Financial Time Series Analysis with Machine Learning

Submitted as Research Report for Master of Data Science in SIT723

SUBMISSION DATE: 06/06/2022
T1-2021

Rakshith Rachenahalli Bettegowda
STUDENT ID 220500496
COURSE - Master of Data Science (S777)

Abstract

For a long time, academics have debated how best to predict the movement of stock prices. Careful modelling and variable design may produce models that adherents of the efficient market hypothesis may be able to use to make accurate forecasts of stock prices and movement patterns. Recently developed data mining methods have been employed by academics in their pursuit of patterns in stock price fluctuations. We use machine learning and deep neural networks when trying to predict the price of a stock. We studied the stock prices of Apple, Microsoft, Tesla, Alphabet, and Amazon from March 2017 to March 2022. So that our forecasting system could be more accurate in its predictions, we developed regression models based on deep learning and LSTM networks. The ARIMA model with a custom p, d, and q value is used provided the best result. In the 80/20 train test split, single-layer LSTM models showed no conversion. The results showed that a two-layer LSTM model with five dense layers performed significantly better than a one-layer model. We then compared the results of various LSTM optimizers and then optimised the LSTM hyperparameters to ensure that validation losses remained stable despite increasing epoch counts. To further investigate the performance of various optimizers, we incorporate constant values for the learning rate, momentum, epoch and batch size into the optimisation process.

List Of Figures

List Of Table

1 Introduction

For a long time, stock price forecasting has been a significant academic topic. There are formal proposals that show proper modelling and designing of appropriate variables may lead to models with which stock prices and stock price movement patterns can be very precisely anticipated by advocates of the efficient market hypothesis. Using modern data mining techniques, researchers have also focused on technical analysis of stocks to detect patterns in stock price movements. For stock price prediction, we offer a hybrid modelling approach that combines machine learning and deep neural networks. We used the stock prices of Apple, Microsoft, Tesla, Alphabet, and Amazon from March 2017 to March 2021 for our research. By creating deep learning-based regression models employing long- and short-term memory (LSTM) networks with a novel walk-forward validation approach, we enhance the predictive power of our forecasting system. Optimization of LSTM hyperparameters using grid-searching ensures stability of validation losses with increasing epoch counts and convergence on validation accuracy for the LSTM models.

1.1 Aim & Objectives

To do a comparative study and data analysis to understand the behaviour and patterns in the trends. Prediction of one week of stock prices using ARIMA, LSTM and other models each with a different architecture and optimizer. To discuss a wide range of measures, risk and their associated findings for each of the regression models to be discussed in detail.

Keywords: Stock Price Prediction, Regression, Long and Short-Term Memory Network, Walk-Forward Validation, Multivariate Time Series.

2 Literature Review

For a long time, scholars have been interested in making predictions about stock price movement in the future. Formal propositions have shown it is feasible to successfully anticipate stock prices, even if the efficient market hypothesis is held by those who believe that it is impossible to effectively predict stock prices and stock price movement patterns. Using the decomposition of time series, Sen and Datta Chaudhuri **[1,3]** developed a new approach to stock price prediction. Using significant capabilities of machine learning and deep learning models, Sen has also offered a granular approach to stock price prediction in a short-term outlook. Machine learning methods like as regressions and classifications have been used to construct a very reliable framework for predicting stock price movements in the future. Accuracy in NIFTY forecasts was achieved by the authors using a non-linear multivariate regression model based on self-organizing fuzzy neural networks (SOFNN). Convolutional neural network (CNN)based models for forecasting multivariate financial time series data were reported.. The purpose of technical analysis of stock prices is to identify patterns in stock movements that lead to profit for investors.[4][6] In the literature, a number of economic and stock price-related indicators have been proposed for this purpose. A few of these indicators include Bollinger Bands, moving average convergence divergence (MACD) RSI, moving average, and momentum stochastic (MS) (MSW). An significant signal for stock market investing isn't just these well-known patterns in stock price movements; it's also the "head shoulders," "triangle," "flag," "Fibonacci fan," and "Andrew's pitchfork." Potential investors can get a clear picture of their investments using these methods. Predictions of the NIFTY 50 stock price movement on the NSE of India are being made using a variety of machine learning and deep learning-based predictive algorithms. **[7,8]**

The training dataset consisted of the NIFTY 50 historical index values from December 29, 2014 to December 28, 2018. For the test period, which lasted from December 31, 2018, to July 31, 2020, the open values of the NIFTY 50 index were forecasted based on the predictive models created from the training data-set. Deep learning-based long and short term memory (LSTM) network is incorporated into the predictive framework to further boost the prediction potential of the models. In this project, four LSTM models were created. Input data for the various models has a variety of different designs and structures. In contrast to three univariate models of LSTM, one model is multivariate. From a purely statistical standpoint, three models used data from the past two weeks to anticipate NIFTY 50 open values for the next week, whereas one model simply used data from the previous week. **[1,7,8]**

Current research on time series forecasting and stock price prediction can be divided into three broad groups based on the variables used and the approach taken when attempting a model of the problem. 1) Bi-variate or multivariate regression models using cross-sectional data make up the bulk of the first group's output. In most cases, these models fail to generate extremely accurate results because of their intrinsic simplicity and the invalidity of their linearity assumptions. Linear regression, for example, will produce an incorrect estimate of variance when observations are correlated (as in time series data). Linear regression can, of course, be fitted, but the inference and estimated prediction error will be anti-conservative. 2) Granger Causality Test, Auto-regressive integrated moving average (ARIMA), Granger causality, ARDL Auto-regressive distributed lag and VAR quantile regression are some of the approaches used to anticipate stock prices in the second category. 3) The third type of work comprises learning-based proposals based on machine learning, deep learning, and natural language understanding. Stock price prediction literature has a severe flaw in its capacity to reliably predict highly dynamic and rapidly changing patterns in stock price movement, except from the category of work that use learning-based methodologies. As a first step, we use machine learning and deep learning models to develop a framework for stock index forecasting that is both robust and accurate. We employed a deep learning model based on an LSTM (long-and short-term memory) network and investigated how well it predicted future stock index values. Table 1 shows all the relevant literature and paper published on this topic.

Table 1 – Key findings

Reference	Dataset	Prediction Technique	Key Findings
Improved Analysis of Stock Market Prediction **[1]**, 2021	Stock market data for Amazon	ARIMA-LSTM-SMP	There is a lack of effective device to undertake stock market analysis. This is also possible since stock market prices fluctuate. Due to this volatile nature, it is difficult to model the stock market effectively. Proposed experimental analysis aimed to help stockbrokers and investors profit from the stock market. As a result of the stock market's vivacious activity, the calculation indicates a vital place in stock market business. The stock price estimations are based on Linear regression, Moving Average, Auto Arima, and LSTM. Examined various strategies based on RMSE and found that LSTM outperforms other methods.
A Deep Neural Network Based Model For Jane Street Market Prediction **[2]**, 2021	Global stock exchange - Jane street data	Neural Network-Based Prediction Model	Apply standard machine learning methods with the same evaluation indicators and data sets. The deep learning model we used had the highest rate of return.
Application of Deep Learning in Stock Market Valuation Index Forecasting **[3]**, 2020	China's Growth Enterprise Market (GEM) from October 1, 2009 to June 28, 2019,	LSTM model in deep learning to learn and forecast the stock market valuation indicator, price-earnings ratio (P/E ratio). Then the prediction bias is measured by forecast trend accuracy (FTA), average forecast deviation rate (AFDR), and root mean square error (RMSE).	Deep learning network algorithms can accurately forecast financial time series, which is useful for stock market research. Paradoxically, this article advances deep learning research in stock market predictions and serves as a valuable resource for investors.
A Comparison of ARIMA and LSTM in Forecasting Time Series **[4]**, 2018	Nikkei 225 index (N225), NASDAQ composite index (IXIC), Hang Seng Index (HIS), S&P 500 commodity price index (GSPC)	ARIMA and LSTM models	The RMSE values clearly indicate that LSTM-based models outperform ARIMA-based models with a high margin (i.e., between 84% −87% reduction in error rates).
Stock price forecast with deep learning **[5]**, 2021	S&P 500 index	RNN - fixed activation function ReLU in all models but experiment with 3 different	The numerical experiments reveal that a single layer recurrent neural network with RMSprop optimizer produces optimal

		optimizers: SGD, RMRprop, and Adam.	results with validation and test MAE of 0.0150 and 0.0148 respectively.
A Review of Stock Market Prediction Using Neural Networks [6], 2019		Comparative study - Deep Learning Neural Networks	The model, when run with only text processing gave moderately accurate results. But when used in conjunction with LSTM, the model accuracy vastly increased. Predicting with financial news however, resulted in a higher accuracy than using a neural network with OHLC parameters. Recommends choosing financial news as the source rather than a forum since financial news is more factual and has a higher degree of accuracy.
Stock Trend Prediction by Fusing Prices and Indices with LSTM Neural Networks [7], 2021	Standard & Poor’s 100, New York Stock Exchange (NYSE) and NASDAQ-100 Technology Sector Index	RNN- LSTM	The mean accuracy achieved for the proposed feature set for the Apple stock is 84.66% while the values of the standard deviation and variance are concentrated around this value. The trained model in combination with our proposed set shows accuracy more than 60% for all the stocks that were taken into consideration. From the same table we can conclude that for all of the stocks examined in these experiments, the accuracy of the proposed model is better when the input feature set consists of stock price data and the selected stock market indices.
An Effective Time Series Analysis for Equity Market Prediction Using Deep Learning Model [8], 2019	NIFTY 50 index and INFYOSYS Ltd historical data	RNN- LSTM	RMSprop optimizer for 60 time steps and 1 LSTM layer through sensitivity analysis. It gives the minimum loss and influences the output variable obtained. Similarly the result has been produced for RMSprop optimizer with different time steps and LSTM's values. From the results obtained, it is clear evident that the deep learning models are capable of forecasting the NSE stock market. Optimum results were obtained for datasets with small values but failed to provide results for datasets having large values.
Prediction of Stock Based on Convolution Neural Network [9], 2020	CSMAR 3 and Wind4. The sample time is from July 1, 2008 to September 30, 2016	CNN and GBoost	This paper proposes a gradient lifting framework based on convolutional neural network to learn financial time series. The framework uses a combination of depth (CNN) and width (Gboost) to predict the stock price trend the next day. This method not only reduces the noise, but also gets a linear relationship between the prediction results and the trend, which makes the prediction more accurate. Moreover, the superiority of CGboost 6 prediction is greatly improved by adjusting the parameter of CGboost to 6.
Predicting Stock Market Price: A	1 year from Dhaka Stock	Long Short Term Memory, Extreme	Experimental results confirm that these models are capable of learning patterns for

Logical Strategy using Deep Learning [10], 2021	Exchange (DSE)	Gradient Boosting (XGBoost), Linear Regression, Moving Average, and Last Value model	time series data. LSTM performed best for our purpose. Practitioners will be able to apply our best found-model for forecasting stocks within DSE. However, for future work, we plan to work with a longer forecast horizon such as 3 month to 1 year. Furthermore, we will explore the possibility of improving the parameters with the use of alternative deep learning algorithms such as RNN, CNN, Deep Belief Network, and Auto encoder etc.
Forecasting with Deep Learning: S&P 500 index [11], 2021	S&P 500 index	Convolution-based neural network	The key insight of our model is the use of convolutional layer consisting of four filters of size 3. As a result, the data for each day in the input layer is considered in the context of its preceding and following days. The application of the filters leads to more informative features in the hidden layer. The results of numerical experiments against 7 benchmark models demonstrate the efficacy of the proposed model. Our model achieves the top accuracy rate of 56.21%.
Sentiment-aware stock market prediction: A deep learning method [12], 2017	18 million posts from constituent stocks of CSI300 index with open-values, closing-values and volume of daily CSI300 index from Wind database.	Long Short-Term Memory (LSTM) neural network and incorporates investor sentiment and market factors to improve forecasting performance.	Deploy a Naïve Bayes sentiment classifier to assign all posts on stock forums to three classes: positive, negative, and neutral. And then generate sentiment time series for subsequent work. Finally, develop a deep neural network model which consists of a Long Short-Term Memory layer, a merge layer, a ReLU linear layer and a SoftMax layer. Trained on 90% of the entire data set, this model gives a prediction accuracy of 87.86% in the rest 10% of testing data, outperforming other input permutations and SVM method by at least 6%. This work shows the potential of deep learning financial time series in the presence of strong noises.

3 Research Design & Methodology

For the purpose of data exploration, it is not necessary to make presumptions about the data before doing the investigation. Traditional hypothesis testing contrasts with "exploratory data analysis," which is a term used to describe data exploration. (key findings are summarised in Figure 2). We divided the experiment into two parts, the first of which was a deeper dive into data exploration and the second of which was a start at modelling. (Refer to Figure 1 for more information.)

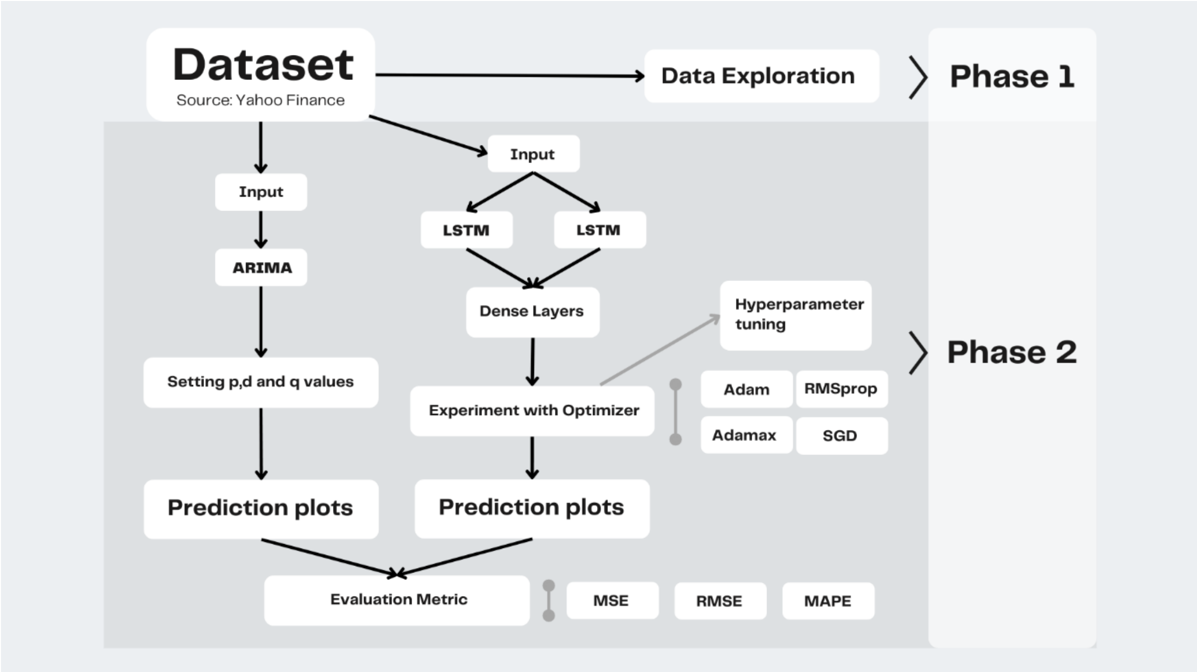


Figure 1: Research Design

The goal of data exploration is to discover patterns in a dataset and determine whether or not the data is cynical. Dickey-fuller tests were used to determine whether or not the data was stationary. The statistical parameters of a stationary time series (e.g., mean and variance) are not affected by time. The status of a stationary time series, then, is known as stationarity. Statistical qualities of a time series that change over time are said to be nonstationary, on the other hand. When attempting to model non-stationary data, estimates of the mean (and often the variance) will change. This allows us to discover the objects that are responsible for it. Other times, it's a slight inconvenience caused by the whims of fate. Since our data is non-stationary. We must proceed with pre-processing before modelling.

We began modelling after completing our data exploration phase. Linear regression models are a subset of ARIMA models, which aim to estimate the future values of the target variable by using the previous observations. Exogenous variables are not considered in the basic version of ARIMA models, which is an important feature. Instead, the forecast is based solely on the target variable's historical values (or features crafted from those past values). As the name suggests, ARIMA stands for Autoregressive integrated moving average. The future can be predicted simply rearranging the notation.

```
Y_forward1 = B0 + B1*Y + B2*Y_lag1 + B3*Y_lag3 + ... Bn*Y_lag(n-1)
```

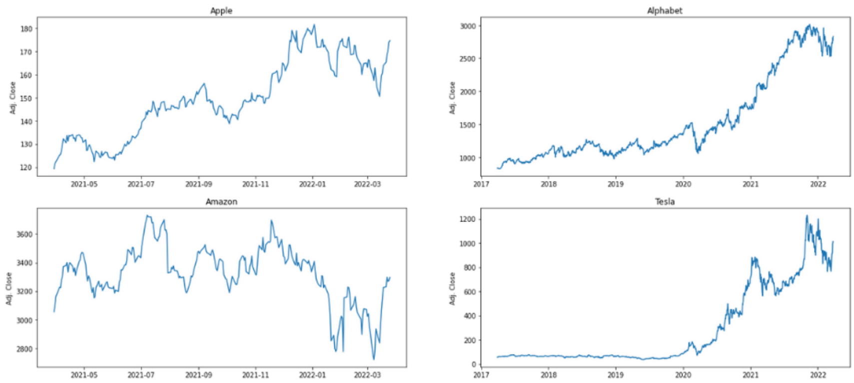
We're now using the present value and its prior lags to estimate the future value (1 time step ahead).
At first, the moving average was nothing more than the Y variable's trailing moving average (e.g. a 200 day moving average). A moving average model is summarized by the following equation:

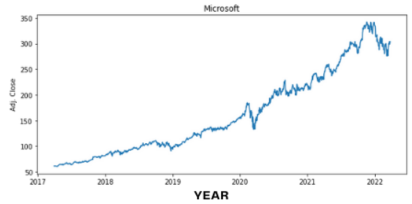
```
Y = B0 + B1*E_lag1 + B2*E_lag2 + ... + Bn*E_lagn
```

The formula for a moving average model is as follows: E is the measure of the discrepancy between our model's exact response and the closest approximation. As a result, MA models project Y based on their historical errors. Stats model's ARIMA function necessitates a minimum of two arguments: 1) A Pandas series of raw real stock values (we don't need to difference it in beforehand since the ARIMA algorithm will do it for us) is given to the algorithm. 2) Order specifies the ARIMA function how many components of each model type to consider in the following order — one for each model (AR lags, time steps between difference, MA lags). Following these procedures, we built a forecasting model by manually putting in p,d, and q data into the model. A single-layer LSTM model did not yield better results. As a result, we devised a two-layer LSTM algorithm. Its compositional character enables us to construct our models in this manner. In order to stack numerous layers, you only need to arrange them so that layer 'I' receives input from layer i-1 and outputs from layer i+1. As a result, it creates a great hierarchical level of features, a multi-layer design often performs better because high-level features simply recycle low-level features by combining them more powerfully and descriptively.

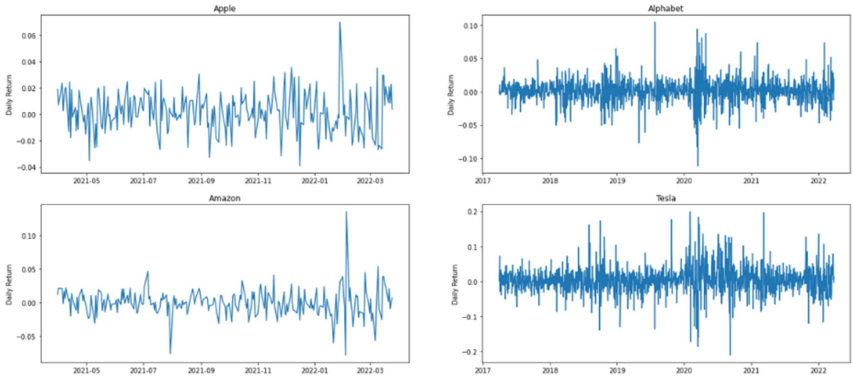
PHASE 1

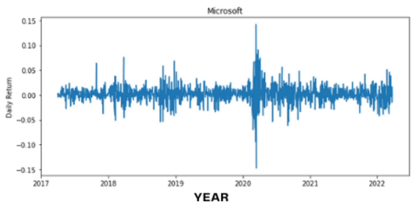
DATA EXPLORATION



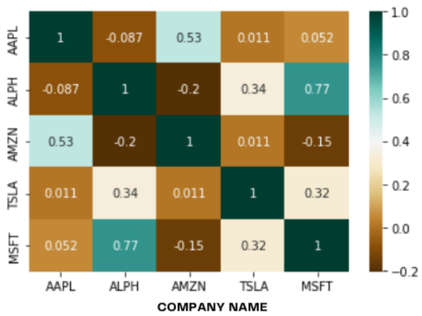


HISTORICAL TREND OF THE ADJ CLOSE PRICES VS YEAR FOR ALL DATASET





DAILY RETURN OF ADJ CLOSING PRICES ON AVERAGE FOR ALL DATASET

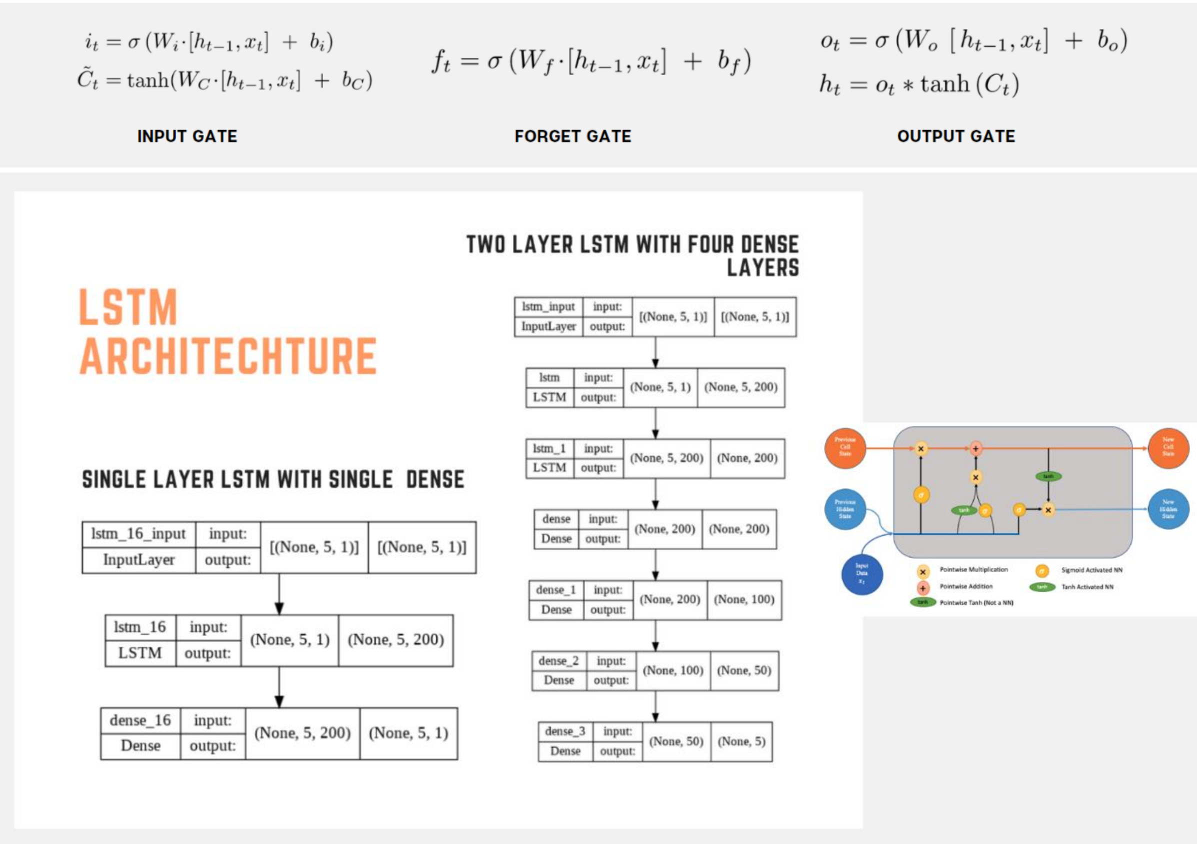


CORRELATION BETWEEN DIFFERENT DATASETS ON ADJ CLOSING PRICES

Figure 2 - Data Exploration

LSTMs differ from more standard feedforward neural networks in that they feature feedback connections. As a result of this, LSTMs are able to process extensive sequences of data (e.g. time series) without treating each point in the sequence individually, but rather, preserving important knowledge about prior data in the sequence to aid in the processing of incoming data points. (Refer figure 2 for model architecture) At any given point in time, the output of an LSTM can be influenced by three factors: Known as cell state, the network's present long-term memory Prior to this moment in time, known as the previous hidden state. The current time step's input data. To govern the flow of data into, through, and out of an LSTM network, a number of 'gates' are employed. The forget gate, input gate, and output gate are all components of a conventional LSTM. Each of these gates is a neural network in and of itself.

Figure 3: LSTM working and architecture



Discover which input value should be utilised to change the memory by using the "input gate". This decision is made by the sigmoid function. and the tanh function assigns a weight to the values provided, ranging from -1 to 1, based on their significance. "Forget gate" — find out what information in the block should be discarded. The sigmoid function determines this. Input data from Xt and the preceding state (ht-1) are analysed, and for each value in the cell state Ct1, a value between 0 and 1 is output (omitted) or kept (kept). "Output gate" — the block's input and memory are utilised to determine the output. To pick which values to pass between 0 and 1, the sigmoid function uses a weighting scale from -1 to 1, which is then multiplied by the result of the sigmoid function. We have used two LSTM layers with five dense layers with an input shape of (5,1) for our experiment.

4 Results & Discussion

We downloaded 5 years of stock market data (25-March-2017 to 25-March-2022) for five companies, namely: Apple, Microsoft, Tesla, Amazon and Alphabet (Google). After preprocessing and visualization, we found that Tesla has the highest projected returns and the biggest risk factor after a deeper dive into the exploration. [18] On the other hand, the safest investments are Alphabet and Microsoft. (refer figure 1,4) The correlation between two stocks' price fluctuations is referred to as stock correlation. Other asset classes, like as bonds or real estate, can also be considered in the context of this term. Investors must be willing to consider risk in a variety of contexts and with a degree of pliability. As an example, risk management calls for diversification. Holding a portfolio of low-risk investments with the same level of risk can be quite hazardous. Investors can use the information gleaned from the data exploration to make more informed decisions. In statistics, kernel density estimation (KDE) is a non-parametric way to estimate the probability density function of a random variable. Kernel density estimation is a fundamental data smoothing problem where inferences about the population are made, based on a finite data sample. We have plotted KDE distribution plots individually for each the 5 companies mentioned above. (Refer figure 5).

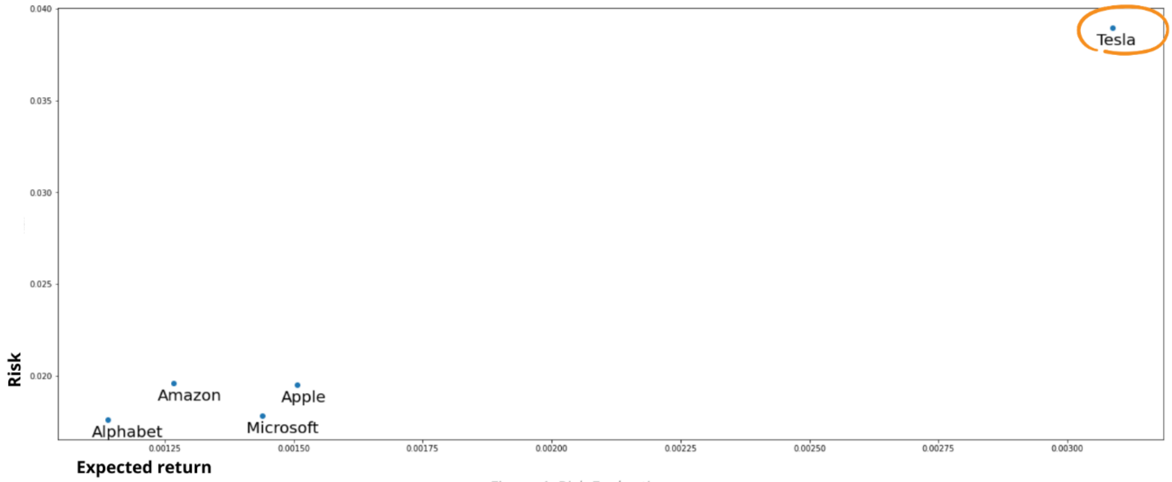
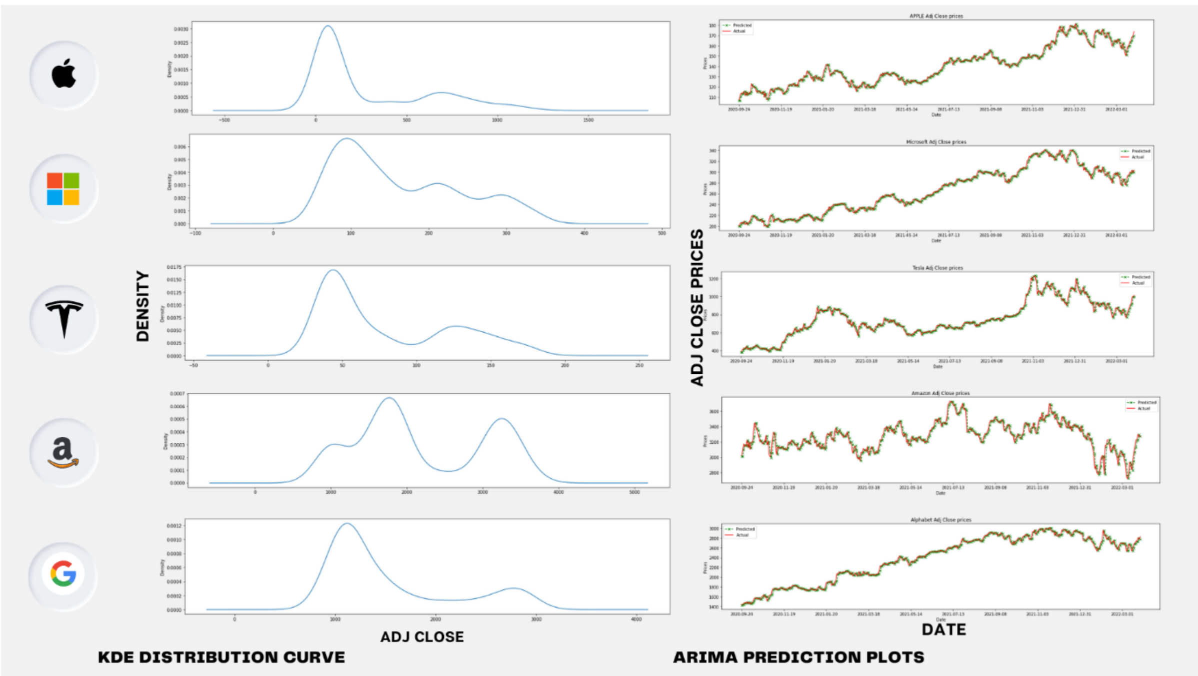


Figure 4: Risk Evaluation

Expected return = (return A x probability A) + (return B x probability B)

Plot for Risk: An expected return and a standard deviation are two statistical measures that investors can use to analyze their portfolios. The expected return is the anticipated amount of returns that a portfolio may generate, whereas the standard deviation of a portfolio measures the amount that the returns deviate from its mean. as we can see here, Tesla has highest returns to high risk factor whereas alphabet and Microsoft have lowers returns and lower risk factor.

Figure 5: KDE Distribution and Arima prediction plots



In phase two, following the exploration of the data, we compared LSTM and ARIMA models. The primary goal for us (In LSTM) was to compare the effectiveness of various optimizers in LSTM models. (Refer figure 6). When tested for all stock datasets, our baseline model ARIMA performed better, showing the least Mean Absolute Percentage Error, which can be defined as the difference between the actual or observed value and the forecasted values. When it comes to predicting stock price for 'Adj Close', ARIMA models can be very accurate and reliable. Plots and results are shown in figure 5. In LSTM, we maintained the same epoch, learning rate, momentum, and batch size throughout the entire testing process. These hyperparameter tunings produced results that were superior to those of the baseline LSTM model. However, in our experiment, the ARIMA model showed the best results, and the best performing optimizer for LSTM was found to be Adam and Adamax. Adam is derived from RMSProp, but it uses an estimate of the gradient to determine the momentum parameter in order to increase the speed of training. According to the findings of the experiments presented in [10], Adam was superior to all other training approaches in a variety of different settings. In the end, we tried our hand at plotting prediction graphs for each separate dataset. (Refer figure 7) At the end of the day, we tested different prediction values for future stock values using the actual value that was available on Yahoo Finance on that particular day. Figure 7 demonstrates the forecast for the next five days.

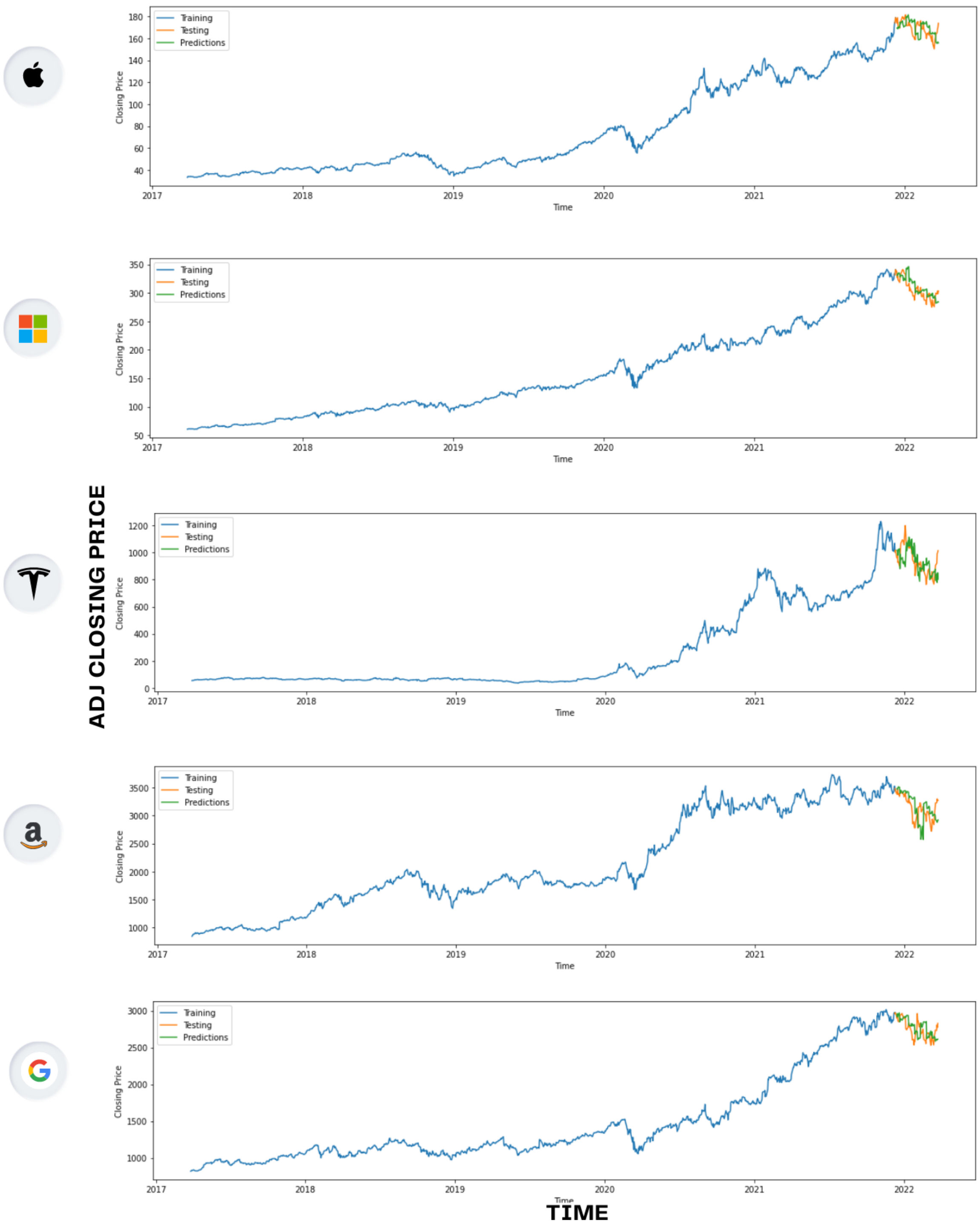
Figure 6: Results using different optimizer computed against evaluation metric for all stock dataset

OPTIMIZER	Adam()	Adamax()	RMSprop()	Adam()	Adamax()	RMSprop()	
MICROSOFT - FOR EPOCH - 800, BATCH_SIZE = 32				AMAZON - FOR EPOCH - 3000, BATCH_SIZE = 32			
	132.5	108.2	113	33518	30566.7	17683	MSE
	11.5	10.40	10.63	183.0	174.8	132.9	RMSE
	0.031	***0.028***	0.029	0.043	0.041	***0.034***	MAPE

OPTIMIZER	Adam()	Adamax()	RMSprop()	
APLHABET - FOR EPOCH - 3000, BATCH_SIZE = 32				
	9016.7	8156.5	8167.6	MSE
	81.6	78.6	90.3	RMSE
	0.023	***0.021***	0.024	MAPE

OPTIMIZER	Adam()	Adamax()	RMSprop()	Adam()	Adamax()	RMSprop()	
APPLE - FOR EPOCH - 500, BATCH_SIZE = 32				TESLA - FOR EPOCH - 3000, BATCH_SIZE = 32			
	44.09	37.83	46.1	6027.4	6740.17	5715.6	MSE
	6.7	6.15	6.9	77.63	82.09	75.6	RMSE
	0.033	***0.029***	0.034	***0.064***	0.069	0.066	MAPE

Figure 7: LSTM Prediction plots for all stock dataset. Future values are also predicted (upto 5 days)



March/2022 -->	28th	29th	30th	31st	Apr 1st
Actual 'Adj Close' from yahoo finance website on that day are -->	2830.43	2839.00	2765.00	2792.99	2814.00

```
Predicted Values are --> [[2822.1938 2825.5334 2818.178  2823.2153 2825.898 ]]
```

5 Threats to Validity

The financial analysis examines the company's current stock prices from a macro perspective to determine how accurately they reflect the company's potential to generate future revenues. When someone attempts to forecast what will happen in the market, they are attempting to forecast something that is dependent on a wide variety of factors that are beyond their control. The price of any stock reflects not only the value of the stock and its potential earnings in the future, but also the sentiments of the market. It is difficult to put one's finger on the pulse of the market because the market is nothing more than the accumulation of millions of different opinions. Admitting that one cannot predict the market with any degree of reasonable accuracy is one of the fundamental steps towards achieving success in trading and investing. One strategy that can be utilised is to make estimates of value and then to wait for the price to approach those estimates. That has the upper hand. It is impossible to take into account all of the factors that have an impact on stock prices. It is really impossible to build a precise model that would rely on all of those factors, and one of the primary reasons for this is that the majority of the factors are not known in advance: even if some events that affect the stock market have happened in the past, you never know what else would happen in the future. It is impossible to build a precise model that would rely on all of those factors. If it were possible to forecast the movements of stock markets with a high degree of accuracy, and if others were also able to do so, then the very fact that stock markets could be forecasted would influence their movements. ARIMA models are to be faulted for their performance in forecasting then in my opinion a strong case can be made about them doing poorly in long-term forecasting only. They could also be faulted for assuming that the error terms is white noise with a constant variance, which translates into a constant prediction error. To train RNNs and LSTMs, memory-bound computation is required, which makes it difficult. For LSTM to run at and for each sequence time-step, each cell must have four linear layers (MLP layers). Because the system does not have enough memory bandwidth to feed the computational units, it is impossible to use many compute units at once when computing linear layers. Adding computational units is simple, but adding memory bandwidth is more difficult.

6 Conclusion and Future work

In this experiment we have attempted to show the capabilities of optimizers and models in Deep learning. Before doing that finding patterns in a dataset and determining whether or not the data is meaningful constitute data exploration. We used statistical tests like the dickey-fuller test to determine whether or not the data was stationary or not. Statistics (such as the mean and variance) of stationary time series are not affected by time. The test showed that our dataset shows some cynical trend and is non-stationary. Following this, we plotted a KDE plot, which depicted the distribution of data points. The correlation graph demonstrates that Tesla has the best returns with the largest risk factor, whereas alphabet and Microsoft have the lowest returns and the lowest risk factor, respectively.

In terms of stock forecasting, ARIMA models can be extremely accurate and reliable. Cononsidering ARIMA as our base model with p, d, and q, where integer values substitute for the parameters to indicate the type of ARIMA model used. The parameters can be defined as: Here we have used a custom parameter input(4,1,0) respectively. The main drawback is that it takes a long time to process the data and a lot of iteration to manually feed values before we get the best results. For all the dataset, we got MAPE under 2% of deviation from the actual value. Whereas, Tesla's stock price has soared in the last few years, therefore this shift in the time series is likely. As a result, ARIMA delivered its results on tesls' stock at a MAPE value of 2.7 percent. The utilisation of LSTMs, which are techniques that are extensively employed for solving sequential modelling problems, can be seen in time series forecasting and language modelling. Convergence was difficult to achieve when only a single LSTM layer was used, and the results were unsatisfactory. When two LSTM layers are piled on top of each other, the model's level of detail and accuracy improves. In order to assist the model in going further into its learning, we made use of five dense layers. When utilising the rescaled gradient, Adam updates directly estimate the first and second gradient moments. On the other hand, the RMSProp makes use of momentum in order to construct parameter updates. The performance of Adam and Adamax is superior than that of other optimizers. When we compared the results of our evaluation model to those obtained by ARIMA, we were unable to obtain better results. ARIMA was most suitable model for our experiment. More LSTM layers can be added to the model to increase its density, and hyperparameter adjustment can be done to improve outcomes.

7 Reference

[1] A. Ashok and C. Prathibhamol, “Improved Analysis of Stock Market Prediction: (ARIMA-LSTM-SMP),” IEEE, 2017.

[2] S. Guo, H. Wang, L. Jiahao and X. Chen, “A Deep Neural Network Based Model For Jane Street Market Prediction,,” IEEE, 2021.

[3] G. L. M. Xiao and Y. Guo, “Application of Deep Learning in Stock Market Valuation Index Forecasting,” IEEE, 2020 .

[4] F. Kamalov, L. Smail and I. Gurrib, “Stock price forecast with deep learning,” IEEE, 2021.

[5] S. J. a. D. Chaudhuri, “An Alternative Framework for Time Series Decomposition and Forecasting and its Relevance for Portfolio Choice - A Comparative Study of the Indian Consumer Durable and Small Cap Sector. Journal of Economics Library,” IEEE, 2020.

[6] J. a. D. C. Sen, “A Time Series Analysis-Based Forecasting Framework for the Indian Healthcare Sector. Journal of Insurance and Financial Management,” IEEE, 2017.

[7] S. J. a. D. Chaudhuri, “A Predictive Analysis of the Indian FMCG Sector Using Time Series Decomposition-Based Approach. Journal of Economics Library,” IEEE, 2017.

[8] A. A. O. a. A. C. K. A. Adebiyi, “Stock Price Prediction Using the ARIMA Model. In: Proc. of the Int. Conf. on Computer Modelling and Simulation, Cambridge, UK,” IEEE, 2014.

[9] S. Mishra, “The Quantile Regression Approach to Analysis of Dynamic Interaction between Exchange Rate and Stock Returns in Emerging Markets: Case of BRIC Nations. IUP Journal of Financial Risk Management,” IEEE, 2016.

[10] A. Menon, S. Singh and H. Parekh, “A Review of Stock Market Prediction Using Neural Networks,” IEEE, 2016.

[11] V. Karlis, K. Lepenioti, A. Bousdekis and G. Mentzas, “Stock Trend Prediction by Fusing Prices and Indices with LSTM Neural Networks,” IEEE, 2021.

[12] A. Hasan, O. Kalıpsız and S. Akyokuş, “Predicting financial market in big data: Deep learning,” IEEE, 2017.

[13] A. Sachdeva, G. Jethwani, C. Manjunath, M. Balamurugan and A. V. N. Krishna, “An Effective Time Series Analysis for Equity Market Prediction Using Deep Learning Model,” IEEE, 2019.

[14] R. Zhang, Z. Wu and S. Wang, “Prediction of Stock Based on Convolution Neural Network,” IEEE, 2020.

[15] A. Shome, M. A. Islam, A. J. Nova and S. Ahmed, “Predicting Stock Market Price: A Logical Strategy using Deep Learning, Milon Biswas,” IEEE, 2021.

[16] F. Kamalov, L. Smail and I. Gurrib, “Forecasting with Deep Learning: S&P 500 index,” IEEE, 2020.

[17] J. Li, H. Bu and J. Wu, “Sentiment-aware stock market prediction: A deep learning method,” IEEE, 2017.