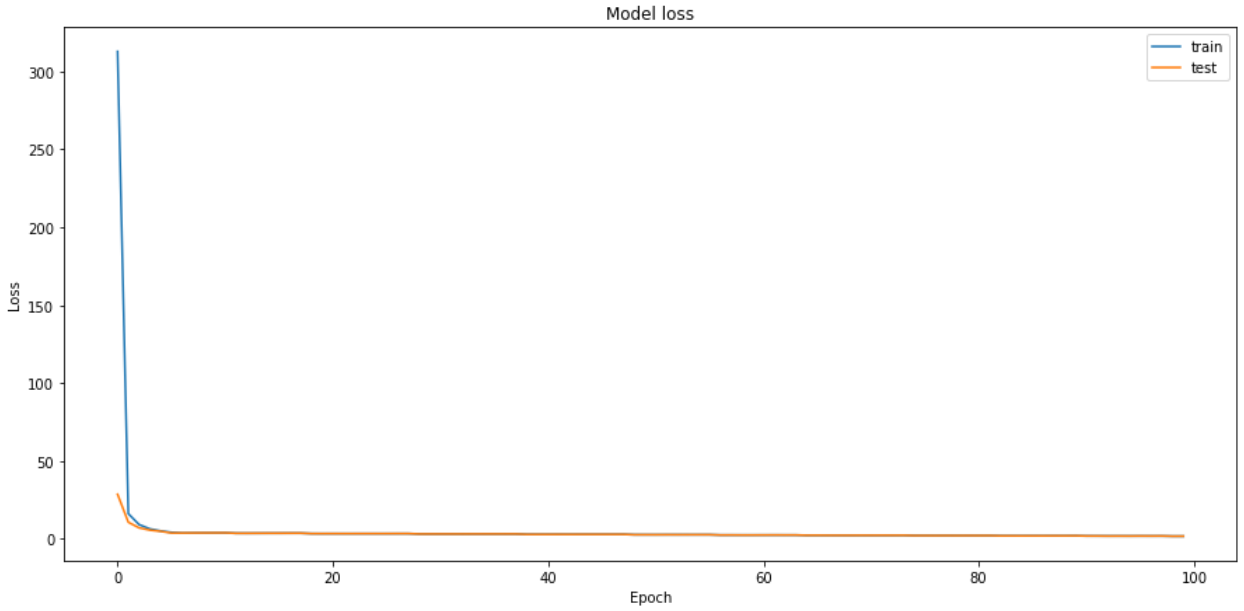
Setup 2: For the second setup we added in the l2 regularizer and increased the number of nodes in the first layer to 512 and added in another layer with 128 nodes in it. We also reduced the learning rate on hitting a plateau and specified the minimum learning rate to be 0.0001. This produced results where the model produced around 55% accuracy on the training dataset and about 50% on the test dataset. The model loss remains almost constant for both training and test datasets. This model doesn't overfit as much though the accuracy isn't increased.





Setup 3: For the third setup we added both l1 and l2 regularizers and increased the number of nodes in the layers to 1024 and added in three more layers with 1024 nodes. We also reduced the learning rate on hitting a plateau and specified the minimum learning rate to be 0.0001. We reduced the batch size to 64. This produced results where the model produced 52.63 % accuracy on the training dataset and about 47.80 % on the test dataset. This accuracy was constant. The model loss remains almost constant for both training and test datasets. This model doesn't overfit though the accuracy isn't increased.





2. Apply RNN or LSTM architecture to predict the value.

LSTM: In LSTM we are predicting the stock closing price based on the past values of the stock closing price. We are using 60 timesteps to predict the value. By adding dropout and increasing the batch size, our model performs very well. In comparison to other models, the LSTM model seems to predict the value very close to actual result, which can be easily seen from the graph.

3. Discuss and provide the results of predicting the values using different deep learning structures.

We get very good results from LSTM; the predicted values are very close to the actual test values. Our LSTM model uses 60 timesteps to determine the stock closing price. For MLP, we are trying to predict whether the stock price will go up or down. Unfortunately, none of the MLP models could make predictions on the test dataset with more than 50 % accuracy. We see a clear trend that without regularization in the first MLP setup we are achieving 80 % accuracy on the training data but only around 50 % on the test data, and with regularization our training accuracy drops to around 55 % but test accuracy remains the same at around 50 %. Thus, the models don't perform well on predicting whether the price will go up or down.