Time Series Forecasting

STOCK PRICE PREDICTION USING LSTM

INTRODUCTION

Time Series Data

Temporal data refers to data that varies with time. It is indexed using timestamps. The process of extracting patterns and knowledge from temporal data is called temporal data mining. Time series prediction and forecasting is a part of temporal data mining. It has been a major research area and finds wide applications across the world of business, finance, energy demand prediction, weather analysis and climate prediction.

There are several techniques that are employed for time series forecasting. These techniques can be broadly grouped into *conventional techniques* and *deep learning based techniques.*

Conventional Techniques - ARIMA

ARIMA stands for Auto Regressive Integrated Moving Average.

"Auto Regression" refers to regression being used to fit the model on the dataset. Thus, historical values of the time series are being used to predict the value for the next lag (time step) of the series.

"Integrated" refers to the use of differencing to remove non-stationary behavior of the time series. Non-stationary behavior of data points refers to the mean, variance and covariance values of the data changing over time. Such data cannot be used for forecasting. It is modelled by the following equations:

i. Random walk without drift:

$$X_t = X_{t-1} + \varepsilon_t$$

ii. Random walk with drift:

$$X_t = X_{t-1} + \varepsilon_t + \alpha$$

where, X_t = value at time t,

X_{t-1} = value at time $t - 1$,

α = drift and ε_t = white noise component (mean = 0 and variance = σ^2)

Using differencing non-stationary data can be converted into stationary data as:

$$X_t - X_{t-1} = \varepsilon_t$$

or

$$X_t - X_{t-1} = \alpha + \varepsilon_t$$

"Moving Average" refers to the average of the data points calculated in a given time period.

Drawbacks

The conventional techniques for Time Series Forecasting have the following major drawbacks:

- They cannot model accurately non linear trends, such as seasonality, in the dataset.
- Their performance severely deteriorates for long term predictions (time period greater than a year).

Thus, Hybrid models such as ARIMA + ANN, ARIMA + SVR etc. and Deep Learning models such as Convolutional Neural Networks, Recurrent Neural Networks and LSTM (Long Short Term Memory) models were devised to overcome the shortcomings of the conventional techniques to Time Series Forecasting.

Our Project

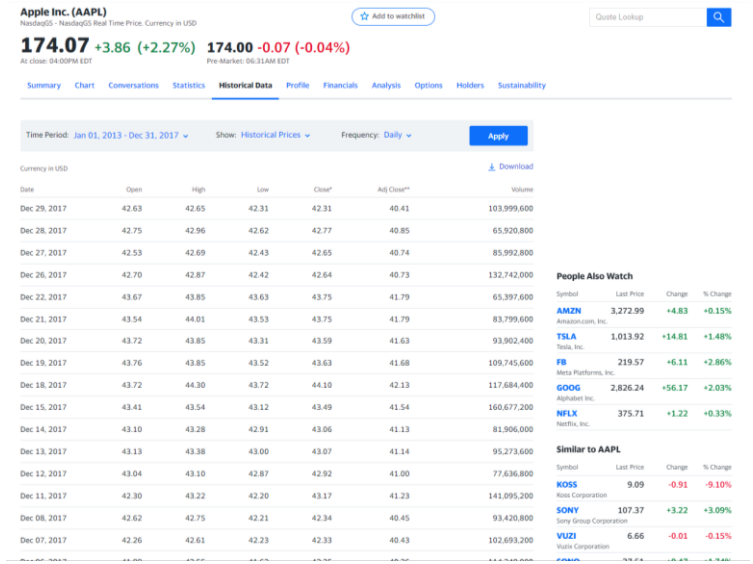
In this project, we aim to develop a LSTM based deep learning model for stock price prediction. The following methodology has been used to implement this work:

- Data Collection – Dataset from Yahoo Finance of AAPL (Apple) Stock Prices from 1 January 2013 to 31 December 2017.
- Preprocessing – Min-Max Scaling and Sliding Window Sequences.
- Model Training and Hyperparameter Tuning.
- Model Evaluation and Selection.
- Prediction and Visualization.

DATASET

Dataset Selection

Dataset from Yahoo Finance of AAPL (Apple) Stock Prices from 1 January 2013 to 31 December 2017 (1259 datapoints). This period has been selected since it has a period of strong growth (between 2013 to 2014), followed by a period of decline in the stock price (between late 2014 to 2015) and then years of successive growth till 2017. This makes the dataset quite challenging for accurate prediction.



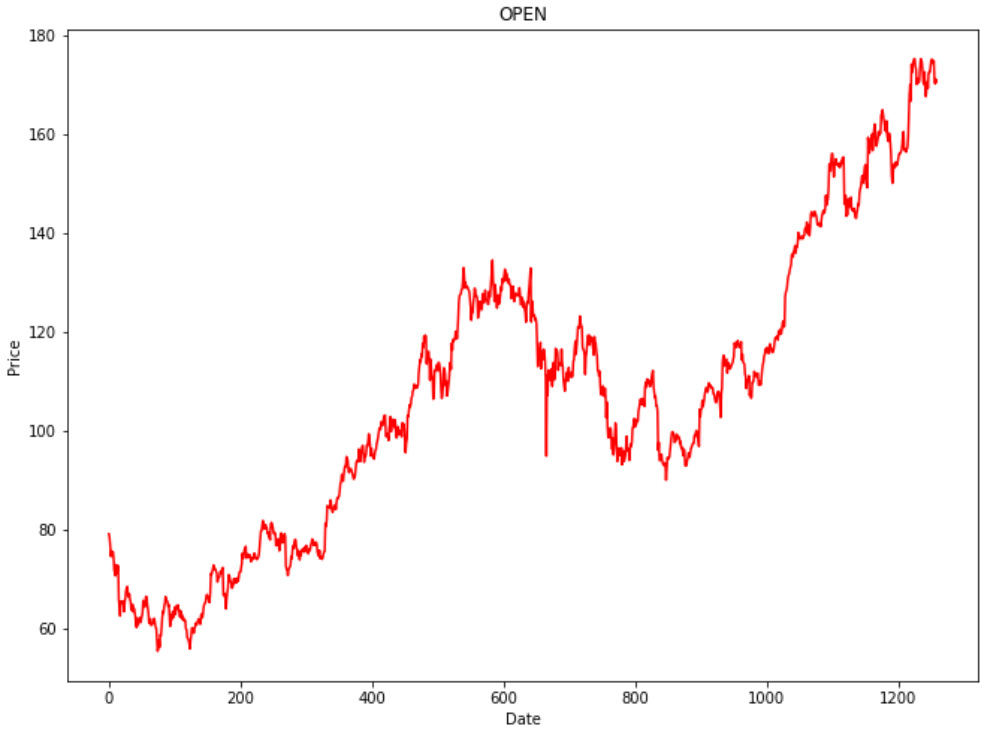
Sample Data

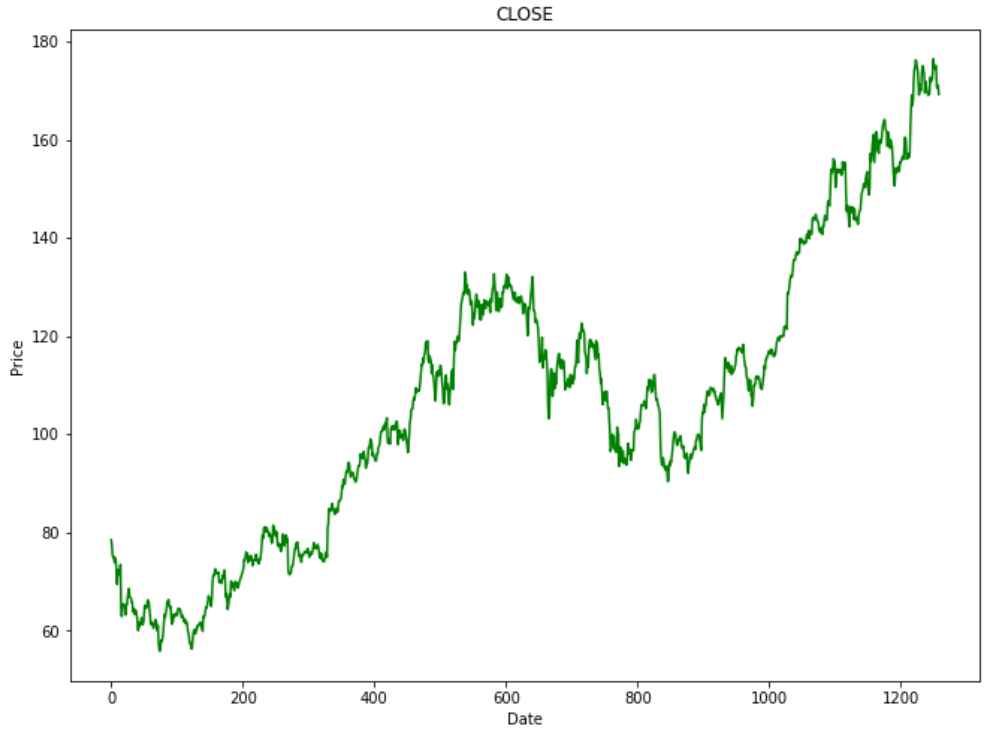
Date	Open	High	Low	Close	Adj Close	Volume
02-01-2013	79.11714	79.28571	77.37572	78.43285	68.68754	140129500
03-01-2013	78.26857	78.52428	77.28571	77.44286	67.82053	88241300
04-01-2013	76.71	76.94714	75.11857	75.28571	65.9314	148583400
07-01-2013	74.57143	75.61429	73.6	74.84286	65.5436	121039100
08-01-2013	75.60143	75.98428	74.46429	75.04429	65.71999	114676800

DATA PREPROCESSING

Data Visualization

The dataset consists of seven columns. Of these, only the ‘Date’, ‘Open’ and ‘Close’ were used to train the model and predict future stock prices. Note that the time series is highly non-linear.





Min-Max Scaling

All the columns apart from ‘Date’, ‘Open’ and ‘Close’ were dropped. The ‘Date’ column was used to index the dataset and Min-Max Scaling was applied to these columns using the ‘sklearn.preprocessing.MinMaxScaler’. This will scale the dataset between 0 and 1. This prevents very high or very low values in the dataset from introducing skew errors.

```
scaler = MinMaxScaler()
print(data_final.loc[1, "Date"].split())
data_final['Date'] = pd.to_datetime(data_final['Date'].apply(lambda x: x.split()[0]))
data_final.set_index('Date', drop = True, inplace = True)
data_final.loc[:, ["Open", "Close"]] = scaler.fit_transform(data_final.loc[:, ["Open", "Close"]])
data_final.head()
```

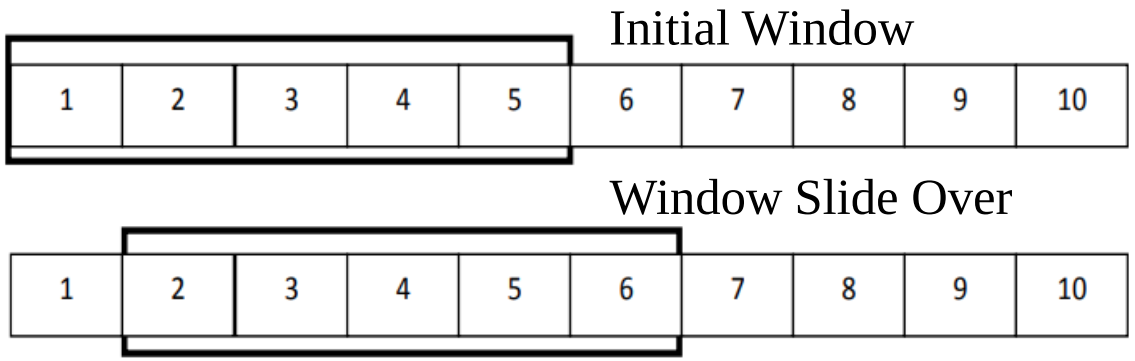
Train-Test Split

Next, the dataset was split into training and testing datasets. 70% of the dataset was used as the training dataset and 30% was used as the testing dataset.

```
#Train - Test split
train_len = round(len(data_final) * 0.7)
training_data = data_final[:train_len]
training_data.head()
testing_data = data_final[train_len:]
testing_data.head()
```

Sliding Window Methodology

Next, the training and testing data were used to create sliding window datasets. A window size of 50 was chosen. Thus, 50 successive timesteps were used as a training sequence and the next value (51^{st}) was used as the prediction label. Thus, this way the Time Series Forecasting is modelled as a supervised learning problem.



Sliding Window Creation

```
#Create training and testing seqeunces along with their prediction labels
def makeSeq(dataset):
  sequences = []
  labels = []
  starting = 0

  for stopping in range(50, len(dataset)):
    sequences.append(dataset.iloc[starting : stopping])
    labels.append(dataset.iloc[stopping])
    starting += 1
  return (np.array(sequences).astype('float32'), np.array(labels).astype('float32'))
```

```
training_seq, training_labels = makeSeq(training_data)
testing_seq, testing_labels = makeSeq(testing_data)
```

```
print(np.shape(training_seq), np.shape(training_labels))
print(np.shape(testing_seq), np.shape(testing_labels))
```

```
(831, 50, 2) (831, 2)
(328, 50, 2) (328, 2)
```

MODEL TRAINING & SELECTION

Model Training

After the preprocessing is done, the data is in the format required for training a model for prediction. We have used Keras (an interface for the TensorFlow library) to implement the LSTM model.

Various LSTM models of different complexities were explored to choose the best model. The use of stacked LSTM models was also explored. The following parameters were chosen to be tuned to generate the best model:

- Number of Layers : [1, 2, 4] (Two and Four LSTM Layers are Stacked LSTM models)
- Number of Epochs : [30, 80]
- Batch Size : [16, 32]

Each LSTM Layer had 50 neurons. The loss function chosen was ‘mean_squared_error’ with ‘ADAM’ optimiser. The evaluation metric was chosen as ‘mean_absolute_error’. A 20% dropout layer was added after each LSTM layer to prevent over-fitting.

Grid Search

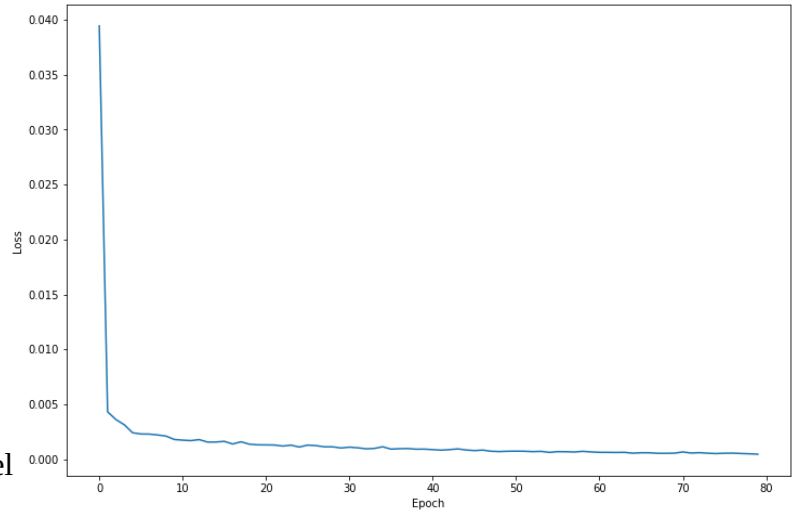
A grid search over the parameters specified in the previous slide was employed to find the best model. The following testing results were obtained from the 3 x 2 x 2 = 12 models evaluated:

No. of LSTM Layers	Epochs	Batch Size	Testing Loss	Testing MAE
1	30	16	0.00049917	0.0171
1	30	32	0.0019	0.0379
1	80	16	0.00043985	0.0166
1	80	32	0.0005685	0.0188
2	30	16	0.0005483	0.0174
2	30	32	0.0008412	0.0235
2	80	16	0.0008584	0.0241
2	80	32	0.0010	0.0264
4	30	16	0.0041	0.0566
4	30	32	0.0010	0.0240
4	80	16	0.0016	0.0334
4	80	32	0.0008636	0.0238

Model Selection

As can be seen form the previous slide, the Stacked LSTM models overfit the data and thus perform poorly over the testing sequences.

Thus, the model that was selected for prediction is a single LSTM layer model with batch size as 16 and trained over 80 epochs. The training loss (MSE) was 4.4828×10^{-4} and training MAE was 0.0159. The testing metrics were loss (MSE) as 4.3985×10^{-4} and MAE as 0.0166.

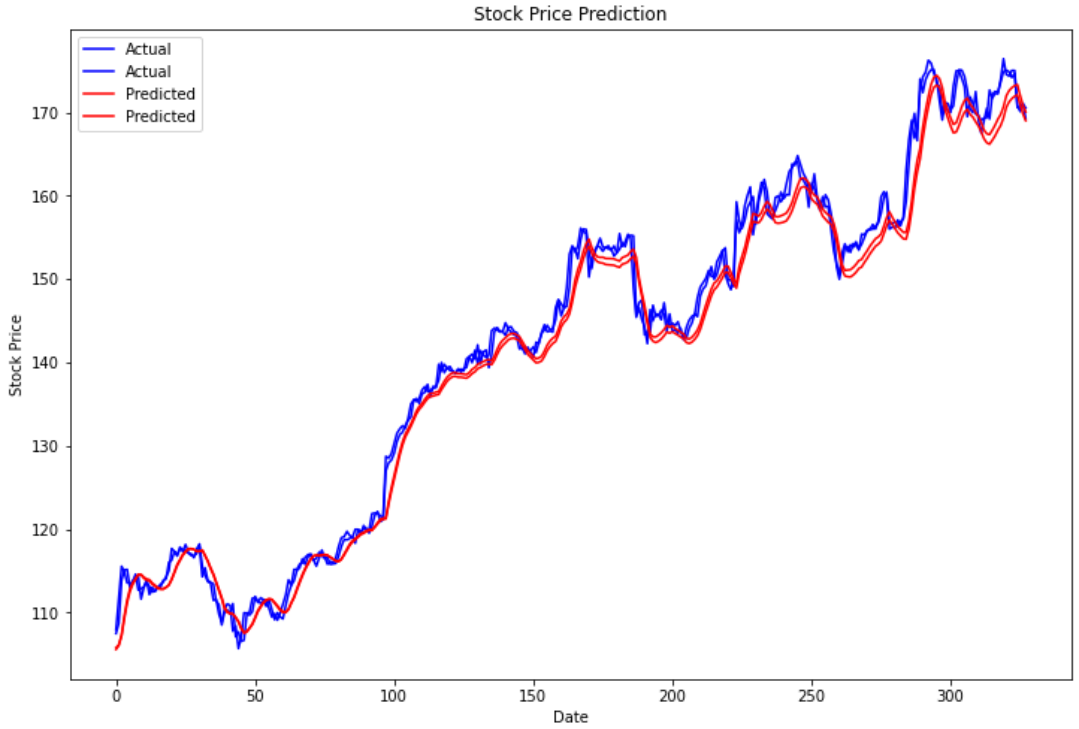


The plot shows the Training Loss vs Epochs for the selected model

RESULTS

Predictions

The model developed was used to predict the stock prices of AAPL (Apple) over the testing sequences. A total of 328 labels were predicted and plotted against the known labels as shown:



Actual vs Prediction Samples

Predicted		Actual	
Open	**Close**	**Open**	**Close**
104.48028	104.76302	107.51	107.95
105.31435	105.49885	108.73	111.77
107.40701	107.44725	113.86	115.57
110.79767	110.71225	115.12	114.92
113.19655	113.14156	115.19	113.58
114.22514	114.29941	113.05	113.57

FUTURE SCOPE

Future Work

In this project, we have focused our efforts on predicting the opening and closing prices of AAPL stock prices by accurately modelling the historical trends.

However, the stock market is very volatile and stock prices of a firm can be affected by various external parameters such as the stock prices of its competitors.

Thus, this work can be extended to model the multivariate time series relationship between the stock prices of Apple and its competitors such as Google and Microsoft thereby increasing the prediction accuracy.

REFERENCES

References

1. S. Siami-Namini, N. Tavakoli, and A. S. Namin, “The Performance of LSTM and BiLSTM in Forecasting Time Series”, 2019 IEEE International Conference on Big Data (Big Data), IEEE (2019).

2. P. T. Yamak, Li Yujian, and P. K. Gadosey, “A Comparison Between ARIMA, LSTM, and GRU for Time Series Forecasting”, Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence (2019).

3. G. P. Zhang, “Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model”, Neurocomputing Vol. 50 pp. 159-175 (2003).

4. P. Mondal, L. Shit and S. Goswami, “Study of Effectiveness of Time Series Modeling (ARIMA) in Forecasting Stock Prices”, International Journal of Computer Science, Engineering and Applications Vol. 4.2 (2014).

5. S. Siami-Namini, N. Tavakoli, and A. S. Namin, “A Comparison of ARIMA and LSTM in Forecasting Time Series”, 2018 17th IEEE international conference on machine learning and applications (ICMLA), IEEE (2018).

6. V. K. R. Chimmula, and L. Zhang, “Time Series Forecasting of COVID-19 Transmission in Canada using LSTM Networks”, Chaos, Solitons & Fractals Vol. 135 (2020).

THANK YOU

PRESENTED BY:

SOUMYA SOMANI

BAIBHAV PADHY