THESIS

# STOCK MARKET PREDICTIONS USING MACHINE LEARNING

Submitted by

Hari Kiran Sai Surayagari

Department of Computer Science

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Spring 2021

Master's Committee:

Advisor: Charles Anderson

Asa Ben-Hur
Christopher Stein

# ABSTRACT

## STOCK MARKET PREDICTIONS USING MACHINE LEARNING

In this thesis, an attempt is made to try and establish the impact of news articles and correlated stocks on any one stock. Stock prices are dependent on many factors, some of which are common for most stocks, and some are specific to a type of company. For instance, a product-based company's stocks are dependent on the sales and profit, while a research-based company's stocks are based on the progress made in their research over a specified time period. The main idea behind this thesis is that using news articles, we can potentially estimate how much each of these factors can impact the stock prices and how much of it is based on common factors like momentum.

This thesis is split into three parts. The first part is finding the correlated stocks for a selected stock ticker. Correlated stocks can have a significant impact on stock prices; having a diverse portfolio of non-correlated stocks is very important for a stock trader, and yet very little research has been done on this part from a computer science point of view. The second part is to use Long-Short Term Memory on a pre-compiled list of news articles for the selected stock ticker; this enables us to understand which articles might have some influence on the stock prices. The third part is to combine the two and compare the result to stock predictions made using the deep neural network on the stock prices during the same period. The selected companies for the experiment are - Microsoft, Google, Netflix, Apple, Nvidia, AMD, Amazon. The companies were selected based on their popularity on the Internet, which makes it easier to get more articles on the companies.

If we look at the day to day movement in stock prices, a typical regression approach can give reasonably accurate results on stock prices, but where this method fails is in predicting the significant changes in prices that are not based on trends or momentum. For instance, if a company releases a faulty product but the hype for the product is high prior to the release, the trends would show a positive direction for the stocks and a regression approach would most likely not predict the fall in the prices right after the news of the fault is made public. It will eventually correct itself, but it would not be instantaneous. Using a news-based approach, it is possible to predict the fall in stocks before the change is noticed in the actual stock price. This approach seems to show success to a varying degree with Microsoft showing the best accuracy of 91.46%, and AMD had the lowest at 40.59% on the test dataset. This was probably because of the volatility of AMD's stock prices, and this volatility could be caused by factors other than the news such as the impact of some other third party companies.

While the news articles can help predict specific stock movements, we still need a trend based regression approach for the day to day stock movements. The second part of the thesis is focused on this part of the stock predictions. It incorporates the results from these news articles into another neural network to predict the actual stock prices of each of the companies. The second neural network takes the percentage change in stock price from one day to the next as the input along with the predicted values from the news articles to predict the value of the stock for the next day. This approach seems to produce mixed results. AMD's predicted values seem to be worse when incorporated with only the news articles.

# ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Charles Anderson for his support during my thesis. It was a pleasure learning under his guidance. His support was essential during the many times I ran into troubles and delays.

I would also like to express my gratitude to my parents for their continuous encouragement during my years of study and my thesis work. I would not have been able to finish my work without them.

TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# Chapter 1

# Introduction

## 1.1 Motivation

Stocks are very volatile; this complex nature of stock prices is a significant attraction for researcher and statisticians to find a way to predict them. Despite the numerous amount of research publications in this field, there are still many that claim that stock markets cannot be predicted. This is primarily because of the number of factors that affect stocks prices and those factors themselves depend on some other, potentially unknown factors.

The more commonly used approach for stock market predictions is to use past experiences in price changes to predict future change in prices. “Financial forecasting is an example of a signal processing problem which is challenging due to small sample sizes, high noise, non-stationarity, and non-linear", according to Lee Giles [1]. With stocks, the data is typically the stock values from real-time transactions and hence getting a large sample will depend on a more extended time period. A longer period of time does not always give the right result as the financial markets are not stable at such intervals. However, if the factors that impact stock prices can be understood, then it could be possible to predict the prices without relying on historical prices so much. Understanding all the factors is nearly impossible, so a potential solution to that problem is discussed in this thesis.

Given the complex nature of stocks, machine learning is probably the best approach for this application. There are many research papers using some form of machine learning, such as feed-forward neural networks [2], SVM [3], and recurrent neural networks [4] for stock market predictions. More recently [5,6] used neural networks to predict the closing price for the next day. [5] did a comparative analysis between a simple artificial neural networks and random forest for various companies with the neural network edging out the random forest performance. [6] on the other hand did a comparative analysis between Recurrent Neural Network, Long Short-Term Memory, Convolutional Neural Network and Multi Layer Perceptron. LSTM seem to perform better in some cases and CNN seem to perform better in some other situations. Both these papers, however, are using the actual stock prices for their work.

This thesis consists of 2 parts - one is to learn context and impact of news articles on stock price which is done by using Long-Short Term Memory and the second part is to learn how general trends along with the news predicted stock movement could be used together to make an accurate prediction of the stock values which is done using a deep network. Applying Recurrent Neural Networks, more specifically Long-Short Term Memory, on news articles related to a company, it may be possible to predict which articles directly impact the company's stock price assuming enough people read the same or similar news. Using news articles for stock market predictions [7–10] is not a new field of research, but this thesis offers slightly different and potentially better results. Most published papers in this field tend to apply sentiment analysis to the news articles to predict the change in stock prices. This approach partly works, but there could be instances where sentiment analysis may not work. Sentiment analysis works in situations where the impact of the news is felt instantly in terms of the change in stock value. Some conditions are not discussed in the above papers such as knowing which articles are to be selected for the predictions. If one was to use sentiment analysis then automatically articles that do not show either positive or negative sentiment are ignored, but that does not automatically imply that such articles have no impact on the stock value. There is also the situation where sometimes the article has a negative sentiment, but its impact on the stock value is positive. This can happen if the purpose of the article is not related to the stock value. All the news from all articles gives a large enough dataset for a machine learning algorithm, but not all news is relevant to the stocks. If it is possible to predict which articles are relevant to the news then using those articles to predict the stock prices may result in a better accuracy of the predicted stock values. A couple of papers [11, 12] seem to be applying the stock price movement to measure the importance of a news article. They are looking at the direction of the stock movement so their 2 classes are "up" and "down". Both are also looking at different sectors rather than specific companies. [11]is using Support Vector Machine and K-nearest Neighbors, while [12] is proposing a new approach using Multiple Kernal Learning. [11]

showed some positive results with this approach. They obtainined an accuracy ranging from 65% to 81% depending on the type of kernel they were using and the data source. This shows that the approach used in this thesis is viable.

Another critical factor for stock prices are the correlated stocks. In modern markets, all stocks are related by some variable factor. Correlation does not imply causation; however, if any two stocks are related, then it is possible that there may be a causation between the two stocks especially if the Pearson's correlation coefficient [13] is very high. Despite the importance of correlation for stock traders and investment companies, there is not much research on this by computer scientists. This thesis will attempt to address if the correlated stocks have an impact on any one company's stock prices and will aim to find the correlation factor above for which there is an impact on the predictions.

The final part of the thesis is to merge the above two solutions to get a more accurate stock market prediction. This is verified by comparing the prediction accuracy with a simple deep network on the same data without the correlated stocks or the LSTM data.

## 1.2 Goals and Research Questions

The primary purpose of this thesis is to try to predict the change in stock prices. It is abundantly clear that stock prices are dependent on various factors. Some of these factors are discussed, and their impact both individually and together is evaluated in this thesis. To validate the overall goal of the thesis, four questions have been formed. Various experiments and their results are discussed, to help answer the research questions.

1. *Is it possible to classify news articles based on the daily change in stock prices using Long-Short Term Memory?*

Using LSTM, it may be possible to extract the features that have an impact on the stock price of a company. Many articles about some company are not relevant to that company's stock prices. If the relevant articles be can be extracted, then they can be used to predict the stock price movements.

2. *Do correlated stocks have an impact on a specific stock?*

Having a diverse portfolio [14] is vital for making a profit to a stock trader, implying that the stocks are related somehow. This correlation factor is uncertain but using machine learning it may be possible to predict the effect of correlated stocks.

3. *Does the market itself have an impact on a specific stock?*

In stock markets, terms like “market is down” or “market is up” indicate that a major market index, usually the Dow Jones or S&P 500 indices, are below or above their trading price at some point in the past. As mentioned above, the stocks are related by some factor. This typically applies to the market as a whole. When the market index is down the stocks in that market also face a downward trend. This is not always true as the trends of certain stocks is so strong that the market movement doesn’t impact the stock as much. Using machine learning, one can potentially predict the factor by which the market can impact a specific stock.

4. *Is it possible to get a more accurate stock price prediction by combining all of the above factors?*

All the factors mentioned above individually impact a specific stock, but it is unclear if the collective impact can be measured. In this thesis, an attempt is made to make a better prediction using the factors from above as compared to a simple prediction from a historical trend.

## 1.3 Outline

This thesis is divided into six chapters: Introduction, Stocks, Machine Learning, Methodology, Results, and Conclusion. The stocks and machine learning chapters explain the concepts behind this thesis. Stocks describes the terms used in stocks and their meaning. In this chapter factors that influence the stock prices are also discussed. In the Machine Learning chapter, as the name suggests, machine learning is explained as a topic, and more specifically the algorithms used in the thesis are discussed. The methodology chapter discusses the techniques and algorithms used in this thesis. This chapter talks about the various features of LSTM and deep neural networks, the reason why they were used. This chapter also talks about the values used for each of the features. Experiments are done on the above-discussed machine learning algorithms, and their results are discussed in the next chapter. The final chapter is the conclusion of the thesis. This chapter discusses the results of the experiments and their future implication on stock markets.

# Chapter 2

# Stocks

According to Investopedia [15], a stock is a type of security that signifies ownership in a corporation and represents a claim on the part of the corporation's assets and earnings. Stocks are also known as shares or equity. Owning shares in a company gives the owner the right to vote in shareholder meetings, receive dividends and the right to sell the shares. If one owns more shares, then they have more voting power and hence indirectly control the direction of a company. However, for a stock trader, the primary concern is the value of a stock.

Stock trading is done at stock exchanges where traders buy or sell stocks. In modern days, most trades are done through electronic communications instead of a physical trading floor. This allows the trades to be instantaneous. There are many stock exchanges around the world such as New York Stock Exchange (NYSE) and Shanghai Stock Exchange (SSE), National Association of Securities Dealers Automated Quotations (NASDAQ) and London Stock Exchange (XLON). Due to the nature of modern markets, many of the exchanges are linked electronically, and hence they can have an impact on each other's performance.

## 2.1 Buying and Selling Stocks

As mentioned above, for most stock traders the most crucial aspect of stocks is the ability to buy and sell to make profits. A company's stock price is initially set based on the company's value, revenues, and some other factors. After that its value changes based on the availability and demand of the stock. When a large portion of the available stock is purchased, then the stock's value goes up after that purchase. Similarly, if a large portion of the stock is sold and available, then its value goes down. Typically, it is essential for a trader to understand what a company's stock should be valued at. Having this information will allow the trader to buy those stocks if its present value is below what it should be and sell it if its value is above its actual value.

In stock markets, there is always an upward trend due to inflation and other forces because new money goes into the market every day. Hence making a profit does not automatically imply a good trading system; one can merely buy well fairing stocks, and over a long period of time, they tend to make a profit.

## 2.2 Types of Stock Trading

There are typically two types of trading techniques – long term and short term. As mentioned above, making a long-term investment in settled stocks will give profits over time. This approach requires less time and effort from the trader and profits over time tend to be more than from bonds and other investments.

In short-term trading, the trader aims to profit from small changes in stock prices. These changes can be over as little as a few seconds, called intra-day stock trading, or over a day to a few days, called swing trading. Short-term trading can be very labor intensive, especially when done manually and the trader must spend a lot of time and effort. The trader has to track the latest stock changes and news to earn a profit wherever possible.

For this thesis, the focus will be a 1-day swing trade approach. Since it is so labor intensive, having a machine learning algorithm predict the price movements can be beneficial for a trader.

## 2.3 Stock Predictability

It was mentioned earlier that some papers indicated that a prediction of stock prices is not possible given the complex nature of stocks. Theory of Random Walk is an idea that gets used a lot in stock markets. It suggests that stock price changes are independent of each other and have the same distribution, so historic prices or trends cannot be used to predict the future. Burton Malkiel [16] first introduced this theory in 1973. On the other hand, there are plenty of papers which show significant empirical signs that indicate stock prices are to some extent dependent on past behavior [17]. The exact prices may not be predictable, but the patterns emerge over time. Since there are patterns, then there is a possibility that through proper analysis and complex models, a better understanding of the patterns can be achieved. Technical traders have a significant influence on the flow of the market every day. Large finance companies use automated algorithms to make high-frequency trading to earn money, so it can be argued that there must be an algorithm that enables a trader to be successful in the market.

In this modern day and age, stock markets have a major role to play in the world economy. There is compelling evidence supporting both sides of the predictability argument, but with the emergence of deep learning in recent years, one can argue that complex patterns and complicated connections between variables can be discovered. One thing for certain, however, is that there is plenty of interest and belief in stock market predictions, which itself can be the driving factor for the predictability of stock prices.

## 2.4 Types of Stock Analysis

Stocks volatility is well-known and the concept is understood. This volatility is because of the many forces that impact the stock prices, and these numerous factors are the reason why predicting stock prices is so tricky. Despite the amount of research done and the numerous articles and thesis papers written on this subject, there is still no complete technique or algorithm for an accurate stock value prediction. Some papers indicate that getting an accurate stock price prediction is nearly impossible, as mentioned previously. Nevertheless, if we did try to predict the stock prices, some of the many factors that can impact stock prices are historical price trends, politics, psychological feelings towards a stock, macroeconomics and a company's financial state [18]. Each of those factors can be further classified. This shows the wide nature of the elements that drive the stock prices. It is just not feasible for one algorithm to be able to understand and evaluate the importance of each of those factors on a specific stock's value and present price. It is, however, possible to get a prediction within a specific error region by considering only some of the many factors that can impact the stock prices.

Given the many factors, people use different methods to analyze stock price movements. Broadly these are classified into two types: Fundamental Analysis and Technical Analysis.

### 2.4.1 Fundamental Analysis

In this approach, a company's fundamental value is calculated based on its many assets and product value. Again, there are many factors that can impact the company's value. Some of the more prominent ones are the following.

**Stock Trading Ratios:** These are the ratios that are used to understand the earning of the company with respect to its competitors, the higher the earning, the higher the value of the company. This can be further subdivided into Earnings Per Share (EPS) which is the amount earned by a shareholder per share, and the other is Price/Earnings (P/E ratio) which is the ratio between the company's stock price to its earnings.

**Market Size:** The size of the market and the company's share of that market is essential for its valuation. If a company provides a vital product and it is the only one that offers that product, then it is likely that its value is very high.

**Demand:** The demand for the product manufactured by a company can have an impact on its stock price. The demand can be dependent on season and economy.

**Management:** The company's management team plays a vital role in the direction of the company's future. A team of proven individuals has more impact on stock prices than a team of unproven individuals.

**Economic Climate:** This is a crucial factor in the stock prices of a company, If the economy is booming, then many companies perform well, and hence their stock prices are high. A prime example is the Financial crisis in 2008 during which a lot of the companies including banks lost their value.

Long-term investors typically use fundamental analysis. Most of the factors in the fundamental analysis do not change day to day, hence they are more useful in a buy and hold approach.

### 2.4.2 Technical Analysis

Technical Analysis in stock trading uses charts and various mathematical formulas to identify trends to predict the stock price movements. In this approach, the trader looks at the price alone without paying attention to the underlying fundamentals. Both approaches have their merits, and since a lot of the researchers do not appear to provide compelling evidence for either approach, anyone researching this topic can only assume that it is still possible and the reason it appears impossible is that it has not been done yet. John J.Murphy gave a comprehensive explanation of technical analysis in his book on financial markets [19].

There are a lot of hedge funds and investment companies that can predict the change in stock prices to make a profit most of the time. As mentioned previously, sometimes the stock prices depend on the macro-economical news. If one can consider the news and get ahead of the changing prices, then it is possible to make a profit. From the computer scientist point of view, technical analysis can be done using neural networks or any other machine learning method, but getting the fundamental analysis right is very difficult because a lot of that depends on understanding the individual situation. This thesis uses an approach that involves using both fundamental and technical analysis to try to get a good prediction of stock prices.

## 2.5 Data Collected

An essential part of any machine learning application is data. For stocks, there are various types of data. Since this thesis uses daily predictions, the data assembled is the Highs, Lows, Open and Close values for each day from Google finance [20]. “High" is the highest value of stock during the day, “Low" is the lowest value of the stock during the day, “Open" is the opening value of the stock for the day and “Close" is the closing values of the stock for the day. There are other terms like volume and adjusted close which are not considered in this thesis for convenience.

There is a large amount of other data that can, in theory, be very useful for stock price predictions from macro-economical data, such as unemployment rates and interests to currency exchange rate and raw material prices. There is also data regarding the economy of every nation in the world, which could also impact the stocks. The listing, collecting and using all available data which can potentially impact stock markets is an enormous task.

For this thesis, however, all the companies listed on NASDAQ, Dow Jones and S&P 500 are used which comes to a total of about 4000 companies. The index values for each of the markets is also downloaded for the last five years (2012-2017). The news articles related to the companies listed in the abstract, released in the last five years, have been used to establish a connection between the articles and their impact on stock prices. The websites from which the articles were downloaded include Marketwatch, Investopedia, Bloomberg, Reuters, TheStreet, and Investors. All this data is obtained using a third party called Event Registry [21]. A total of around 200,000 articles were collected with 25,000 - 30,000 per company. The idea is that any relevant news regarding a specific stock will be in the downloaded data and this can be predicted using Long Short-Term Memory and, hence, the final stock price can be more accurate.

While stock prices are dependent on various factors, it is wise to understand that there is a lot of noisy data in finances as well. This can be caused by the uncertainty about events that might occur in the future, such as technological breakthroughs and change in trends. It can be caused by delays in sales or purchases of stocks and expectations other than the rational rules. These imperfections can cause the market to function illogically and make the market predictions more difficult [22]. This is maybe some of the reasons that some believe that stock markets cannot be predicted.

# Chapter 3

# Machine Learning

Since the invention of computers people have been wondering if they can be made to learn. The impact this can have on humanity is endless. From medical diagnosis to space research, machine learning can be used everywhere. There has been plenty of research done in this field, and we had come a long way from when the computers were first created. There is still, however, a long way to go. Computers have gotten a lot more capable and are faster than ever before, allowing us to apply machine learning to complex applications, but it is still nowhere near as good as humans when it comes to learning. A human's ability to create intricate patterns from the given data is challenging to recreate synthetically.

In terms of finances, we know that the stock market can be predicted to some extent given the profits that good traders make. Using machine learning to predict if the stock market is moving up or down still only achieves a little over 50% accuracy [23] . While humans are somehow able to create patterns regarding stock values and the factors that can affect them, a computer is as yet unable to do so to a satisfactory level.

According to Tom Mitchell, Machine learning is that domain of computational intelligence which is concerned with the question of how to construct computer programs that automatically improve with experience [24]. "Improve with experience" means learning. This begs the question, what is learning? Learning is a function that allows animals to modify and reinforce or acquire new information, skills, patterns and behaviors [25]. This ability that animals possess is what sets them apart from machines.

Most of the modern machine learning research focuses on creating various algorithms that can be used on specific applications. Using machine learning a model is created from data. This model can then be used to make predictions. The main difference between regular computer code and a machine learning code is that in a machine learning code the rules are not hardcoded, the code creates a model which has its own set of rules.

A prime requirement for machine learning is having access to a large amount of data. Generally, the better the data the better is the machine learning algorithm. The algorithm tries to find a correlation between the input and the expected output. This is accomplished by separating the data into two groups – training data and testing data. The model is generated with the training data, and its performance is tested on the testing data. A significant part of machine learning is getting the right data. The data must be relevant, and of the correct size. More about this is discussed in the later part of this thesis.

Broadly, machine learning is split into three parts – supervised learning, unsupervised learning, and semi-supervised learning.

## 3.1 Supervised Learning

A supervised learning algorithm observes some example input-output pairs and learns a function that maps from input to output [26]. Supervised learning algorithms can be used where we have labeled data, meaning there is an explicit dataset consisting of an input and expected output data. Expected output, also known as target data, is the output for a given input data which the machine learning algorithm should learn. Data can be generated using various means depending on the application and the datatype. For finances, input data is usually the stock values, and the target data is usually a value over a particular time jump. If we are trying to predict stock value over one day, then the expected output would be the close values of stock after the next day.

Supervised learning algorithms analyze the training data and create a function, and this function can be used for mapping new examples. An ideal case scenario can produce fantastic results with supervised learning algorithms. It can give insights into instances which were unexpected and help improve the overall functioning of an application. For instance, in stock markets, if the right data, in the right conditions are applied to a supervised learning algorithm then it should be able to give a very high accuracy for the output because, theoretically, it can create patterns that were previously unknown to man. The whole reason for there being so much research into stock predictions is because the ”ideal” conditions are yet unknown. Supervised learning is further divided into two types based on the kind of problem Regression and Classification.

### 3.1.1 Regression

Regression is a concept that was first introduced in the early 18th century, but regression as we know it today was developed by Karl Pearson [27]. Regression is used where continuous values are being predicted. Stock price prediction is typically a regression problem. In this application, we have continuous data that changes every minute of every day as long as the market is open. Regression analysis is primarily used for estimating a relationship between two or more variables. Hence it is used in predictions or forecasting based applications. Regression analysis can not only indicate a significant relationship between dependent or independent variables, but it can also indicate a strength of the relationship between multiple independent variables on a single dependent variable. This is a very important part of the regression analysis that is used in this thesis. Market data, news articles, correlated stocks, and its stock data are used as input. They are all independent variables or at least as independent as features can be in stocks. This thesis attempts to prove that all these various factors can provide a more accurate prediction of stocks and by using regression analysis over all these seemingly independent variables, it is possible to get a very accurate stock price prediction.

Regression analysis can be further divided into various types. Some of the more common types of regression are linear regression, logistic regression, polynomial regression, and stepwise regression. These types are mainly divided based on the number of independent variables, the number of dependent variables and the shape of the regression line. Linear regression, which is the most widely used regression analysis technique, has a linear regression line and it establishes a relationship between a dependent variable and one or more independent variables. Logistic regression is used for binary dependent variables.

### 3.1.2 Classification

Classification is used where discrete values are being predicted. This is applied where there is input data that is to be classified into discrete classes. An example of classification is image classification or speech recognition. In many real-world cases, however, classification might not always be a good approach. There might be inputs that can belong to border cases with features from both classes. Sometimes there might be inputs that do not belong to any of the classes. So even though classification seems to be a more natural approach on paper, it is wise to use regression when possible.

## 3.2 Unsupervised Learning

Unsupervised learning algorithms are used where there is training data which does not contain any information about the desired output. The main purpose of unsupervised learning is to understand the data by getting the basic structure of the data. Unlike supervised learning, with unsupervised learning, there is no correct answer as there is no output for the training data to learn from. The algorithms learn from just the input data and discover some exciting structure from the data. Unsupervised learning can be divided into two kinds.

### 3.2.1 Clustering

Clustering can be loosely defined as "the process of organizing objects into groups whose members are similar in some way" [28]. In this algorithm, the data is portioned into distinct groups such that each of the groups is similar in some way. A common type of clustering algorithm is k-means clustering.

### 3.2.2 Dimensionality Reduction

As the name suggests, this algorithm is about reducing the complexity of data while keeping its relevant structure [29]. Many times in classification problems, we come across a situation where there are too many features. When there are too many features, the training set can be too large to visualize and work on; some features can also be correlated and so unnecessary. The most common type of dimensionality reduction method is Principal Component Analysis (PCA) [30].

## 3.3 Semi-Supervised Learning

This type of learning is used when there is a large amount of input data, and only some of that data has labels. Semi-supervised learning has many real-world applications. In real-world applications collecting labeled data is very expensive, whereas unlabeled data is much cheaper. Unsupervised learning can be used on the unlabeled data to make a prediction. That predicted data can be used as input for the supervised learning algorithm, and the final model can be used on a separate test data. There have been many models of semi-supervised learning going back to the 1960s, but the more modern form of semi-supervised learning was developed by Dr. Oliver Chapelle in 2006 [31].

## 3.4 Neural Networks

One of the more widely used machine learning algorithms is artificial neural networks. It is a robust approach that can approximate various types of target functions. Target function is the function used to map the input data to the target value. This ability of neural networks to extract patterns from complex data is what makes them so accessible. Neural networks consist of a large number of interconnected elements called neurons which work in parallel to solve a problem.

Neural networks were first introduced in 1943 by Warren McCullough and Walter Pitts [32]. This initial approach to neural networks relied heavily on the neural setup of the human brain, an idea that is still loosely followed to this day. Neural networks have been on and off in terms of popularity over the years, since their introduction, mainly because of the lack of hardware capability to handle neural networks for complex data. In recent years, however, there has been much interest in neural networks, accompanied by numerous research papers. In this thesis, deep networks are used extensively, which is an extension of neural networks.

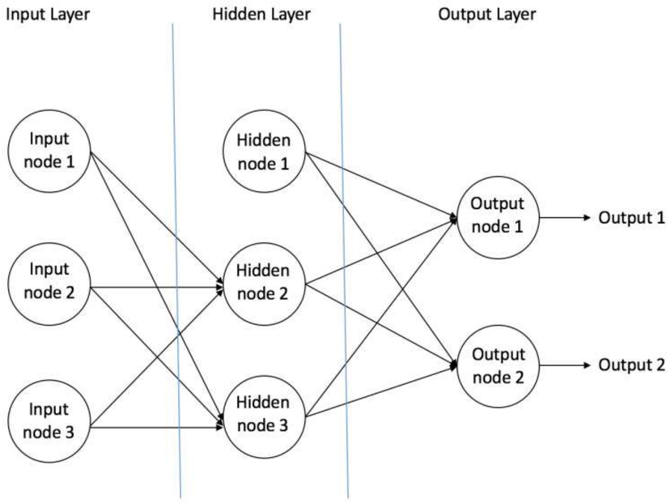

**Figure 3.1:** Feed Forward Neural Network.

Figure 3.1 is an example of a simple feed-forward neural network. In this type of network, no information is ever fed backwards, when calculating the outputs. Neural networks consist of the following components.

### 3.4.1 Neurons (Nodes)

A neuron or a node is the most basic component of a neural network. It receives one or more inputs from other nodes or a source and computes an output.

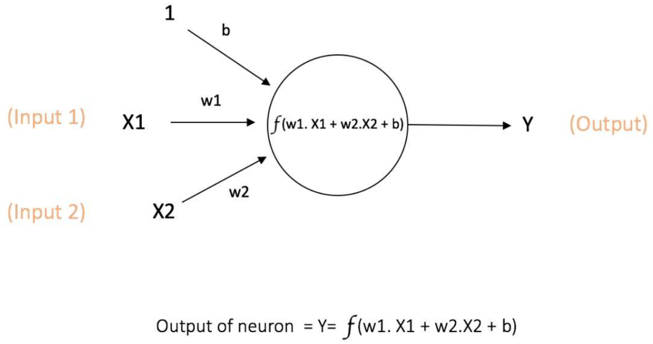

**Figure 3.2:** Nodes.

Each input is connected with a weight (w). The weight is based on the importance of the input. From Figure 3.2 , x1 and x2 are the inputs, w1 and w2 are the weights, b is the bias, y is the output and f is the activation function.

The original concept of a neuron was introduced by Warren McCullough and Walter Pitts [32] but the modern version of a neuron was developed by Frank Rosenblatt [33].

### 3.4.2 Activation Function

Real world data is usually non-linear, so it is vital the neurons learn this non-linearity. The activation function is used to introduce non-linearity into the output of a neuron. The activation function is a fixed mathematical operator; it takes a single number and performs certain fixed mathematical operations on it [34]. There are different activation functions, some of the more common activation functions are:

**Sigmoid:** It gives an output in the range between 0 and 1.

$$\sigma(\mathbf{x}) = \mathbf{1}/(\mathbf{1} + \mathbf{e}^{-\mathbf{x}}) \tag{3.1}$$

**Tanh:** It gives an output in the range between -1 and 1.

$$\tanh(\mathbf{x}) = \mathbf{2}\sigma(\mathbf{2x}) - \mathbf{1} \tag{3.2}$$

**ReLU:** In this type of function the activation is thresholded at zero.

$$\mathbf{f}(\mathbf{x}) = \mathbf{max}(\mathbf{0}, \mathbf{x}) \tag{3.3}$$

### 3.4.3 Input Layer

The input layer is the first layer of a neural network. It takes external input values and bias as its nodes. There is no computation in the input layer and the values from the input layer are fed into the nodes of the hidden layer.

### 3.4.4 Hidden Layer

The hidden layer takes input from the input layer and does some computation before moving it on as shown in Figure 3.3. In this example, we can see how the output is computed in the selected node where "f" is the activation function. The output from the hidden layer is forwarded to the output layer, but it is possible to have multiple hidden layers wherein the output is fed to the next hidden layer.

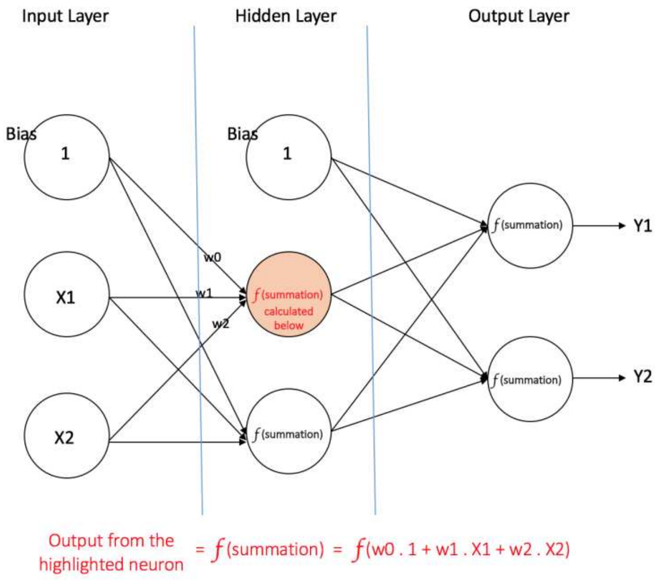

**Figure 3.3:** Hidden layer in a Feed Forward Neural Network

### 3.4.5 Output Layer

The output layer takes inputs from the previously shown hidden layer and performs similar computation as in the hidden layer. In most cases, the output layer does not have an activation function as this layer is used to represent the output values. The output values are generally a real-valued number or some real-valued target.

### 3.4.6 Weights

A weight is used to represent the strength of the connection between values in the nodes. A higher weight value implies the stronger connection between the nodes and vice-versa. This stronger connection can be positive or negative based on whether the weights are positive or negative.

### 3.4.7 Learning Rate

With neural networks, weights are optimized using gradient descent. Learning rate determines the rate of update of the weights. Learning rate is significant to machine learning as a high learning rate can cause the weights to find local minima or not converge and a low learning rate can cause the weights to take too long to converge to an appropriate value.

### 3.4.8 Overfitting and Underfitting

Overfitting is an occurrence in machine learning algorithms during which the model can learn the noisy training data to an extent where it can have a negative impact on the final output. This happens when the model gives a high accuracy with the training data, but the results are not replicated with new or testing data. Overfitting typically occurs when the model is too complex. One of the most common ways to avoid overfitting in neural networks is to use a technique called dropout [35]. In this technique, some units are randomly dropped from the network during the training phase. Using a random probability method any unit can be dropped, which in turn would create a subset of the units from the original network.

Underfitting is when the model has not learning enough from the training data. If the model is under fitted, then it does not perform well on both the training and test phase. If a model is under fitted the easiest way to solve it is to increase the complexity of the model.

## 3.5 Deep Neural Networks

The neural network in Figure 3.1 has a single hidden layer, also known as a shallow neural network. A deep neural network has multiple hidden layers. In many complex cases and if trained properly, a deep neural network can produce much better results than a shallow neural network. A deep neural network can have some issues, such as vanishing gradient, but it has been applied in various applications in recent years such as image recognition and speech recognition. More about vanishing gradient will be discussed in a later chapter.

Two typical applications of Deep Neural Networks are image-based and sequence-based. Broadly speaking an approach called Convolutional Neural Network or its variant can be used in image-based applications and Recurrent Neural Network or its variant is particularly useful in sequence-based applications, it can provide good performance in text-based and speech-based applications. Since this thesis uses a Recurrent Neural Network variant, it is discussed in more detail.

## 3.6 Recurrent Neural Networks (RNN)

The recurrent neural network [36] is one of the few types of neural networks that have feedback. A traditional neural network assumes that the inputs are independent of each other. For many real-world applications, data may not always be independent, particularly in sequential data such as music or textual data. Having feedback allows the network to understand the context within a sentence or a paragraph from textual data or a sequence of tones within a musical track that is pleasing to human ears. The feedback loop in RNN can be thought of as ”memory” since it allows the captured information about certain things that were calculated up to that point.

Figure 3.4 shows a rolled and an unrolled structure of RNN. Unrolled or unfolded simply means that the network is written out for the full sequence. The terms in Figure 3.4 are:

$x_t$ = input at time step t,

$h_t$ = hidden state at time step t,

$y_t$ = output at time step t

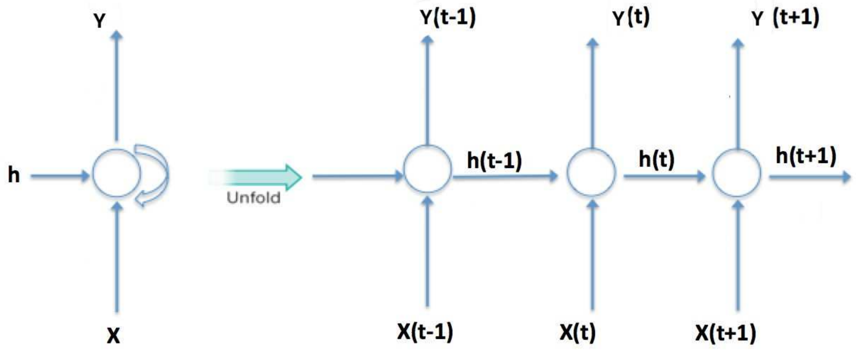

**Figure 3.4:** Recurrent Neural Networks.

Recurrent Neural Networks are particularly useful for short-term dependencies. Long-term dependencies, however, face some issues, primarily an issue called the vanishing gradient [37]. Vanishing gradient occurs when the gradient becomes very close to zero. This phenomenon is not exclusive to RNN's. This is fairly common in many deep networks. Sometimes with deep networks, the weights in the starting layers change at a very slow pace which means that the right solution is never achieved.

## 3.7 Long Short-Term Memory Networks

Long Short-Term Memory units are a special type of units within a Recurrent Neural Networks. An RNN consisting of Long Short-Term Memory units is called a Long Short-Term Memory network and will be referred to as LSTM henceforth in this thesis. The vanishing gradient issue can be overcome using this approach. LSTM was first proposed by German researchers Sepp Hochreiter and Juergen Schmidhuber in 1997 [38]. Despite the complicated architecture of this network it is shown to perform well on a large variety of problems [39–41] and hence is used extensively around the world.

The main difference between a regular RNN and an LSTM is the ability of the LSTM to distinguish between when the memory needs to be cleared and when data needs to be read from memory. In an RNN, incremental memory is always used, which makes it impractical for long-term dependencies. An LSTM unit shown in Figure 3.5 uses a concept called gating which is essentially just component-wise multiplication. LSTM's complex structure can be split into four components - cell, input gate, output gate and forget gate. All the gates are essentially switches; the input gate decides if the input is written to the cell, output gate decides if the output from the cell should be read and forget gate decides if the previous cell value should be reset. The cell is where the output value is calculated using the input and the previous cell value. Below $i_t$, $f_t$, $o_t$ are outputs from the input gate, forget gate and output gate, respectively.

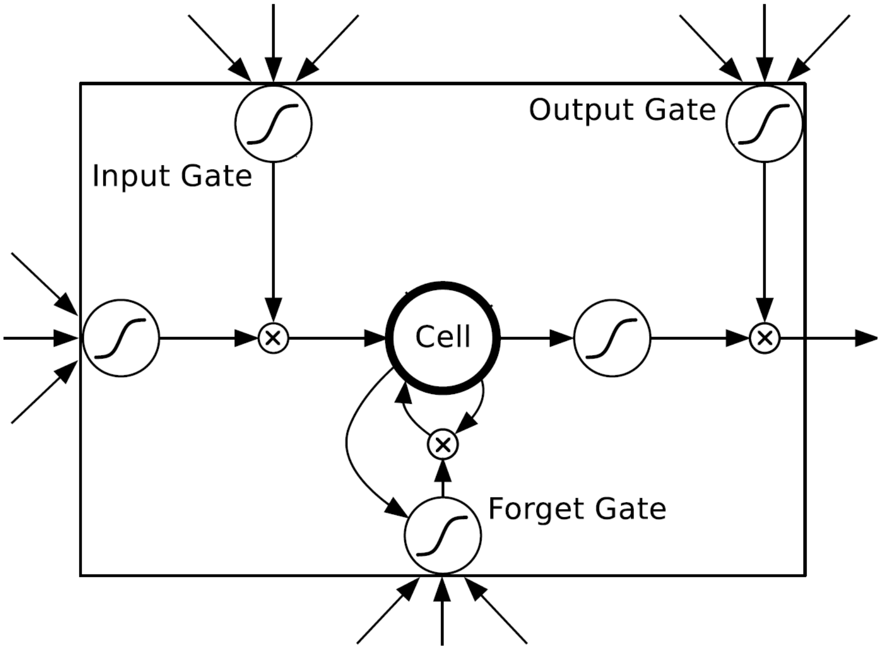

**Figure 3.5:** The internal architecture of a LSTM.

$$\begin{aligned}
\mathbf{i_t} &= \sigma(\theta_{\mathbf{xi}}\mathbf{x_t} + \theta_{\mathbf{hi}}\mathbf{h_{t-1}} + \mathbf{b_i}), \\
\mathbf{f_t} &= \sigma(\theta_{\mathbf{xf}}\mathbf{x_t} + \theta_{\mathbf{hf}}\mathbf{h_{t-1}} + \mathbf{b_f}), \\
\mathbf{o_t} &= \sigma(\theta_{\mathbf{xo}}\mathbf{x_t} + \theta_{\mathbf{ho}}\mathbf{h_{t-1}} + \mathbf{b_o}), \\
\mathbf{g_t} &= \tanh(\theta_{\mathbf{xg}}\mathbf{x_t} + \theta_{\mathbf{hg}}\mathbf{h_{t-1}} + \mathbf{b_g}), \\
\mathbf{c_t} &= \mathbf{f_t} \odot \mathbf{c_{t-1}} + \mathbf{i_t} \odot \mathbf{g_t}, \\
\mathbf{h_t} &= \mathbf{o_t} \odot \tanh(\mathbf{c_t})
\end{aligned} \tag{3.4}$$

They each give a value of close to 0 or 1 acting as a switch. $g_t$ is input that enters the cell ($c_t$) and $\theta_{xi}$, $\theta_{xf}$, $\theta_{xo}$, $\theta_{xg}$ are values that are learned over multiple iterations of training and $\odot$ is dot product of the values. The beauty of this network is using the gating approach; the network learns when to turn the switches on or off. Hence it will decide which part of the information is necessary and which part is not.

# Chapter 4

# Methodology

This chapter discusses the method used to obtain the results. The first section discusses the environment of the test including the tools used. The second section discusses the data used and how it was obtained and preprocessed. The third section talks about the implementation of the algorithms.

## 4.1 Test Environment

The main target of the thesis is to establish the connection between news articles and stock prices using machine learning. One of the methods for establishing this connection was to compare the predicted stock prices using results from the news articles with the predicted stock prices without the news articles.

As mentioned in Chapter 2, stocks are dependent on various factors. News alone may not be enough to give an accurate stock prediction. Correlated stocks and the market as a whole can have a massive impact on the selected stock and the best way to verify this impact, if there is any, is to compare the stock price predictions with and without the correlated stock values and the market value. Taking the above into consideration, two separate datasets were generated. First was the stock data of various companies along with the correlated stock's data for each of those companies and the market value for that time period. The second was the articles based on the selected companies for the same time period.

After the data was created, an LSTM network was trained to predict the change in the Close values of a stock given the relevant news articles. The output from the test data for the LSTM was also used as part of the input for a second deep network that was used for the actual stock predictions. The second deep network was used to train the stock values of the selected stock, its correlated values and the market price. Typically, the open, close, high and low values are enough to give a reasonably accurate close value of a stock for the next day. The idea behind using the other factors is to try to achieve a better prediction.

Since the algorithms are heavy in terms of computation, the code was written in Keras with Tensorflow-GPU backend. This enables the keras code to run on the GPU, decreasing the computation time. The stocks selected were Microsoft, Google, Netflix, Apple, Nvidia, AMD, and Amazon. The algorithm was used on each of them individually to check if they are all impacted by the news in similar ways.

## 4.2 Data Used

For a machine learning algorithm to work properly, one of the most important factors is data. Choosing the right data is extremely important, especially in the field of stock markets. The right data is hard to come by as it is not really known what the right data is. There are many papers using different data for stock prices, and not many of them agree on which is most relevant. To eliminate this issue, the entire stocks prices available in NASDAQ were used to get the stock prices over the last five years. This data was also used to find the correlated stocks.

There is always the issue of overfitting with machine learning, and it is also possible that the algorithm can find correlations without causation. There is a possibility that there may not be enough data concerning stock prices as mentioned in Chapter 2. There is no real way to know if there is too much or too little data. For each of the stocks from the NASDAQ market, Open, Close, High and Low values were used.

The input data was the data from all the above, and the output was the close value of the same stock on the next day.

It has been mentioned many times in the previous chapters that using information that is too old can cause incorrect predictions, but at the same time, there might not be enough data if only the recent history is considered. In an attempt to avoid this, the change in stock prices is used instead of the actual stock prices. Using this approach, the network might be able to learn the reason for the change in stock prices, which could remain the same over a significant period of time.

Once the stock prices were downloaded, the next step was to get the news articles for the stocks. Using a third party company called the Event Registry, a list of all the news articles generated on some of the most popular financial news websites, such as marketwatch.com, Investopedia.com, Bloomberg.com, and thestreet.com over the last five years were compiled. These articles were sorted based on their relevance to each of the stocks mentioned above. The relevance was determined by calculating the number of times a company's name was mentioned in the article. Once the articles were separated, they were then joined with the change in close values of the stock prices of the respective companies. The change in stock values was calculated as the percentage change in close value between the day the news was generated with the value of the close stock price on the next day.

The final stage was to deal with the missing data. The news is generated every day, whereas the stocks are generated only on days when the market is open. This was dealt with by adding the news articles to the last open market day. For instance, if the news is generated on the weekends, then this news is added to the Friday's news. Another issue was dealing with inactive tickers. As mentioned above all the stock data over the last five years was used to find the correlated stocks. Over that period of time, some companies that were correlated to some of the companies used in this thesis have gone defunct. These were removed from the final dataset.

### Splitting the Training and Testing Datasets

For the news articles, the training data was the news generated between 2012 -2016 shown in figure 4.1. The testing data was the news generated in the last 11 months (2016-2017). An LSTM network was used to predict the potential movement of stocks for the next day based on the news articles from the previous day.

This output generated from the news data was then used as part of the input used in the second deep neural networks for the stock price prediction as can be seen in fig. This data henceforth called LSTM data was used alongside the open, high, low, close values and the respective correlated stocks for the stock predictions. The training data for the second deep neural networks was the

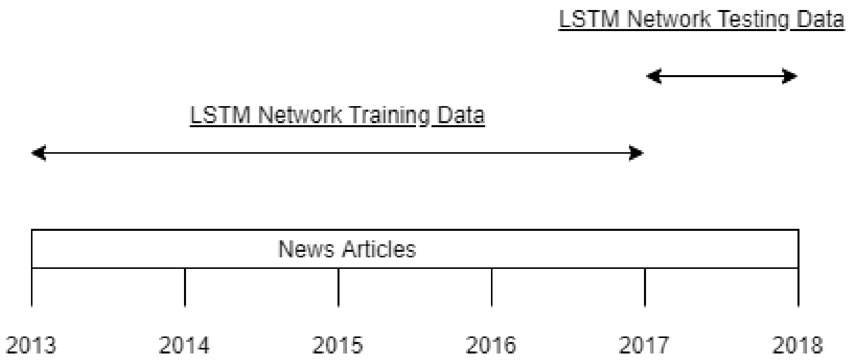

**Figure 4.1:** Deep Neural Network Train and Test data split

first 9 months of the 11 months and the last two months were used as the testing dataset as can be seen in figure 4.2.

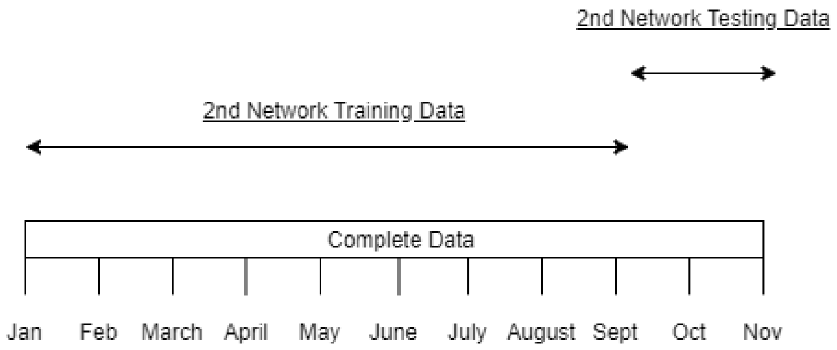

**Figure 4.2:** Deep Neural Network Train and Test data split

Before getting into the finer details of the algorithm, the other factors which were considered need to be discussed in depth. The other factors are the correlated stocks and the market value. The five-year data of the stock prices were used to find the correlated stocks for each of the companies mentioned in the thesis. Market prices were also downloaded for the same period which was included in the final dataset. When the data is ready to be trained, it will look something like the example in table 4.1, which contains the values:

1. "Pred" – predicted output from the LSTM data,

2. DJI – market value of Dow Jones Industrial Average,

**Table 4.1:** Typical Dataset

| Date | Pred | DJI | ADBE | Open | High | Low | Adj Close | Adj Close Output |
|---|---|---|---|---|---|---|---|---|
| 6/8/2017 | 0 | -0.05672 | -0.03211 | 0.000903 | -0.0036 | -0.06273 | -0.04732 | -0.0417 |
| 6/9/2017 | 0.5 | -0.00894 | -0.0058 | -0.06598 | -0.06459 | -0.04006 | -0.0417 | 0.008452 |
| 6/12/2017 | 0 | 0.011502 | 0.013406 | -0.00592 | 0.000964 | 0.012272 | 0.008452 | -0.0034 |
| 6/13/2017 | -0.5 | -0.0194 | -0.00604 | -0.00026 | -0.00039 | 0.000999 | -0.0034 | -0.00289 |
| 6/14/2017 | 0 | -0.01864 | -0.00528 | -0.03175 | -0.01966 | -0.01983 | -0.00289 | 0.004085 |
| 6/15/2017 | 0.5 | -0.00116 | 0.002327 | 0.01345 | 0.006358 | 0.020978 | 0.004085 | 0.006694 |
| 6/16/2017 | 0 | 0.027372 | 0.01821 | 0.018752 | 0.013352 | 0.013432 | 0.006694 | -0.0088 |
| 6/19/2017 | 0.5 | -0.021 | 0.00399 | -0.00395 | -0.00694 | -0.00663 | -0.0088 | 0.019599 |
| 6/20/2017 | 0.5 | 0.005996 | 0.023632 | -0.00768 | 0.00568 | 0.00568 | 0.019599 | -0.0009 |

3. ADBE – correlated stock,

4. High, Low, Open and Adj Close – Values of a particular stock

The correlated stocks were only considered if the correlation value was higher than 0.5 and some of the highly correlated stock are defunct and hence were removed.

The data was a collection of the percentage change in stock values for all the columns. There are 2 'Adj Close' columns, the first one is part of the input and the second is the output. If we look closely at the values, we can see that the first row of the 'Adj Close' is the same as the value in the second row of the 'Adj Close Output' column. This is because the input is predicting the Close values for the next day.

The 'Pred' value, as mentioned before, is the output of the LSTM data. The way the LSTM data is built is that all articles generated on any particular day are given an output value of either +1, 0 or -1. These values are calculated based on the percentage change in the Close value of the stock for the next day. If the change is between -1.5% and +1.5% then a value 0 is given as output. If the change is less than -1.5% then a value of -1 is given as output. If the change is greater than +1.5% then a value of 1 is given as output. Using a threshold higher than 1.5% seems to give too many days with a 0 value or no movement in stocks and using a value below 1.5% threshold seem to give too many +/-1 values even if the news articles probably had no impact on the stock movement. Because the data is considered in this fashion, there can be over 50 news articles that all lead to the same output. The general idea is that over time the algorithm can understand which articles can actually impact the stock prices and which ones do not. This does not only consider the sentiment of the articles, but it can potentially consider the other factors that can have the impact of the stocks.

## 4.3 Metrics

Metrics are fundamental to judge the performance of an algorithm. For this, since the thesis has two networks, logically it would be fair to get two different metrics to calculate the performance. Although this is done and the results are explained, the results of the classification from the LSTM network does not count for much as the goal of the thesis was to establish a correlation between news articles and the respective company's stock prices. To establish this connection, the best way to do it was to combine the results from the LSTM network with the input of the deep neural network to make a prediction for the stock values. Using this approach, if the results with and without the LSTM output are compared, then there should be a clear difference in the final output. The metric used to measure the model performance is RMSE divided by the difference between the maximum target value and the minimum target value. Additionally a graphical approach is also applied; the difference in the graph should show the impact of the news articles.

To make the impact of the news and correlated stocks more distinct, there are four final graphs for each of the companies:

1. With LSTM network results and Correlated Stocks – Graph A

2. With Correlated Stocks and No LSTM network results – Graph B

3. With LSTM network results and No Correlated Stocks – Graph C

4. No LSTM network results or Correlated Stocks – Graph D

A bar chart is added as a secondary axis value consisting of the LSTM network results. This secondary axis is only added to graphs that contain the LSTM results. This should help us determine if the LSTM network had any impact on the final results from the second deep network.

## 4.4 Long Short-Term Memory Network

This thesis uses two networks. To describe the methodology better, each network is explained separately. The first one is the LSTM network that is used to classify the news articles. Pre-processing is an essential part of any machine learning algorithm. LSTM also needs the same kind of attention in terms of pre-processing. In the previous chapters, the quantity of data was discussed in great detail, but the part that was not mentioned too much was the quality of data. The quality of data is just as important as quantity; it is vital to get the right data and in the right format.

### 4.4.1 LSTM Network Pre-Processing

As mentioned in Chapter 3, the LSTM network takes input in the form of numbers. So the first part of setting up the network is to convert the articles into a series of numbers, with each word getting a unique number to represent it. Before that can be done, the commonly used words such as 'the', 'a', 'it'; also known as stop words need to be removed along with special characters like '.' , ',' , ':', etc. These are part of sentences that get repeated a lot in an English paragraph. An LSTM can make wrong predictions based on the many repetitions of the stopwords. By removing them, one can ensure that only the relevant words are considered in the network. Using NLTK [42] functions called stop words and word_tokenize, the stopwords in the articles and remaining words were converted into unique numbers. An additional process called stemming was also done so as to further decrease the word count and keep only the unique words which tend to have more impact in the classification process. Stemming is the process of reducing derived words from their stem or base words, such as reducing running or runner to run. LSTM can not read strings so the individual stemmed words needed to be converted to unique numbers, this was achieved using a function called token2id from the gensim library. After this the sequences had to be padded together using pad_sequences from keras library. This process is needed since keras needs vectorized inputs of equal length.

The next part of the LSTM network was to limit the size of each input. In this thesis, the size of the input was set to 50 words which means that only the first 50 words were used as input to

LSTM to classify the whole article. This number was set arbitrarily with the assumptions that a typical introduction paragraph for an article would be around this size. The dataset for each of the selected companies consists of around 25000 articles, but this was not a large enough dataset for the task at hand. Using the whole article to train the network is not realistic.

### 4.4.2 LSTM Model

Long-Short Term Memory is a deep network, so the model used has four layers. The first layer was the sequential input layer. The second is an embedded layer. Our input is a series of catogorial indicies and the embedded layer converts these positive integers into a dense vector of fixed sizes. Internally the embedding layer converts the positive integer into one-hot vector and is then multiplied by a random weight matrix. These weights are updated through the training process. The inputs given to this layer are the dictionary size, which was a collection of unique words in the dataset. Since we can't use all the unique words in the dataset, this number was limited to 10,000. The way a multilayer neural network works is by connecting one end of a layer to another, this can only be done if the output dimensions of one layer match the input dimension of the next layer. In case of our embedded layer, the output dimension was 128. The second layer was the LSTM layer; this layer takes the input from the embedded layer and performs its operations before giving the output to a dense layer. The dense layer, which is also called an output layer in a neural network gives an output from one of the three potential outputs $(-1, 0, 1)$. The activation function used was 'Softmax', the optimizer was 'Adam' [43] with a learning rate of 0.005 and decay of 0.05. The loss was calculated using 'categorical cross-entropy'; this was used because each example belongs to a single class. Since this is a classification problem, using softmax was the best approach as it can give a probability that a class is true. All the parameters were chosen by conducting multiple trials and the parameters that provided the best results were selected. 'Adam' is an optimizer that is becoming popular in recent times. It is different from traditional stochastic gradient descent. Adam can be called a combination of Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp). RMSProp is a commonly used optimizer; it adapts the parameter learning rates based on the average first moment of the gradient. Adam builds on RMSProp by using the average of the second moments of the gradients as well. The fit function used ten epochs and a batch size of 512, numbers obtained by multiple tests to get the best solution. The bias was initialized to 0, and the weights were initialized using the glorot uniform function also known as the Xavier initialization [44].

The dataset for the LSTM network has multiple articles for each day but the dataset for the deep network, used for the stock price prediction, has only one row to represent a day. The mean average of the output from all the news articles of the LSTM network for each day was taken to join these datasets.

## 4.5 Second Deep Network

The second network is a simple multi-layer neural network. This network is run multiple times for different datasets as mentioned in Section 4.2.

### 4.5.1 Deep Network Pre-Processing

The pre-processing for the deep network is much simpler compared to the LSTM network pre-processing. The stock data obtained from Yahoo finance API, using ’pandas_datareader’ library, is in the form of stock values for each day when the market is open. This included the Open, Close, High and Low values for each company. These values were transformed into percentage change values, as historical data from a more extended period of time will have less impact on the stock prices but using a percentage change in stock prices from one day to the next can offer more accurate predictions as the network can find different relations within the data. Once the data is obtained and merged with the LSTM network, the data is ready to be used in the deep network.

### 4.5.2 Deep Network Model

The deep network has four layers: the first layer has 128 nodes, the second layer has 64 nodes, the third layer has 16, and the fourth layer has 1. All the layers use ’Tanh’ as the activation function, and the optimizer is set to ’Adam’ with a learning rate of 0.001 which is the default value. Since this is a regression problem, the loss function is Mean Square Error. The bias was initialized to 0, and the initial weights are set using the glorot uniform function. Similar to the LSTM network, the parameters were selected from the best performing network after various trials.

As mentioned in Section 4.3, the metrics used to measure the performance was a comparative analysis of stock predictions under various conditions. The whole dataset is similar to the table shown in Table 4.1 but for the rest of the graphs mentioned in the metrics section, if the graph shows LSTM and no correlated stocks then all the correlated stocks are removed. Similarly, if the graph shows correlated stocks and no LSTM, then the ’Pred’ column was removed from the dataset.

# Chapter 5

# Results

In this chapter, the LSTM results for each of the companies is discussed first, and then the stock predictions for each of the companies will be compared for each of the conditions mentioned in Section 4.3.

## 5.1 LSTM Network Performance

Table 5.1 shows the training and testing accuracy of the LSTM network from the news articles for each of the companies. We can see from the table that the accuracy of the test dataset fluctuates widely. This is because of the volatility of the stock values of the companies themselves. Companies such as AMD, Netflix, and Nvidia are not as stable as the other companies, and every small change in these companies stock value seems to generate a chain reaction that causes a higher difference in value for each day. Over the years, these companies pivoted to more more extensive areas in their respective markets causing more factors to have an impact on their stock value, some of which may not have been picked up by the training network.

The results show the accuracy of the LSTM, but it does not automatically imply that the target of this thesis was achieved. The accuracy only shows that the LSTM network can potentially identify which articles can have an impact on stock prices. To be able to unequivocally prove that articles have an impact on the stock prices, the output from the LSTM network needs to be

**Table 5.1:** LSTM Network Performance

| Companies | Training | Testing |
|---|---|---|
| Apple | 96.860 | 84.793 |
| Amazon | 97.358 | 81.471 |
| AMD | 93.978 | 49.424 |
| Google | 98.047 | 87.885 |
| Microsoft | 96.664 | 91.416 |
| Netflix | 96.065 | 63.280 |
| Nvidia | 97.013 | 63.242 |

**Table 5.2:** LSTM Network Precision Values

| Companies | Predicted Value | Precision | Recall | Total Actual Values |
|---|---|---|---|---|
| Apple | -1 | 0.16 | 0.53 | 506 |
| Apple | 0 | 0.91 | 0.91 | 6221 |
| Apple | 1 | 0.12 | 0.52 | 816 |
| Amazon | -1 | 0.15 | 0.51 | 242 |
| Amazon | 0 | 0.97 | 0.82 | 7449 |
| Amazon | 1 | 0.09 | 0.54 | 124 |
| AMD | -1 | 0.34 | 0.32 | 1760 |
| AMD | 0 | 0.53 | 0.63 | 4095 |
| AMD | 1 | 0.39 | 0.35 | 1961 |
| Google | -1 | 0.33 | 0.54 | 311 |
| Google | 0 | 0.97 | 0.90 | 7326 |
| Google | 1 | 0.16 | 0.53 | 180 |
| Microsoft | -1 | 0.15 | 0.52 | 82 |
| Microsoft | 0 | 0.98 | 0.92 | 7669 |
| Microsoft | 1 | 0.09 | 0.62 | 66 |
| Netflix | -1 | 0.16 | 0.23 | 681 |
| Netflix | 0 | 0.86 | 0.69 | 6442 |
| Netflix | 1 | 0.19 | 0.47 | 693 |
| Nvidia | -1 | 0.29 | 0.39 | 822 |
| Nvidia | 0 | 0.78 | 0.73 | 5696 |
| Nvidia | 1 | 0.32 | 0.33 | 1298 |

combined with the stock values and the stock predictions during that period need to be compared with the predictions without the results from LSTM network. This was a complex network but since the size of each sequence was set to 50 words, all networks took between 3-5 minutes to get the results.

Table 5.2 shows the precision and recall values, this is another way to look at this network's performance. Majority of the values are 0 and this is by design. We only need the LSTM network to pickup the articles that are directly impacting the stock price movement. The stocks that are relatively stable seem to have comparitively lot more 0's as can be expected and for this reason the precision values for both 1 and -1 are very low. This seems to be very prominent with Amazon and Microsoft in particular.

### 5.1.1 Second Deep Network Performance

As mentioned earlier the network had four layers with 128, 64, 16 and one nodes, respectively. This is probably too powerful and maybe unnecessary, but the idea was that given the complexity of the stock markets, this network might give better accuracy. The complete input data for the second deep network would include the results from the LSTM network, the top correlated stock with correlation higher than 0.7 or less that -0.7 and the history open, close, high and low of that selected company stock. Before discussing the results, it's important to reiterate some of the vital information mentioned in the previous chapters - the stock prices can be affected by various factors, during certain periods of a year some factors can be more influential than the others and regardless of how important some of the other factors are, general trends of a particular stock have the most impact on its future price.

Since historic trends are one of the main factors and historical data of each of the stocks is used in all the training data, all the graphs for each of the companies look very similar. The focus, however, is on the peaks. If the network picks up the change in the direction of the stock value and can predict the major change in prices, then it suggests that stocks can be predicted using correlation and news articles. As mentioned in section 4.3, we will look at 4 different RMSE values and their respective predictions for the test data. Each of these companies have also been tested with a baseline model to see if the Deep Neural Network provides a better solution. The baseline model is a neural network with no hidden layers, which is essentially a linear model. This can be considered equivalent to linear regression. The RMSE values from the baseline model is also shown along side the RMSE values from the Deep Neural Network.

The results for each of the selected companies are discussed below. A common point that we will notice with all the graphs is that there is very little difference in the results for each of the networks. This indicates that the features that seem to impact the predicted values most are the historic stock values of that same stock. The news articles and the correlated stocks seem to have little impact on day to day stock prices.

**Table 5.3:** RMSE values for Apple on test dataset

| Dataset | RMSE Testing | RMSE Baseline |
|---|---|---|
| LSTM Network Results and Correlated Stocks | 0.0109 | 0.0113 |
| Correlated Stocks | 0.0109 | 0.0112 |
| LSTM Network Results | 0.0117 | 0.0120 |
| Neither LSTM Network Results or Correlated Stocks | 0.0120 | 0.0120 |

## Apple

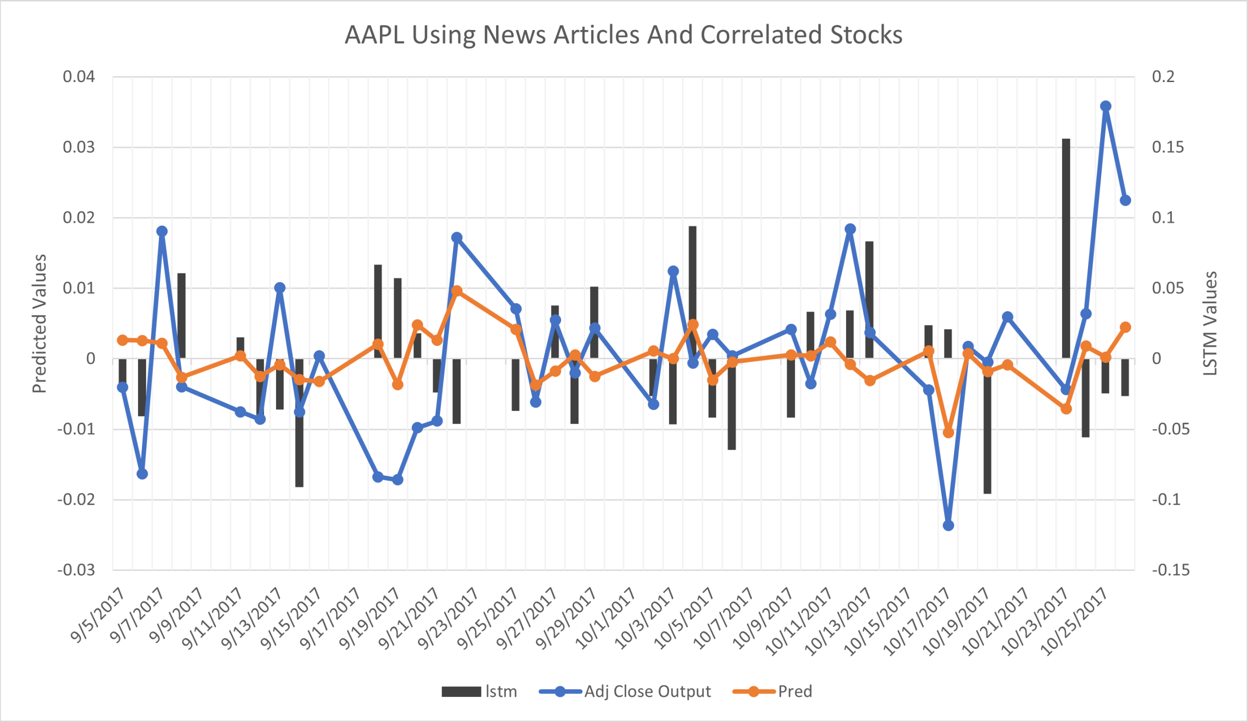

**Figure 5.1:** Apple's graph with Complete Data

One of the more volatile stocks used in this thesis is Apple. Figures (5.1, 5.2, 5.3, 5.4) show that most of the stock price change is between 1.5 and -1.5 percent. However, it has 7 values that are more than this range. It is interesting to note that figures 5.3 and 5.4 have similar predicted values and figures 5.1 and 5.2 have similar values. This could indicate that the correlated stocks seem to have more impact on the predicted values. However, if we look at the table 5.3, RMSE values for figures 5.1 and 5.4 have the lowest scores.

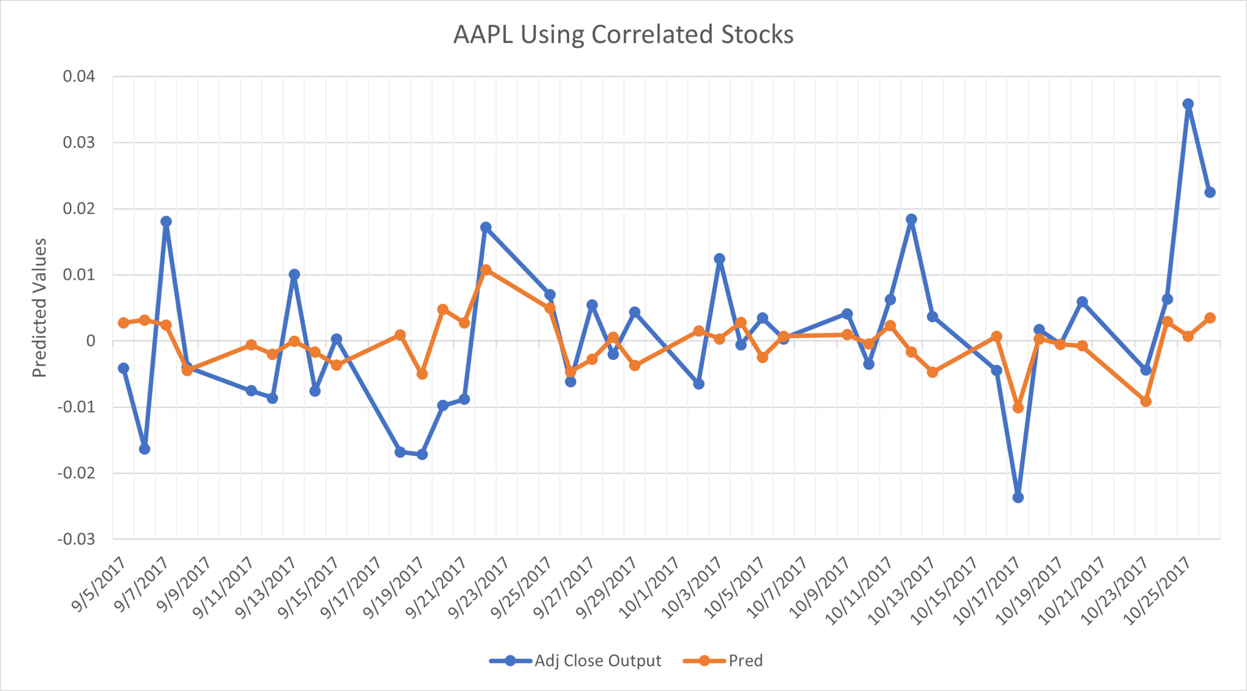

**Figure 5.2:** Apple's graph with Correlated Stocks

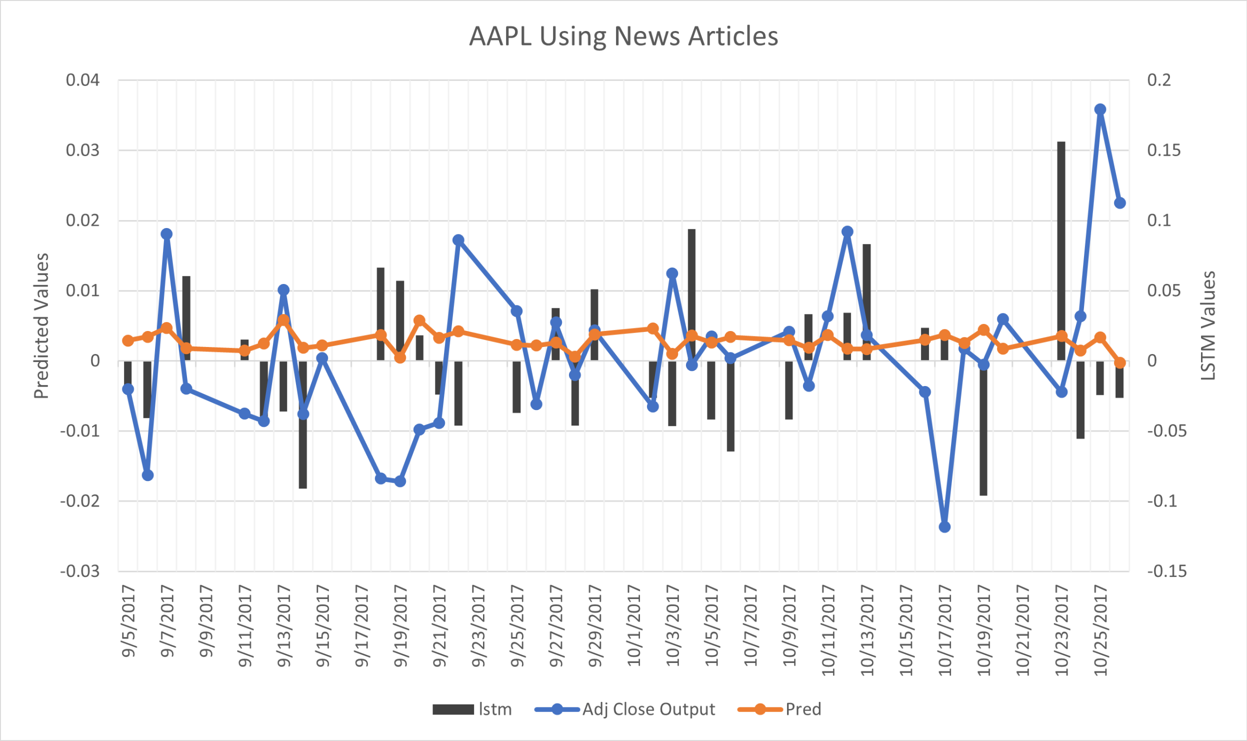

**Figure 5.3:** Apple's graph with LSTM Network Results

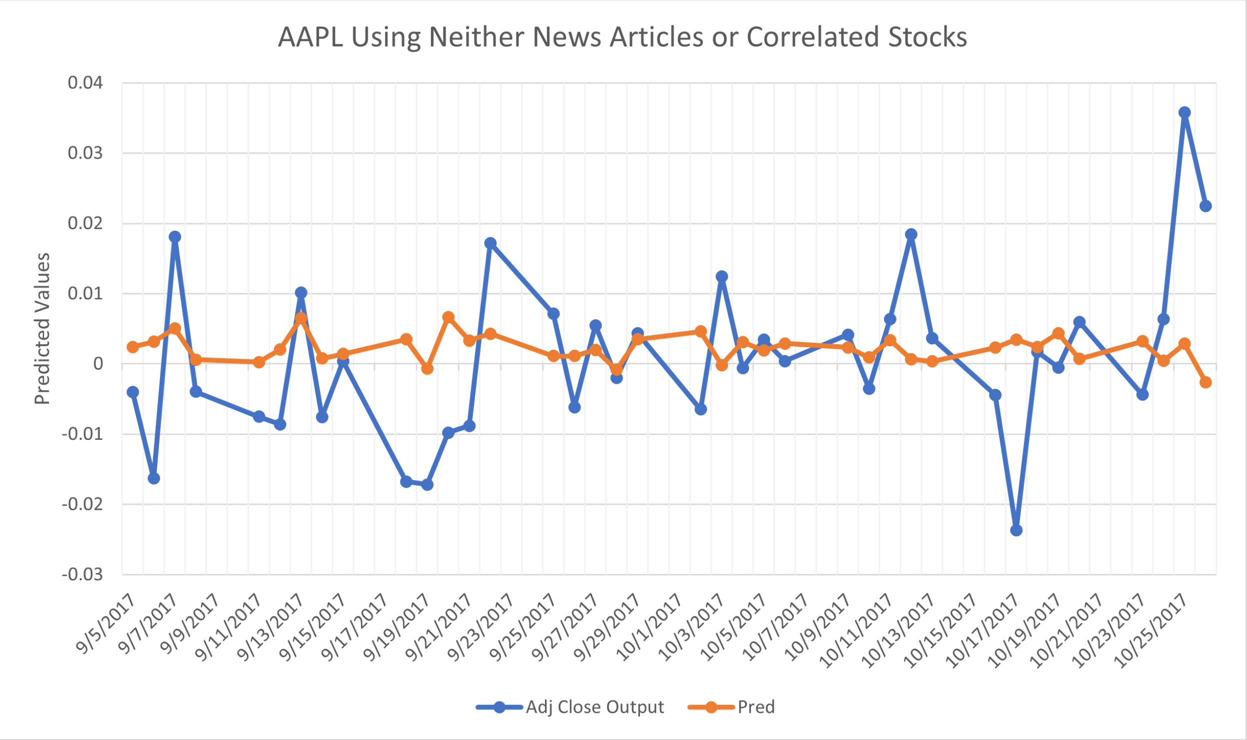

**Figure 5.4:** Apple's graph with neither Correlated Stocks or LSTM Network Results

The bar graphs showing the results from the LSTM network; the length of the bars is not necessarily an indicator of the prediction's strength. The value that we see in these graphs is the average value for the day, and this can be skewed if there are few related news articles for that day. Let us also consider a scenario where the percentage change in the stock price was within the threshold; then, the expected LSTM prediction would be zero. However, if there are some errors in the predictions, and we have a few values that are not zero, then we will end up with an average that is not zero.

It should also be noted that the dates for Apple are different from the other companies. This is because, due to certain restrictions, news articles beyond October 27th were unattainable.

### Amazon

Amazon is a relatively stable stock, evident from figures (5.5, 5.6, 5.7, 5.8). The graphs might look a little misleading because of its large peak value which is over 10% or 0.1. However, most of the values range between 1.5 and -1.5 percent, with values crossing this range only 4 times. The

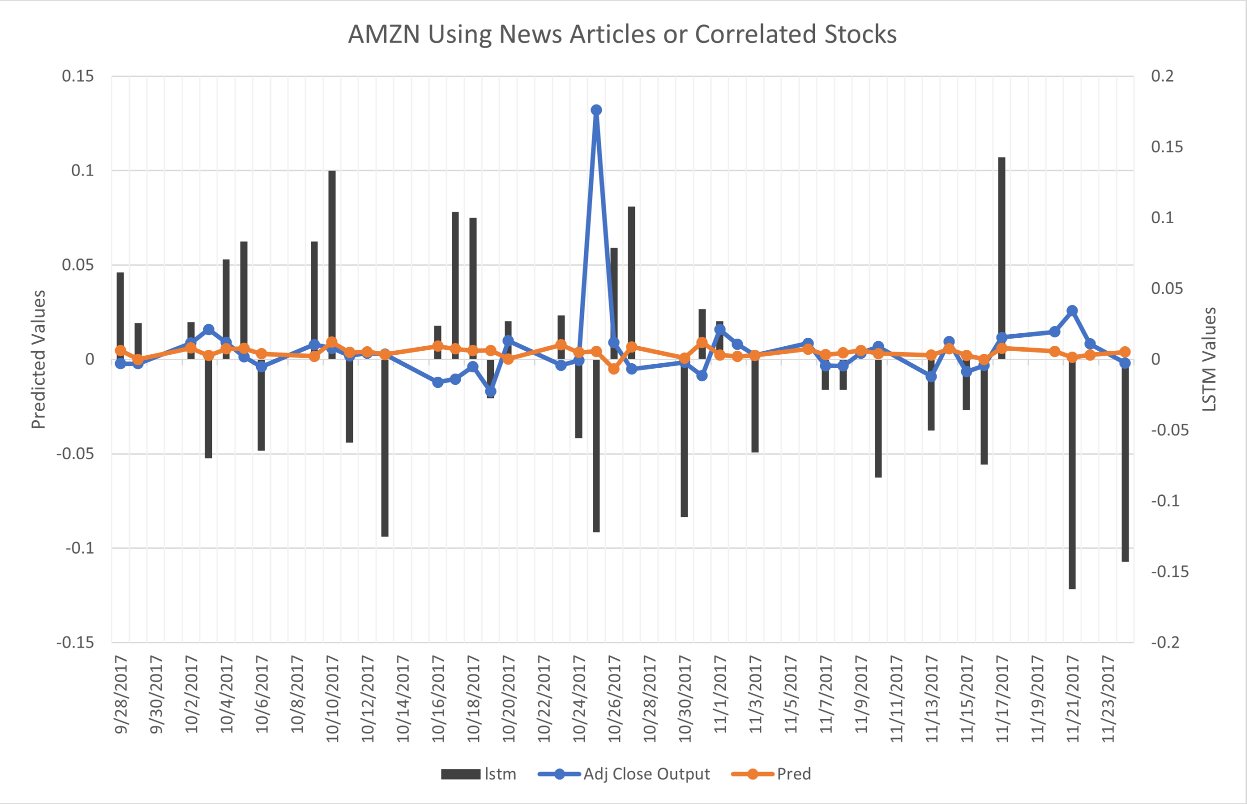

**Figure 5.5:** Amazon's graph with Complete Data

**Table 5.4:** RMSE values for Amazon on testing dataset

| Dataset | RMSE Testing | RMSE Baseline |
|---|---|---|
| News Articles and Correlated Stocks | 0.0218 | 0.0220 |
| Correlated Stocks | 0.0220 | 0.0221 |
| News Articles | 0.0221 | 0.0221 |
| Neither News Articles or Correlated Stocks | 0.0221 | 0.0221 |

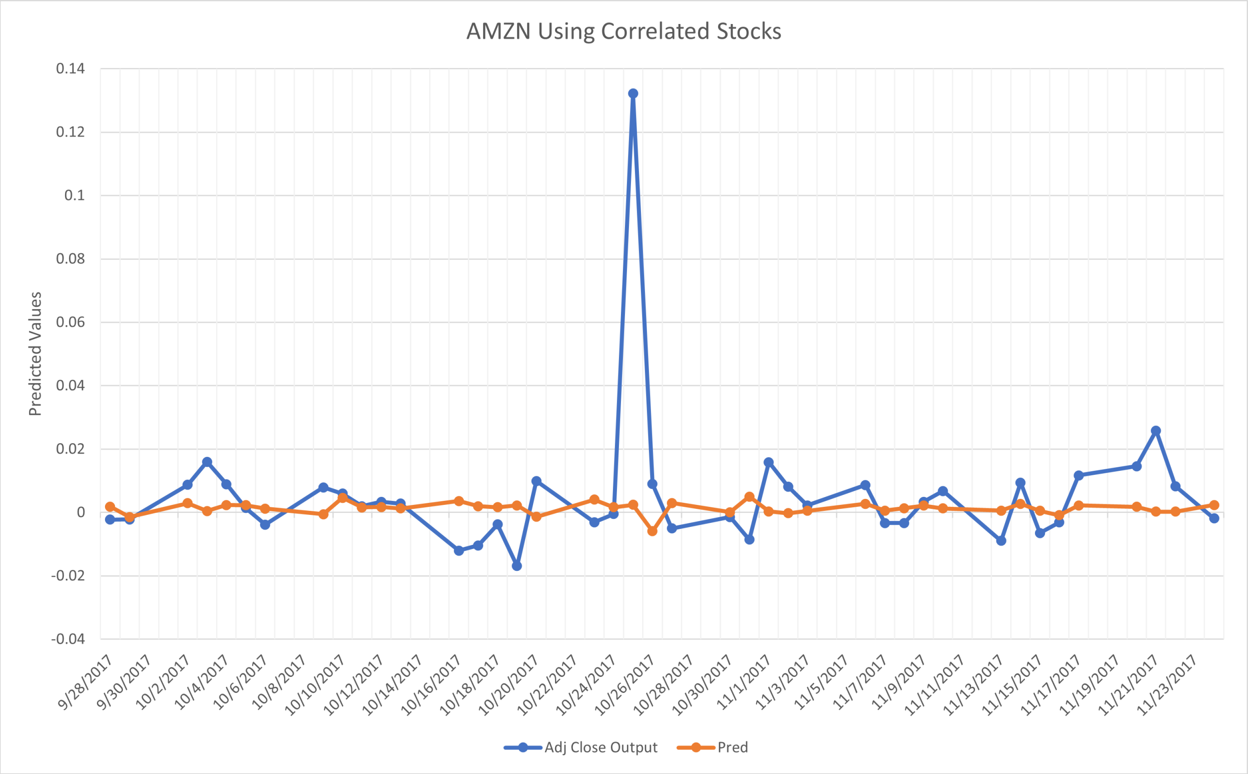

**Figure 5.6:** Amazon's graph with Correlated Stocks

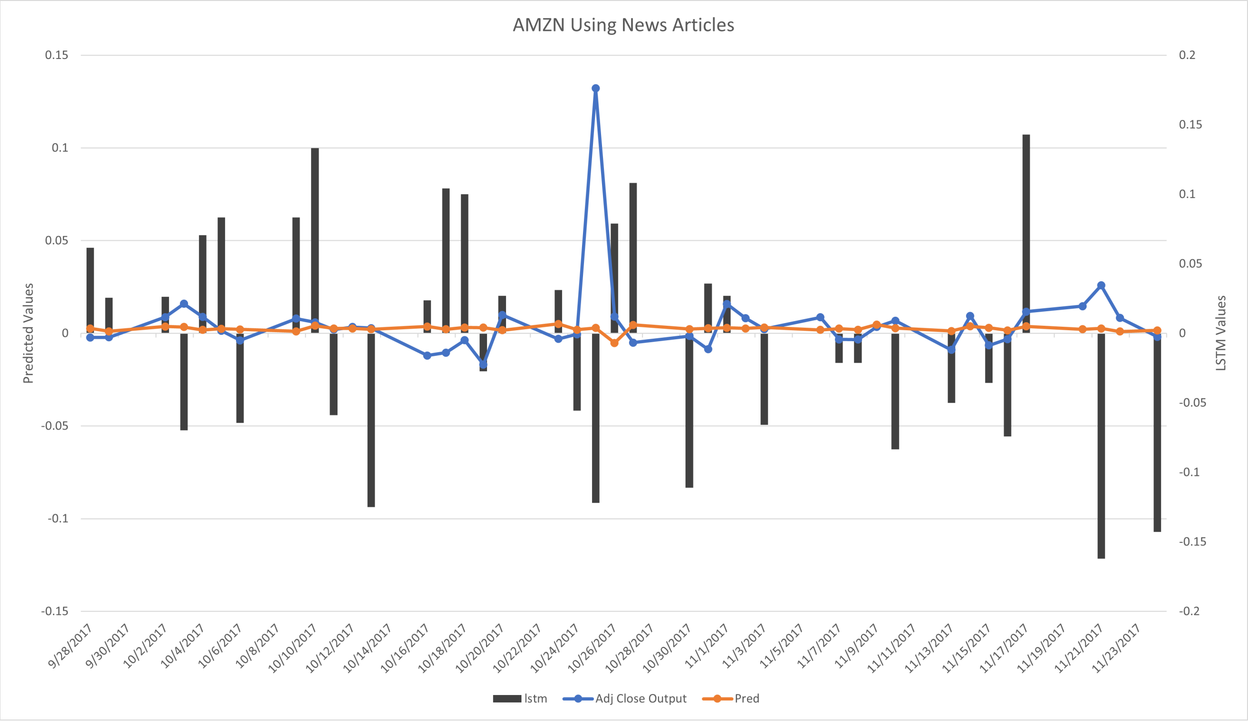

**Figure 5.7:** Amazon's graph with LSTM Network Results

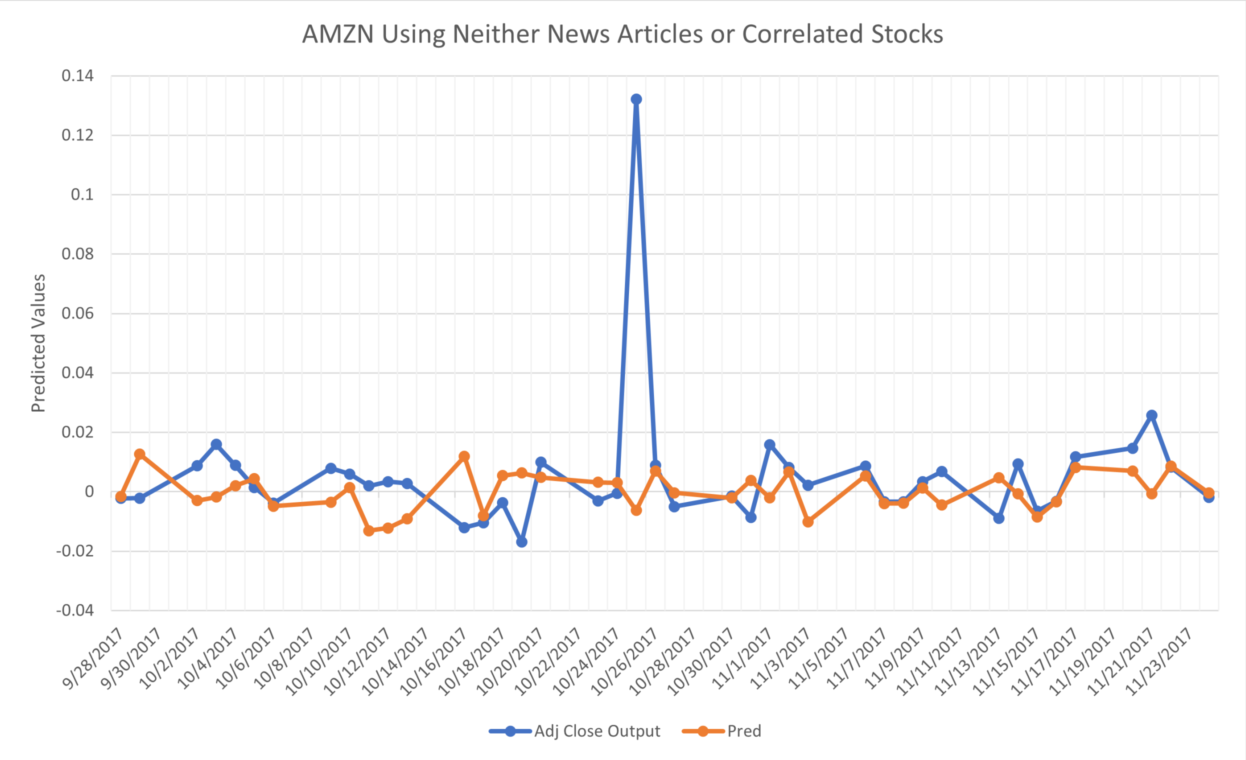

**Figure 5.8:** Amazon's graph with neither Correlated Stocks or LSTM Network Results

| Dataset | RMSE Testing | RMSE Baseline |
|---|---|---|
| News Articles and Correlated Stocks | 0.0353 | 0.0351 |
| Correlated Stocks | 0.0326 | 0.0331 |
| News Articles | 0.0339 | 0.0344 |
| Neither News Articles or Correlated Stocks | 0.0428 | 0.0440 |

**Table 5.5:** RMSE values for AMD on testing dataset

four times are on 10/19, 10/25, 11/1 and 11/21. It is interesting to note that the LSTM network predicted this movement correctly 2 out of the 4 times.

The news articles' results do not seem to have much impact on Amazon's stock values. Table 5.4, however, shows that the graphs are quite accurate.

### AMD

AMD had the least accuracy on the test dataset for the LSTM network. This could be because of the high volatility which can be seen in figures (5.9, 5.10, 5.11, 5.12). Unlike most of the other companies, AMD's stock change values seem to lie between mostly between 3.0 and -3.0 percent.

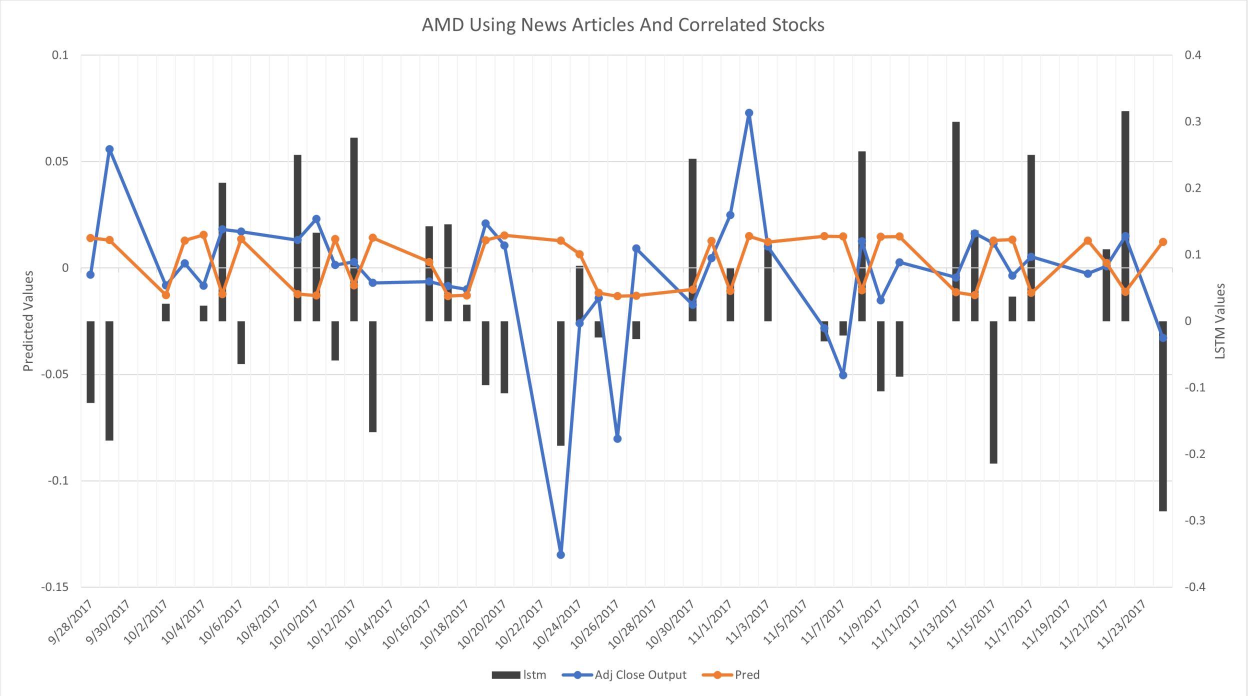

**Figure 5.9:** AMD's graph with Complete Data

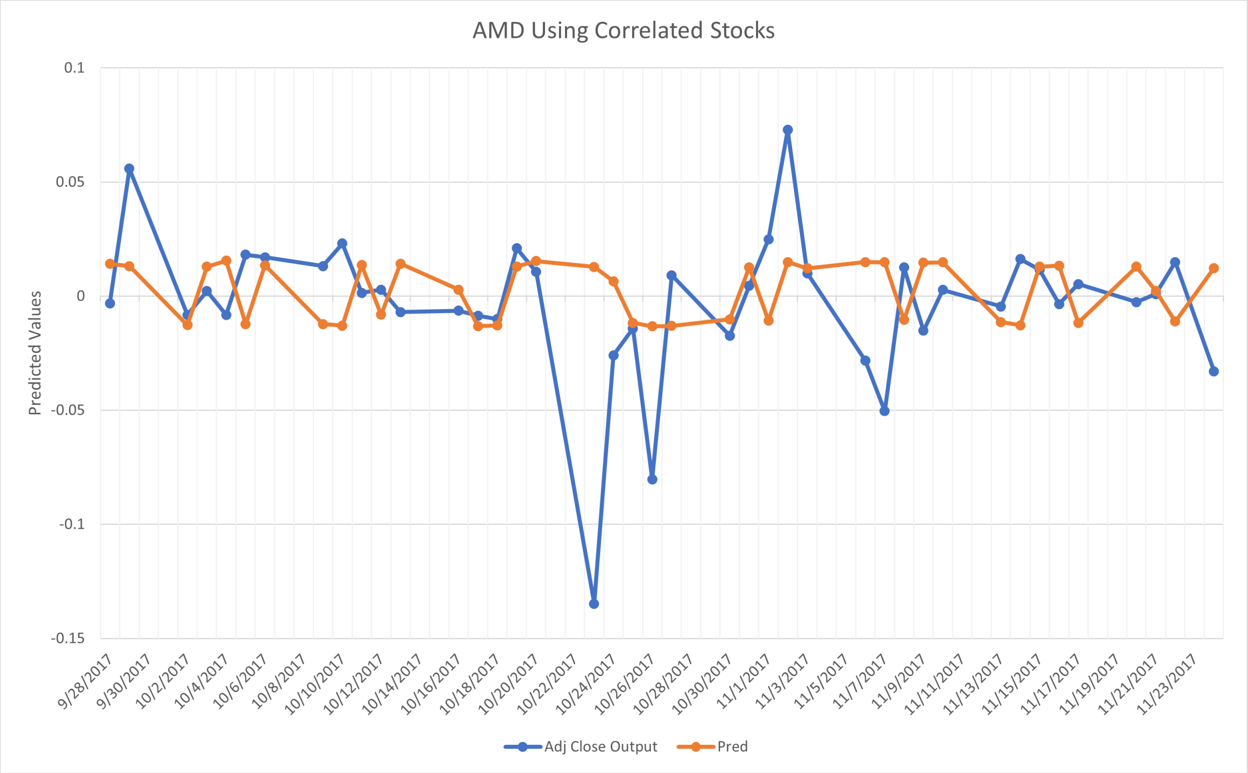

**Figure 5.10:** AMD's graph with Correlated Stocks

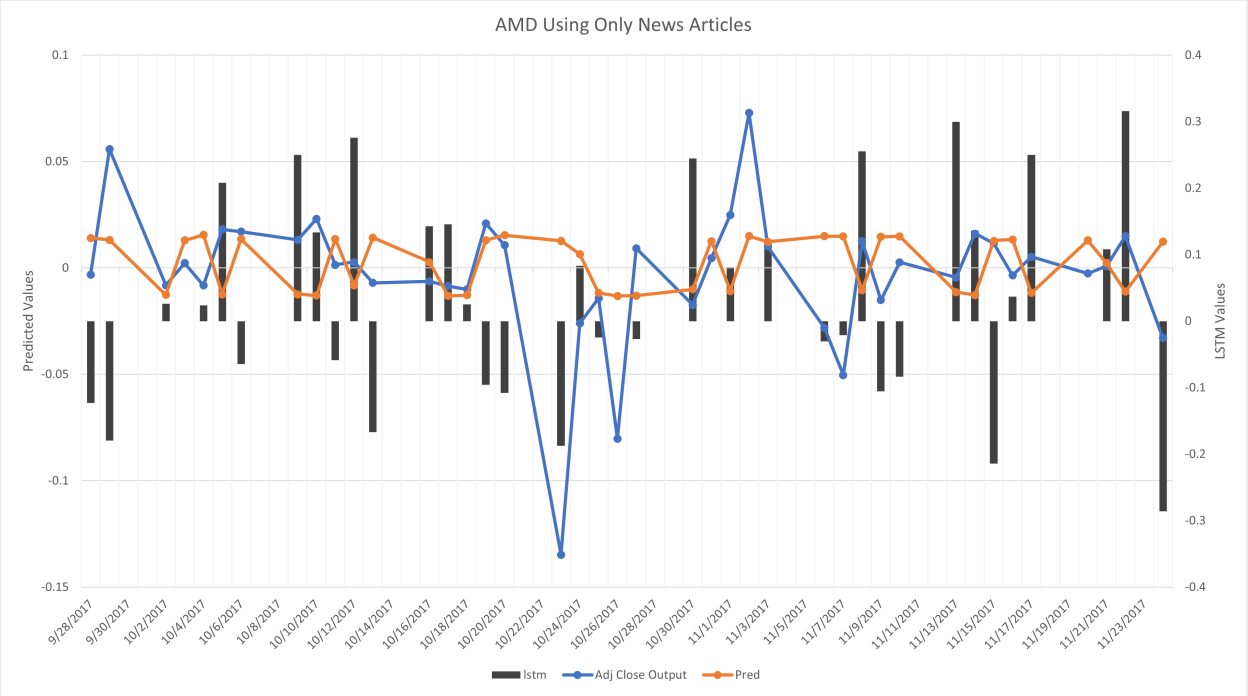

**Figure 5.11:** AMD's graph with LSTM Network Results

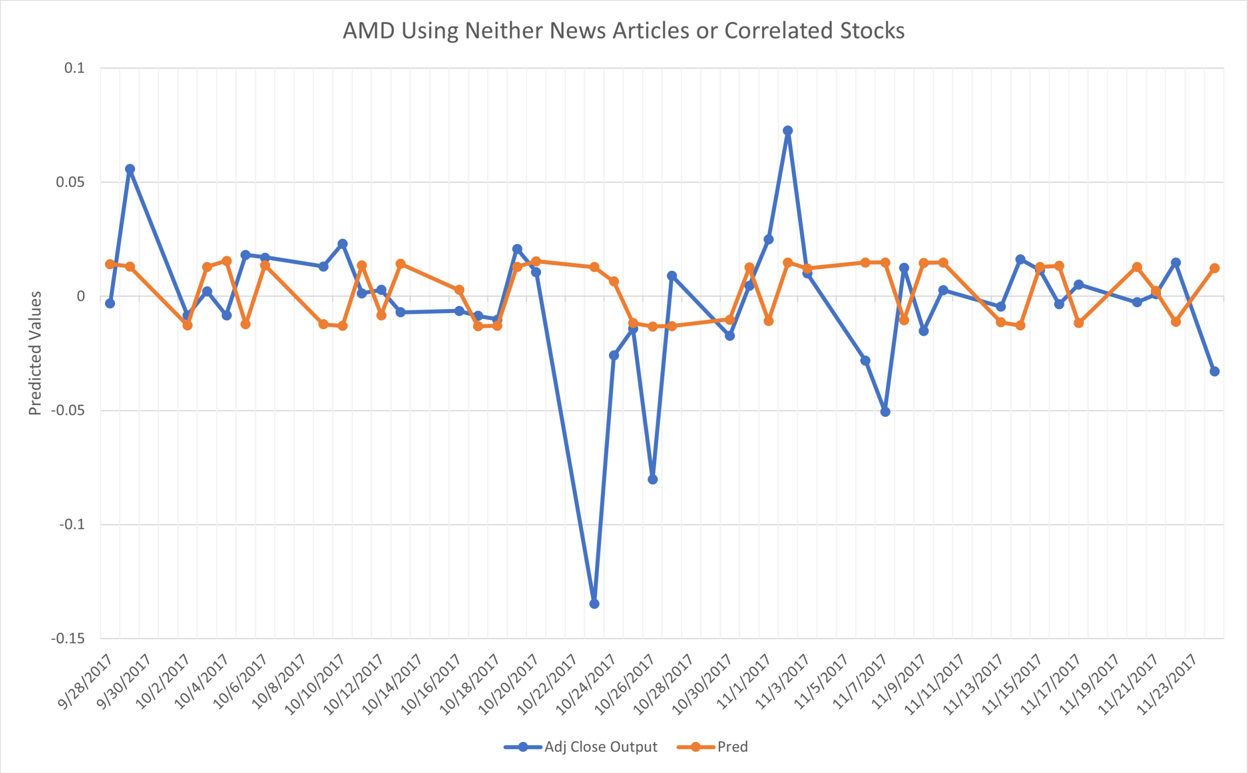

**Figure 5.12:** AMD's graph with neither Correlated Stocks or LSTM Network Results

**Table 5.6:** RMSE values for Google on testing dataset

| Dataset | RMSE Testing | RMSE Baseline |
|---|---|---|
| News Articles and Correlated Stocks | 0.0085 | 0.0088 |
| Correlated Stocks | 0.0082 | 0.0086 |
| News Articles | 0.0082 | 0.0087 |
| Neither News Articles or Correlated Stocks | 0.0085 | 0.0085 |

Even with this wide range there are 6 values outside this range. This seems to indicate that various outside factors have a significant impact on AMD's stock value, which theoretically should have been captured in the news. However, the LSTM network did not train as efficiently on this data which can be seen in the table 5.5.

### Google

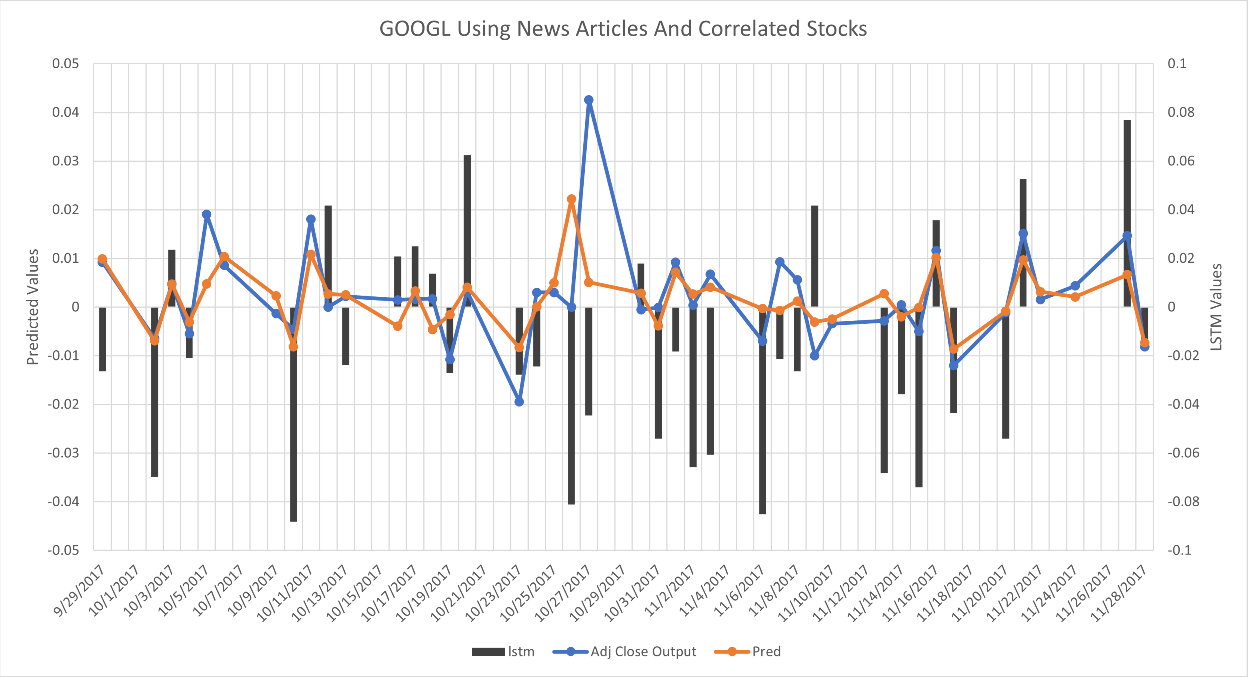

**Figure 5.13:** Google's graph with Complete Data

Google had the second-highest accuracy on the test dataset for the LSTM network. This is mainly because Google, just like Microsoft and Amazon, is also a very stable stock. The network

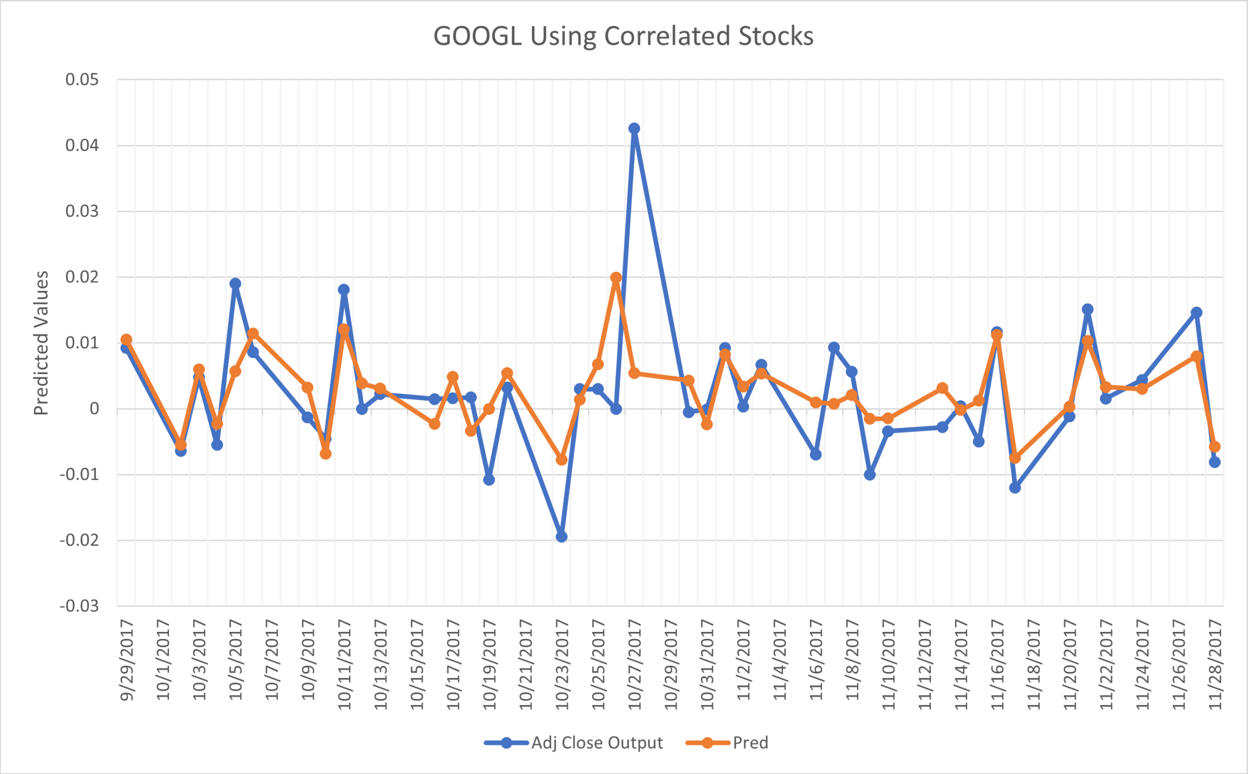

**Figure 5.14:** Google's graph with Correlated Stocks

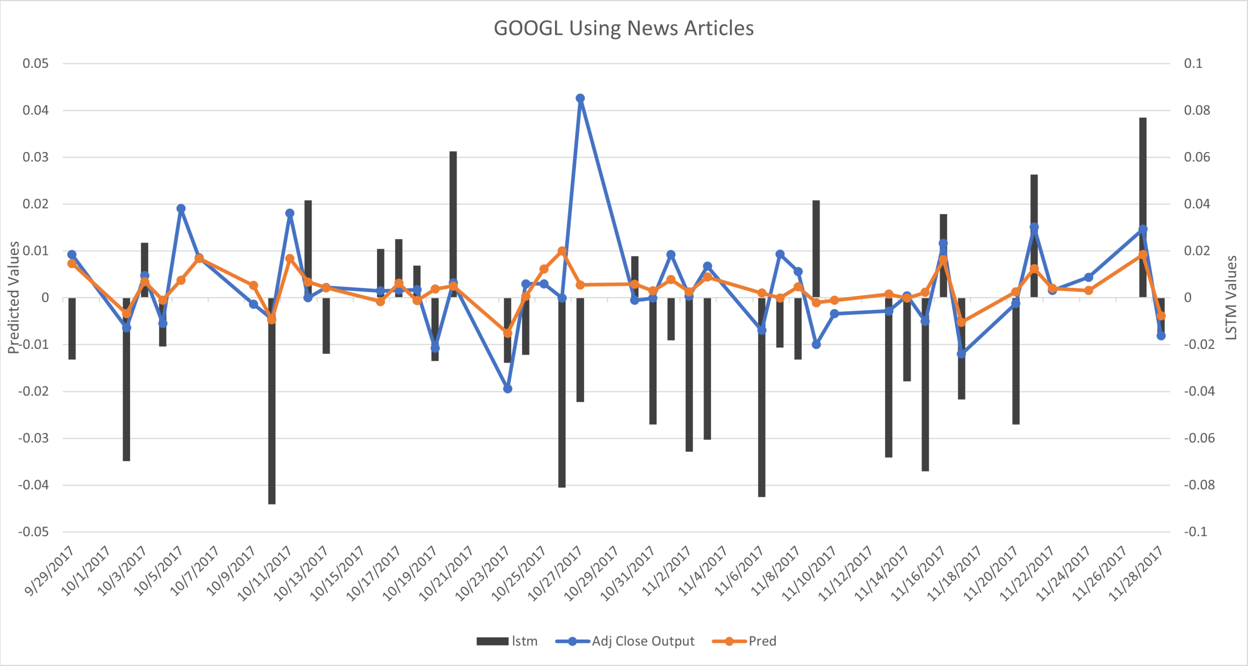

**Figure 5.15:** Google's graph with LSTM Network Results

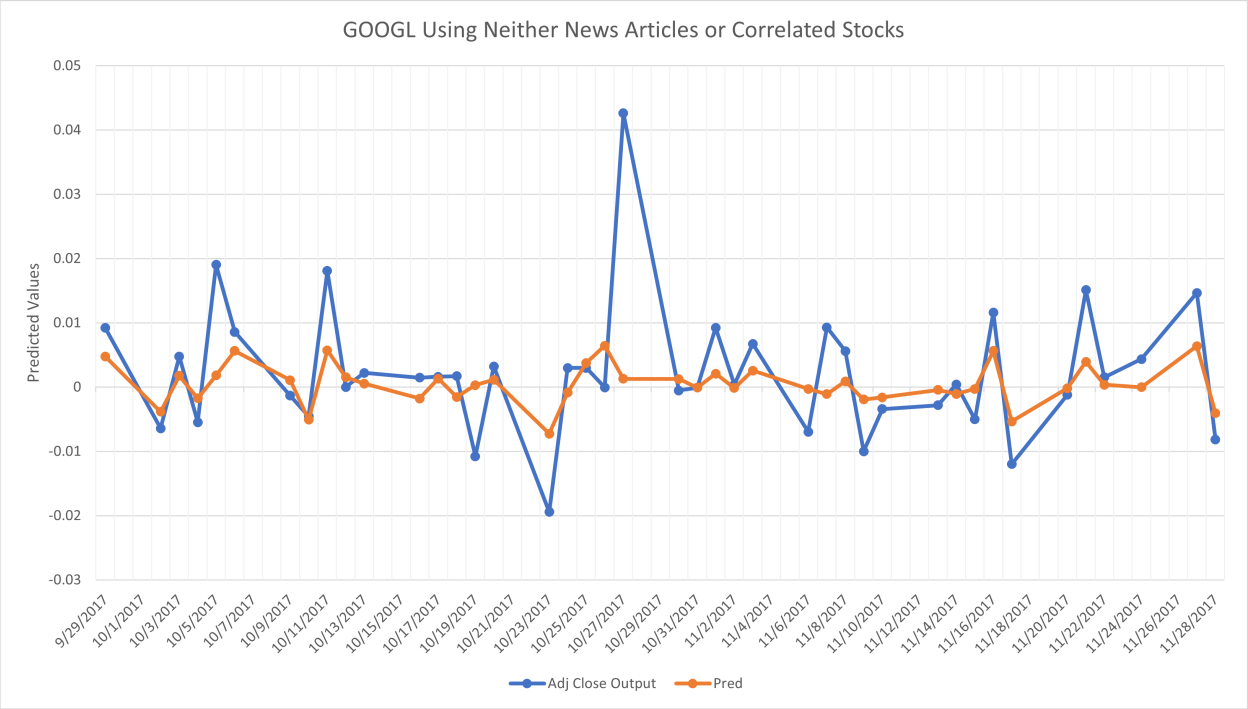

**Figure 5.16:** Google's graph with neither Correlated Stocks or LSTM Network Results

**Table 5.7:** RMSE values for Microsoft on testing dataset

| Dataset | RMSE Testing | RMSE Baseline |
|---|---|---|
| News Articles and Correlated Stocks | 0.0116 | 0.0118 |
| Correlated Stocks | 0.0117 | 0.0118 |
| News Articles | 0.0114 | 0.0116 |
| Neither News Articles or Correlated Stocks | 0.0116 | 0.0116 |

seems to have picked up the movement of the stock values quite accurately. We can see that the RMSE values in table 5.6 do indicate that the predicted values are still quite close to the real values.

### Microsoft

Microsoft seems to be the most stable stock among the companies considered in this thesis. This can be seen from the figures (5.17, 5.18, 5.19, 5.20). The flat nature of the graph means that there is not much for the network to learn. The higher the stock stability, the easier it is to get very high accuracy in terms of its stock value predictions, as can be seen from table 5.7. This was also true with Google stock values.

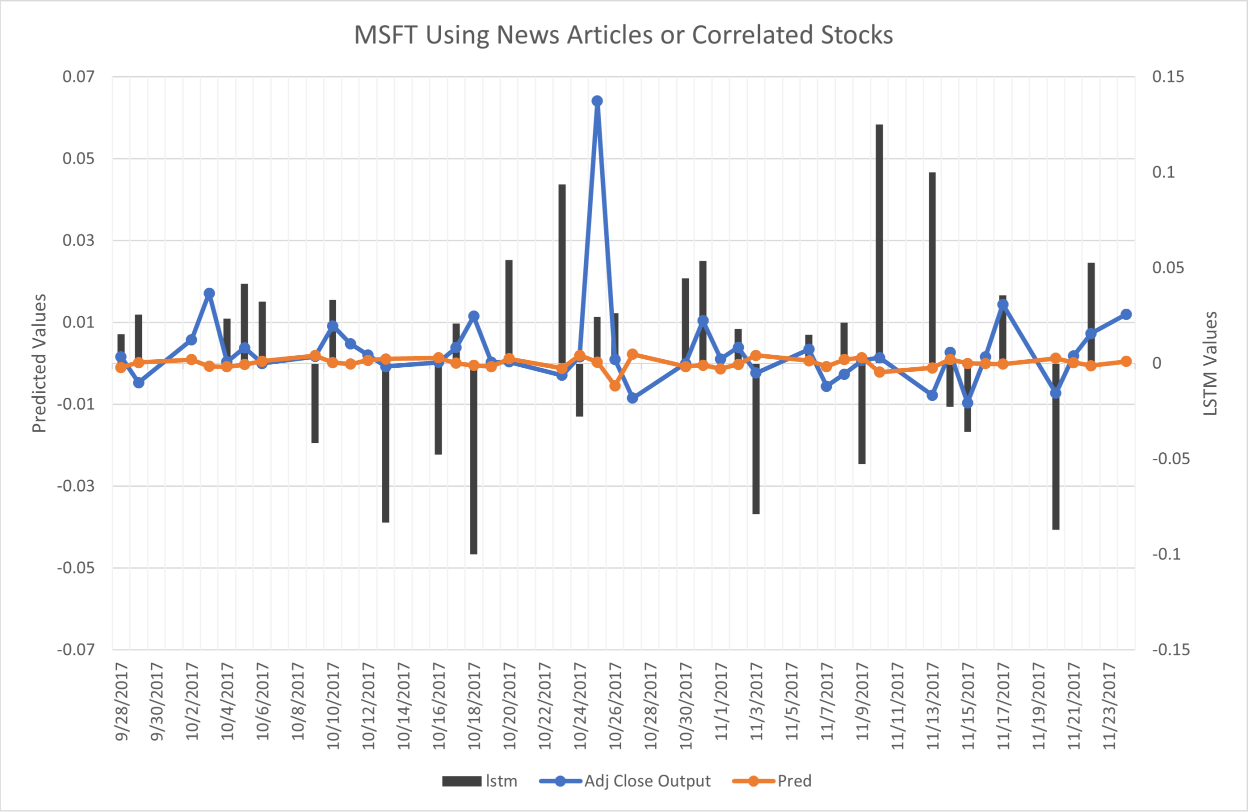

**Figure 5.17:** Microsoft's graph with Complete Data

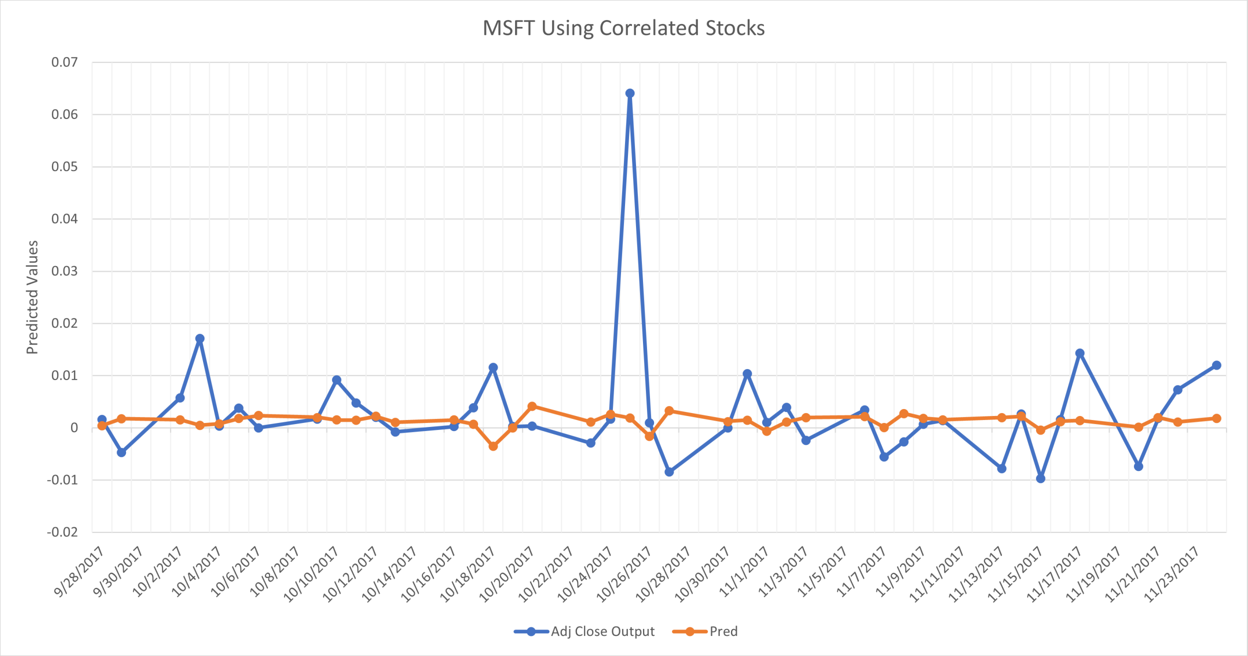

**Figure 5.18:** Microsoft's graph with Correlated Stocks

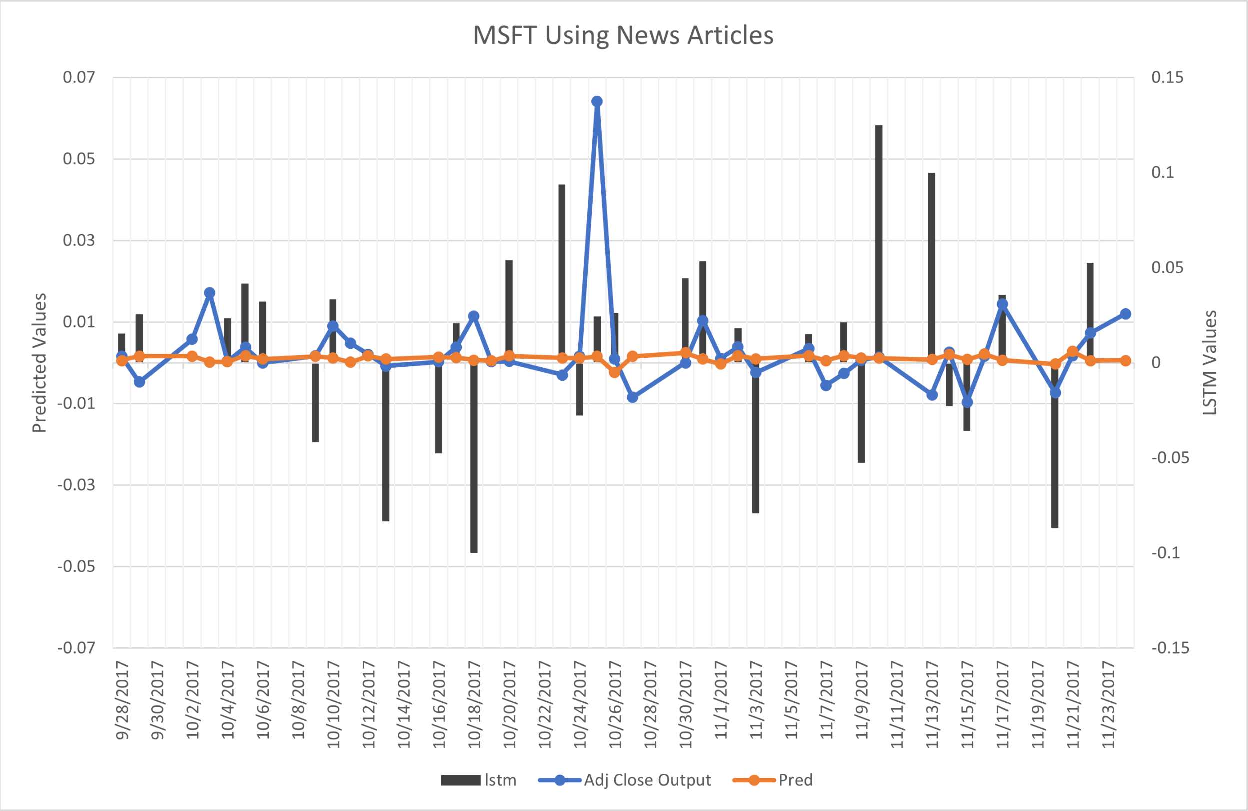

**Figure 5.19:** Microsoft's graph with LSTM Network Results

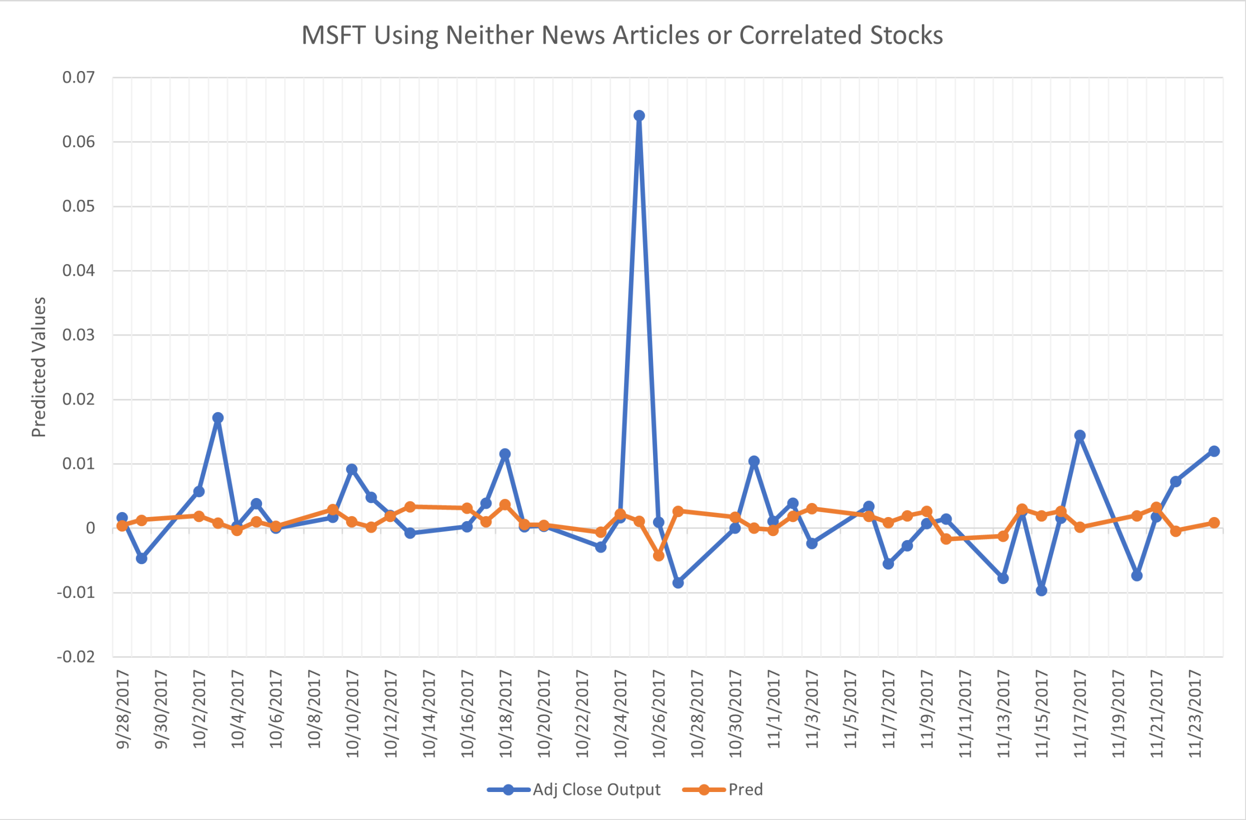

**Figure 5.20:** Microsoft's graph with neither Correlated Stocks or LSTM Network Results

**Table 5.8:** RMSE values for Netflix on testing dataset

| Dataset | RMSE Testing | RMSE Baseline |
|---|---|---|
| News Articles and Correlated Stocks | 0.0165 | 0.0167 |
| Correlated Stocks | 0.0165 | 0.0165 |
| News Articles | 0.0164 | 0.0165 |
| Neither News Articles or Correlated Stocks | 0.0162 | 0.0164 |

### Netflix

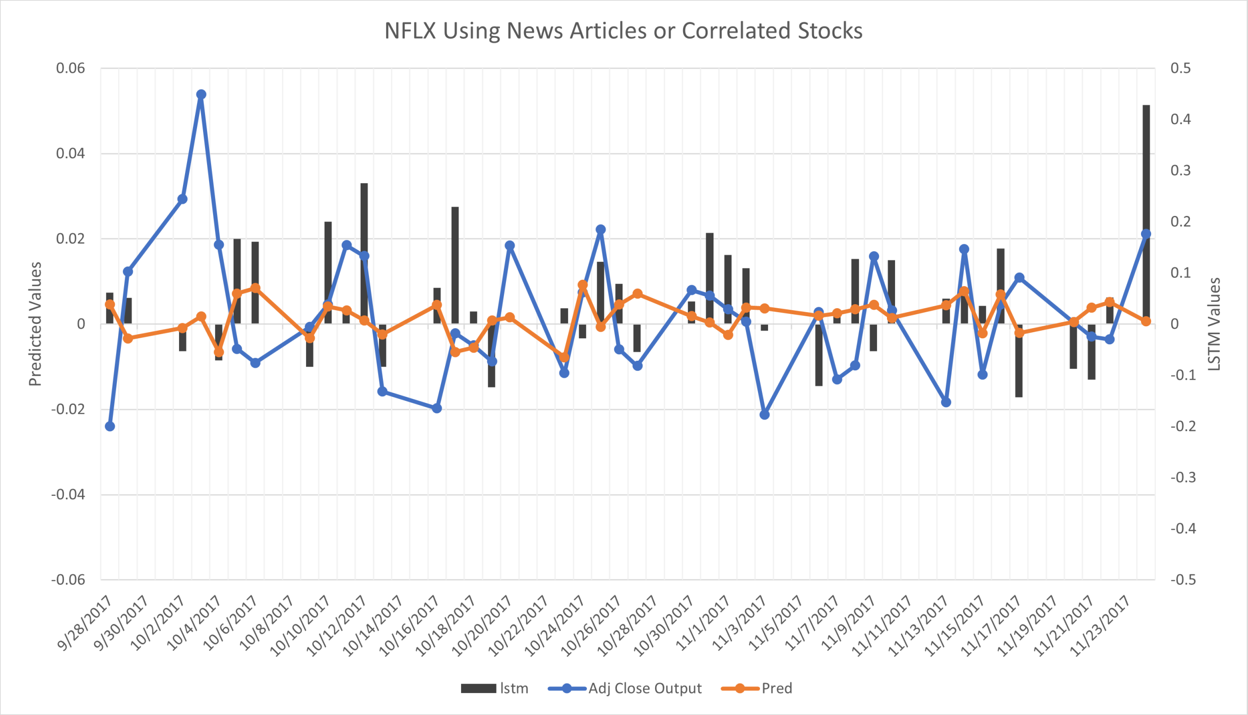

**Figure 5.21:** Netflix's graph with Complete Data

Netflix is one of the most volatile stocks in the set of companies used in the thesis. As shown in figures (5.21, 5.22, 5.23, 5.24), the meandering nature of the graph indicates this unstable structure of the company's value. All graphs appear to show similar predictions. This could be because Netflix stock is not depending too much on correlated stocks or news articles. The network seems to perform worst with Netflix and the RMSE values in table 5.8 indicate the same. According to the RMSE values, Netflix has the worst performance.

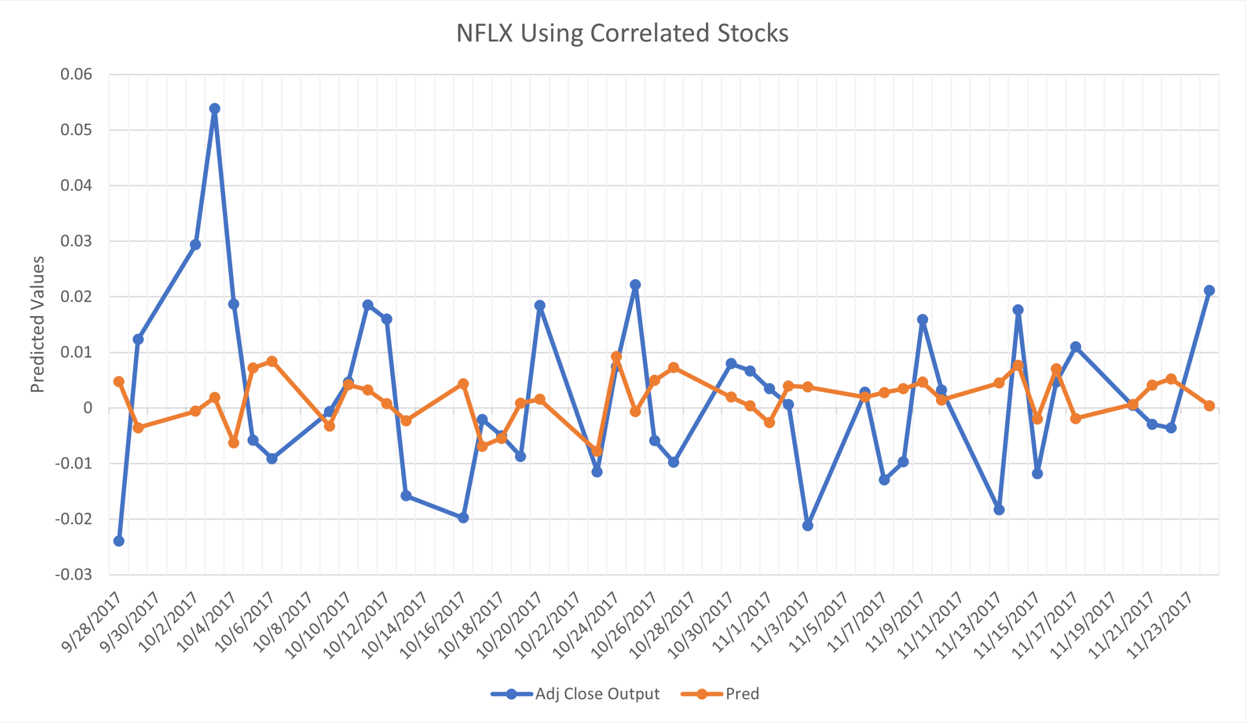

**Figure 5.22:** Netflix's graph with Correlated Stocks

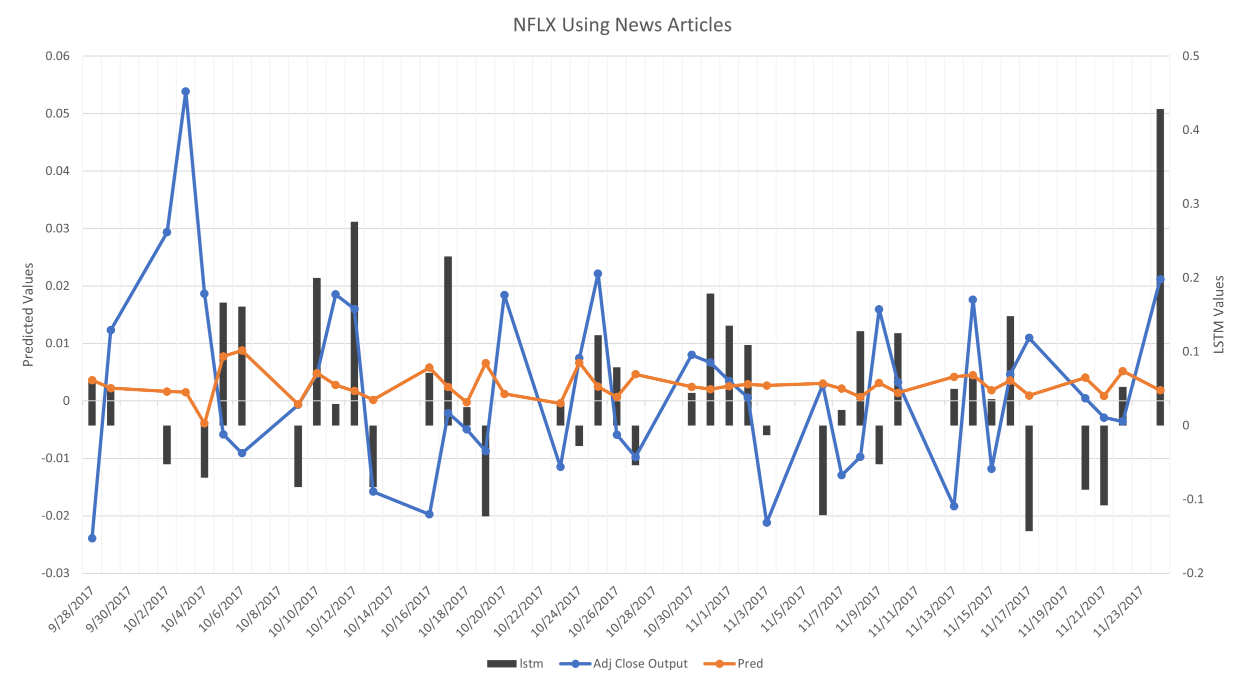

**Figure 5.23:** Netflix's graph with LSTM Network Results

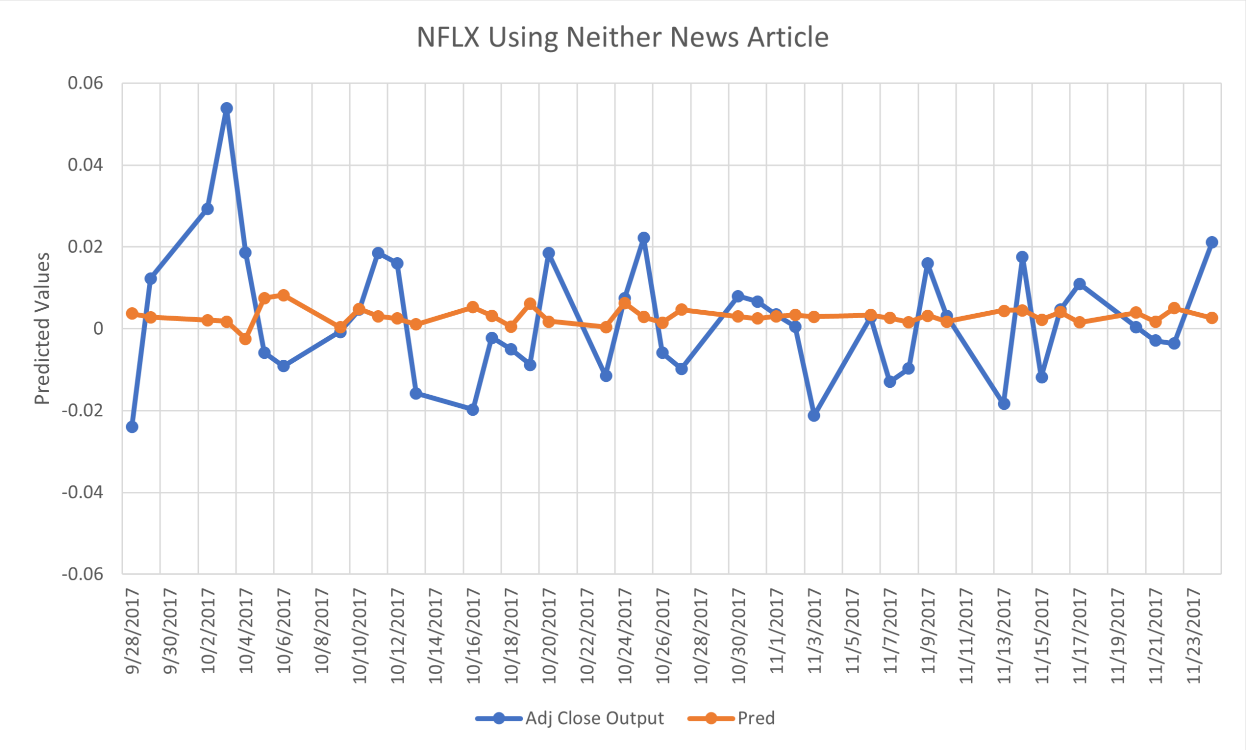

**Figure 5.24:** Netflix's graph with neither Correlated Stocks or LSTM Network Results

**Table 5.9:** RMSE values for Nvidia on testing dataset

| Dataset | RMSE Testing | RMSE Baseline |
|---|---|---|
| News Articles and Correlated Stocks | 0.0153 | 0.0155 |
| Correlated Stocks | 0.0145 | 0.0144 |
| News Articles | 0.0144 | 0.0149 |
| Neither News Articles or Correlated Stocks | 0.0148 | 0.0148 |

### Nvidia

Nvidia is another stock that had only about 62% accuracy from the LSTM network. The figures (5.25, 5.26, 5.27, 5.28) indicate that none of the graphs are incredibly accurate. Nvidia is one of the more volatile stocks, and this could explain the lack of accurate results.

The graphs indicate that there is no significant difference in terms of output accuracy. The RMSE values also indicate that the Deep Neural Network seem to be performing slightly better than the baseline model in most cases. This thesis's main idea was to see if the algorithm can pick up the significant changes in the stock values. While the algorithm seems to maintain the general trends, it did not pick up the peaks for any of the companies on regular bases. However, the overall

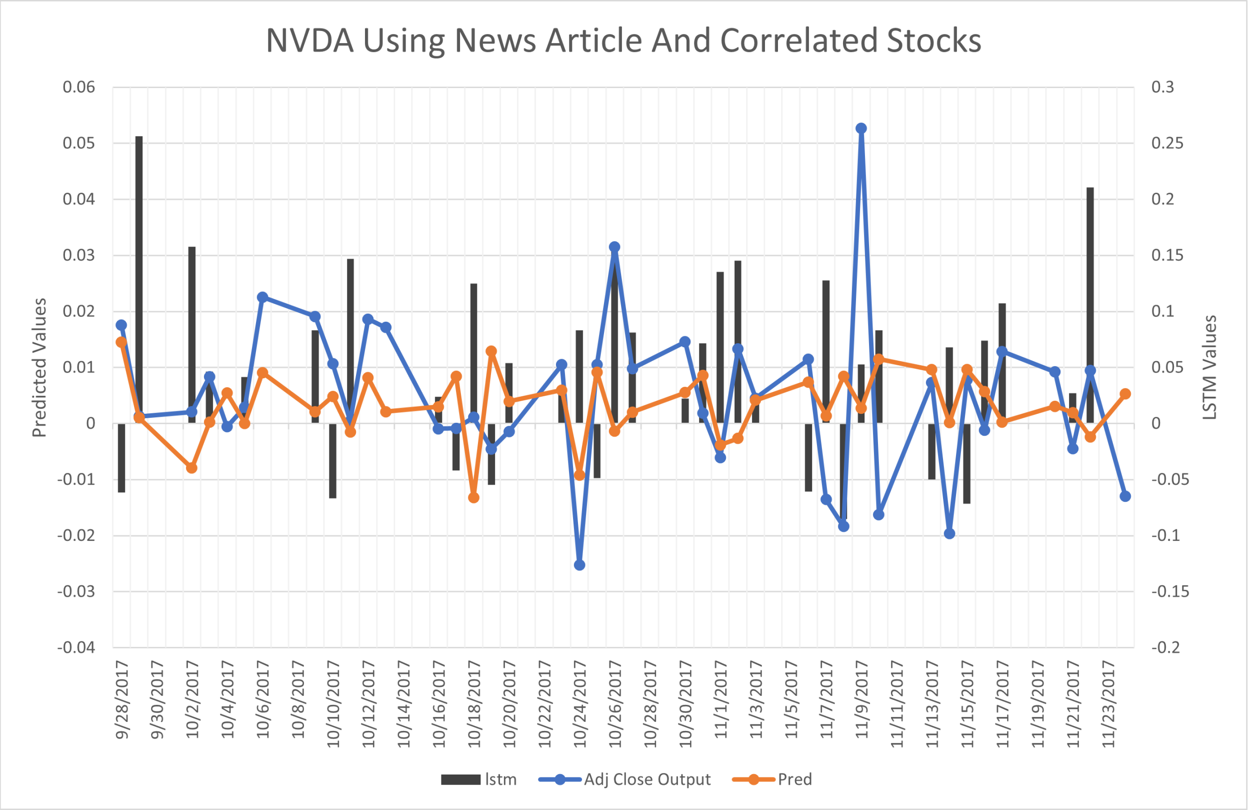

**Figure 5.25:** Nvidia’s graph with Complete Data

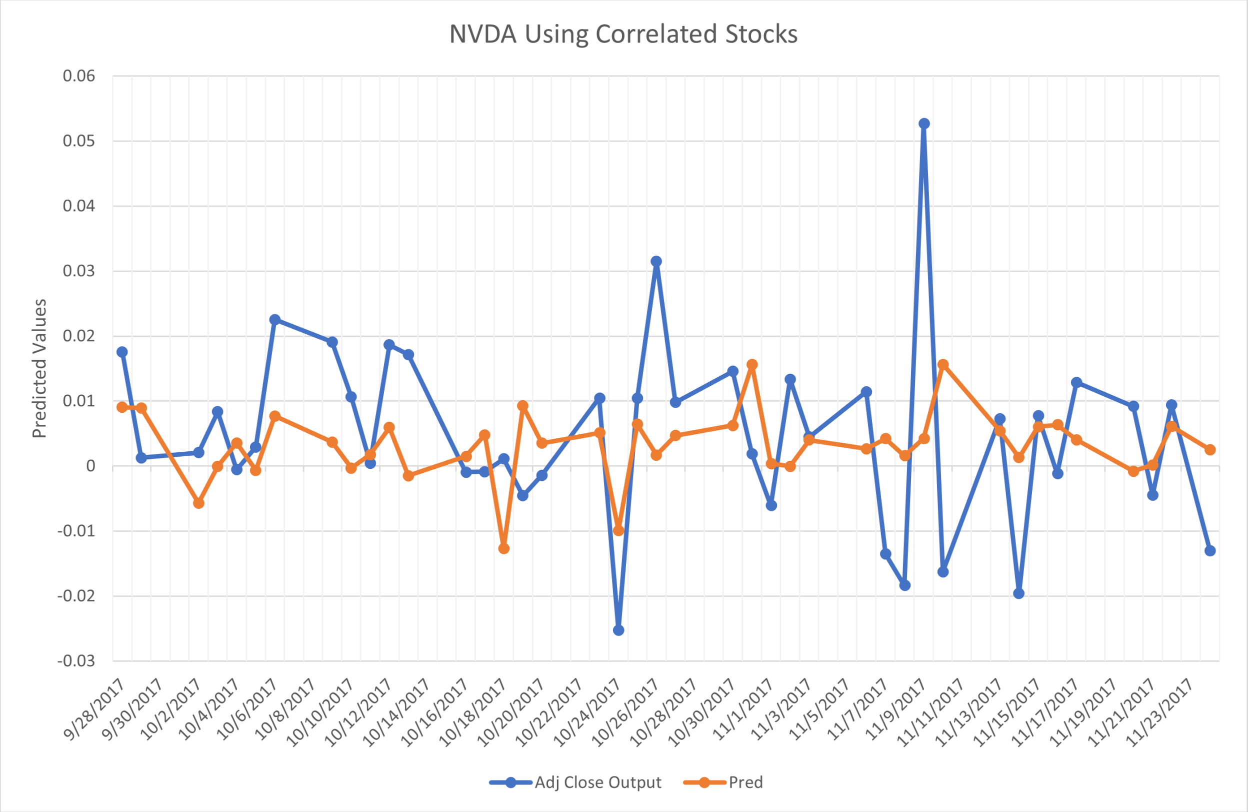

**Figure 5.26:** Nvidia's graph with Correlated Stocks

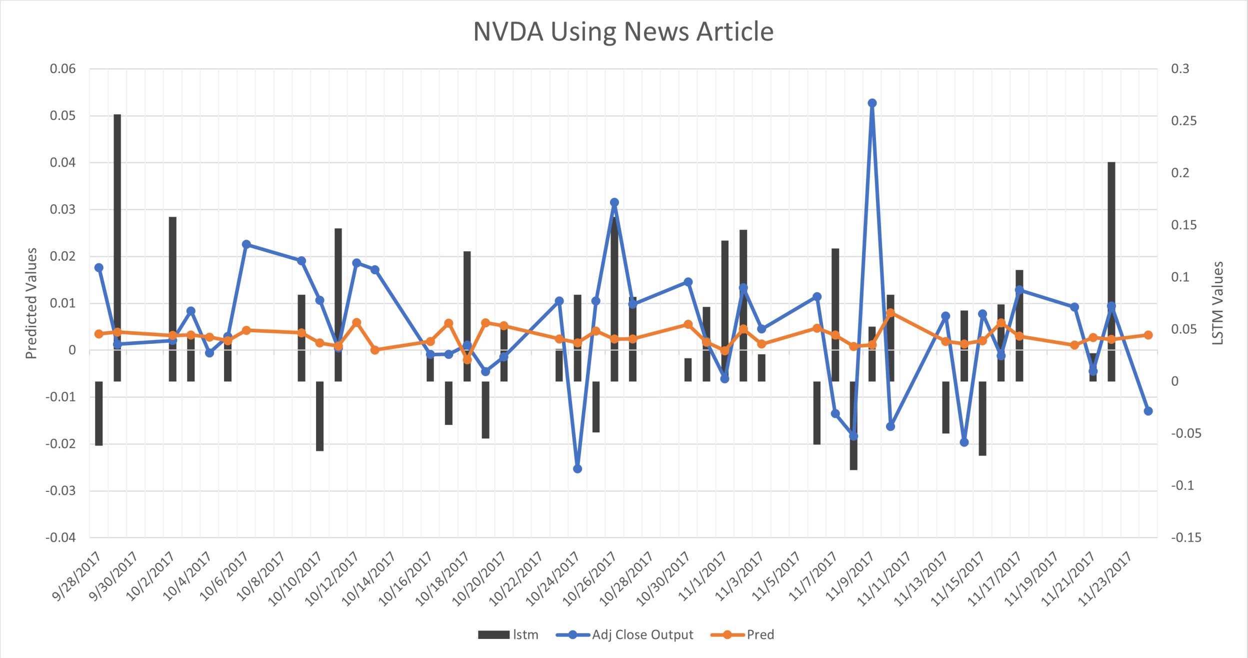

**Figure 5.27:** Nvidia's graph with LSTM Network Results

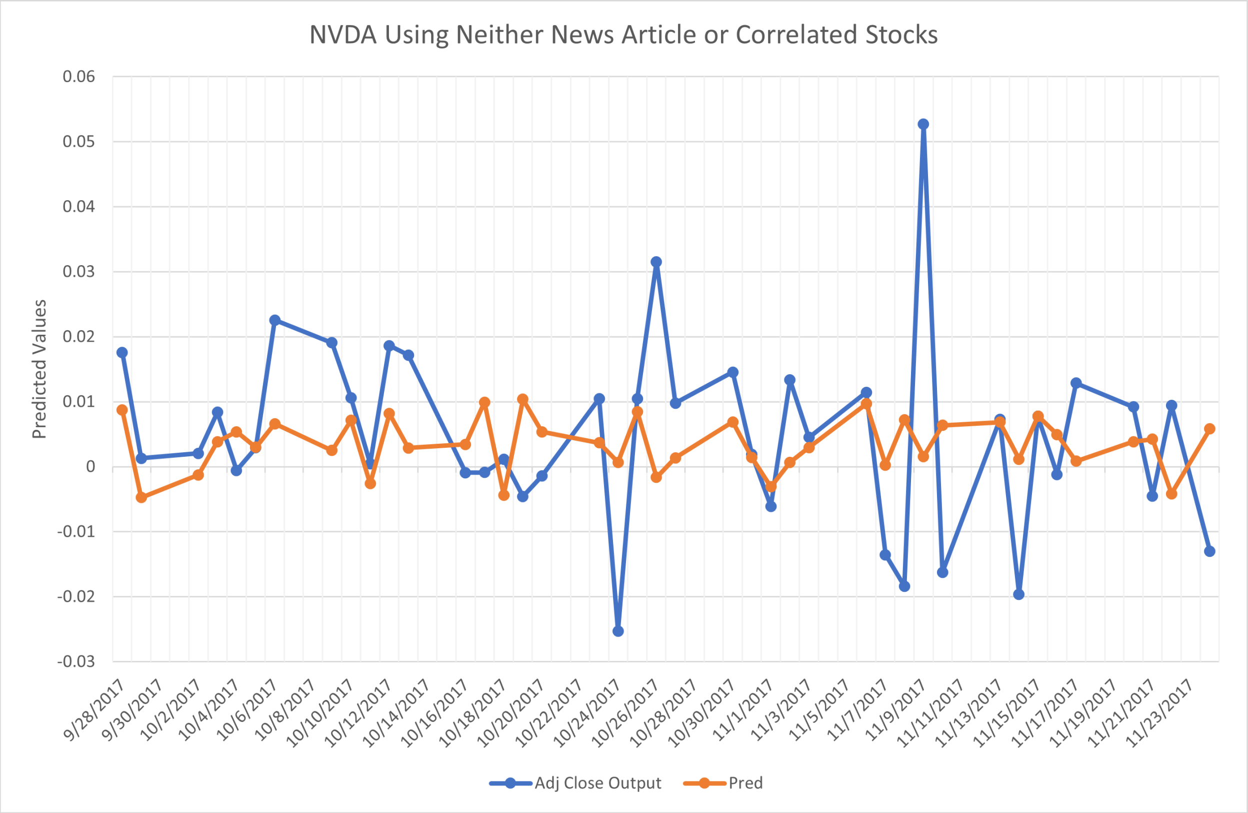

**Figure 5.28:** Nvidia's graph with neither Correlated Stocks or LSTM Network Results

accuracy of all the graphs for all the companies seems very high, and the specific use of news articles does not seem to improve the output. It is also very interesting that the since the RMSE values of the baseline model is so similar to the ones from the Deep Neural Network, the simpler regression model may have been enough for this part of the thesis.

Another important to note is that the length of the black bars for the LSTM network does not necessarily indicate the strength of the LSTM network's predictions of the stock movement, even though that was the intent behind the usage of the data. There are many days where the number of articles regarding a specific stock is very few, and if the lstm networks get the prediction wrong, then the bar can be really high.

# Chapter 6

# Conclusion and Future Work

Stock market prediction is a very sought after field in the modern world. Many researchers have built many models to try to achieve very high accuracy on stock price predictions. Since the stock market is affected by various factors, most researchers focus on different features to make a prediction, but only a few try to incorporate more than one such factor. A more commonly used approach for stock predictions is using a trend-based approach, which uses a company's historical stock prices to predict its future value. This is a tried and trusted approach, mainly because investors or traders as a whole have the most impact on the value of a company, and if everyone uses the same approach, which gives the same result, then people follow this approach, thereby giving the expected result. Hence the investors themselves are the cause of that final value of a company. The general trends are easy enough to predict, but the part that has always bewildered the researchers is trying to predict the sudden change in the value of a company due to various factors like important news about the company or change in the general stock market. Neither of these factors is used in a trend-based analysis. Hence that model will never predict the said changes in the stock value of a company. This thesis was an attempt on a different approach that could have potentially solved most of the stock market issues. There were some positives, but a lot more data is needed to make it a more compelling algorithm. There was a general assumption that using time-based data instead of sentiment-based data for the news articles might give a better result. Unfortunately, this was not the case due to certain factors which are discussed below.

This thesis assumed that any factor that can potentially affect a company's stock value would be made public as part of the news on some financial website, but the part that was not possible to incorporate into the approach was the time period. Some articles are not published instantaneously, or sometimes they may be published immediately but will not reach the mainstream audience quickly enough. In the first case, the change in the stock value does not coincide with the related news articles, which can then be wrongly predicted by the LSTM network. In the second case also the news article and the related stock change do not coincide, causing errors during the training stage, but in this case, the stocks are affected a few days after the news is published.

The amount of data needed to train an LSTM network of this size successfully is huge, which in turn meant that data over a period of five years was needed. Over such a significant amount of time, many companies change their priorities, and there can be a new competition, or some old competition no longer exists. Hence the network would have to learn many things which change over time, making it very difficult for any machine learning algorithm. Despite this, the LSTM network performed well, which is a huge positive. Another issue was that different articles are of different length. To maintain consistency, I assumed that the length of the first paragraph is about 50 words and this length was used size of each article.

All the stocks seem to perform better under different conditions, and the difference is not that great. I took four different conditions: with news, with correlated stocks, with both the news and correlated stocks and without either of them. Using news seemed to give the highest accuracy in two cases each; using only correlated stocks seems to give the highest accuracy in two instances, and using both appeared to provide the highest accuracy in two instances. Looking at the result graphs, the more volatile a stock, the lower the quality of the final predictions. This is expected since the amount of data available for training is less. A fascinating discovery was that Nvidia seems to get the best accuracy when neither news nor correlated stocks are used. Nvidia is interesting in the fact that AMD is their direct rivals, so one can expect that it performs in a similar pattern as AMD, but this does not seem the case. Nvidia is relatively more stable than AMD, mainly because Nvidia's market share has been relatively stable despite AMD's improved product quality. The baseline model also seem to perform almost as good as the Deep Neural Network and the fact that this model did not pick up any of the major change in stock prices, a simpler regression model might have a performed just as accurately.

We know that news plays an integral part in stock values, but understanding when and by how much has always been an issue for any researcher to answer. In this thesis, using the stock changes itself was used to train the LSTM network so that the network can learn when the news is relevant to the changes in stock values. To some extent, this approach seems to work mostly when used alongside the correlated stocks, as can be seen with Amazon and Netflix. This result was expected, and hence the correlated stocks were used in the thesis. News can impact the stocks, but all news does not impact the stocks all the time. Despite all the issues with this approach, there is much potential in using LSTM for trying to understand when news articles can impact stock values and when they do not.

The primary goal of this thesis was to establish the connection between the news and the stock movement. Typically when researchers try to establish such a connection, they use the sentiment analysis to do so. In this thesis, the stock movement was used as the target variable instead of the sentiment. The idea being the network might be able to know which news impact the stocks. The LSTM network results show that it is possible to make a reasonable classification for which news can impact the stocks, but this does not translate well when these results were used for stock predictions. Potential reasons for this could be the lack of data since only about nine months of data was used for stock price predictions. There is no practical way to increase the number of examples, as using older data may not be too relevant for the latest market.

There are a few limitations to this thesis. The main issue is the lack of data. If we are looking at daily prices, then there are only about 260 working days in a year or 260 examples in a year. Since the stock prices are reliant on many factors, many features are needed for the machine learning model. If we are looking too far back for more examples, for instance, consider data of 2-3 years, then the market might not be acting the same way as it did three years ago. Another limitation is that this thesis is considering the news will impact the stock prices immediately, which might not always be the case. The third limitation is that the correlation factor used for this thesis was Pearson's correlation. This is a correlation and not necessarily a causation. While correlation might be useful, information about the causation can be immensely more helpful. Another limitation is the number of words used per article. The assumption to use the first 50 words per articles means that all the other words in each article are ignored. We don't know what impact that could have had on the results from the LSTM network.

Some of these limitations are easier to solve than others. The causation issue can potentially be solved using a delayed correlation approach where we use a certain time lag and calculate the correlation of the stock under consideration and the rest of the market. As for data limitations, one potential solution could be to build multiple machine learning models for different features, similar to the model used in this thesis but use a more sophisticated approach to combining the different machine learning models into one for the final price predictions. The final issue about understanding when certain news can impact the stock prices is a much harder problem, an issue for which there is no clear solution yet. One way could be to introduce a time lag factor into this model as well. One model could take the news as the input and change in stock price immediately after the news comes out as the target value. Another model would take a one-day delayed price change as target value and more models in a similar fashion. This approach can potentially tell us which news is impacting the prices, at which point in time in the future.

# Bibliography

[1] C.Lee Giles, Steve Lawrence, and Ah Chung Tsoi. Noisy time series prediction using a recurrent neural network and grammatical inference. *Machine Learning, Volume 44*, pages 161 – 183, 2001.

[2] George S Atsalakis and Kimon P Valavanis. Surveying stock market forecasting techniques - part 2: Soft computing methods. *Expert Systems with Applications, Volume 36, Issue 3*, pages 5932 – 5941, 2009.

[3] Wei Huang, Yoshiteru Nakamori, and Shou-Yang Wang. Forecasting stock market movement direction with support vector machine. *Computers and Operations Research, Volume 32, Issue 10*, pages 2513 – 2522, 2005.

[4] Qi.M. Nonlinear predictability of stock returns using financial and economic variables. *Journal of Business And Economics Statistics, Volume 17, Issue 4*, pages 419 – 429, 1999.

[5] Mehar Vijh, Deeksha Chandola, Vinay Anand Tikkiwal, and Arun Kumar. Stock closing price prediction using machine learning techniques. *Procedia Computer Science*, 167:599–606, 2020. International Conference on Computational Intelligence and Data Science.

[6] Hiransha M, Gopalakrishnan E.A., Vijay Krishna Menon, and Soman K.P. Nse stock market prediction using deep-learning models. *Procedia Computer Science*, 132:1351 – 1362, 2018.

[7] Robert P.Schumaker and Hsinchun Chen. Textual analysis of stock market prediction using breaking financial news the azfin text system. *ACM Transactions on Information Systems, Volume 27, Issue 2*, pages 1–19, 2009.

[8] Ryo Akita, Akira Yoshihara, and Takashi Matsubara. Deep learning for stock prediction using numerical and textual information. *Proceedings of the 15th International Conference on Computer and Information Science (ICIS)*, pages 1–6, 2016.

[9] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of computational science, Volume 2, Issue 1*, pages 1–8, 2011.

[10] Xiaodong Li, Haoran Xie, Li Chen, Jianping Wang, and Xiaotie Deng. News impact on stock price return via sentiment analysis. *Knowledge-Based Systems, Volume 69*, pages 14–23, 2014.

[11] Yauheniya Shynkevich, T.M. McGinnity, Sonya A. Coleman, and Ammar Belatreche. Forecasting movements of health-care stock prices based on different categories of news articles using multiple kernel learning. *Decision Support Systems*, 85:74 – 83, 2016.

[12] KiHwan Nam and NohYoon Seong. Financial news-based stock movement prediction using causality analysis of influence in the korean stock market. *Decision Support Systems*, 117:100–112, 2019.

[13] Karl Pearson. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London, Volume 58*, pages 240–242, 1895.

[14] Investopedia portfolio. https://www.investopedia.com/terms/p/portfolio.asp.

[15] Investopedia stocks. https://www.investopedia.com/terms/s/stock.asp. accessed: 08/23/2018.

[16] Burton Malkiel. *A Random Walk Down Wall Street*. William Warder Norton And Company, USA, 1973.

[17] Andrew W Lo and A Craig MacKinlay. Stock market prices do not follow random walks: Evidence from a simple specification test. *Review of financial studies, Volume 1, Issue 1*, pages 41 – 66, 1988.

[18] Intelligent economist. https://www.intelligenteconomist.com/stock-market. accessed: 08/23/2018.

[19] John J Murphy. *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. Penguin, 1999.

[20] Google finance. https://www.google.com/finance/.

[21] Event registry. https://eventregistry.org/.

[22] Fischer Black. Noise. *Journal of finance, Volume 41, Issue 3*, pages 528–543, 1986.

[23] Carson Kai-Sang Leung, Richard Kyle MacKinnon, and Yang Wang. A machine learning approach for stock price prediction. *Proceedings of the 18th International Database Engineering and Applications Symposium*, pages 274–277, 2014.

[24] Tom M Mitchell. *Machine Learning*. McGraw-Hill Education, 1997.

[25] Ryszard S Michalski, Jaime G Carbonell, and Tom M Mitchell. *Machine learning: An artificial intelligence approach*. McGraw-Hill Education, 1997.

[26] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2010.

[27] Karl Pearson, G U Yule, Norman Blanchard, and Alice Lee. The law of ancestral heredity. *Biometrika Trust*, pages 211–236, 1903.

[28] Ramachandra Rao Kurada. Unsupervised classification of uncertain data objects in spatial databases using computational geometry and indexing techniques. *International Journal of Engineering Research and Applications*, pages 806–814, 2012.

[29] S T Roweis and L K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science, Volume 290, Issue 5500*, pages 2323–2326, 2000.

[30] Karl Pearson. On lines and planes of closest fit to systems of points in space. *Journal of the American Statistical Association, Volume 58, Issue 301*, pages 559–572, 1901.

[31] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. *Semi-supervised learning*. MIT Press, 2006.

[32] Warren McCulloch and Walter Pitts. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics, Volume 52*, pages 115–133, 1943.

[33] Frank Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Brain Theory. Springer, 1961.

[34] Convolutional neural networks for visual recognition. http://cs231n.github.io/neural-networks-1/.

[35] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research, Volume 15*, pages 1929–1958, 2014.

[36] Jeffrey L Elman. Finding structure in time. *Cognitive Science Volume 14*, pages 179–211, 1990.

[37] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jurgen Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In *A Field Guide to Dynamical Recurrent Networks*. Wiley-IEEE Press, 2001.

[38] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. *Neural Computation, Volume 9*, pages 1735–1780, 1997.

[39] Santiago Fernandez, Alex Graves, and Jurgen Schmidhuber. An application of recurrent neural networks to discriminative keyword spotting. *17th International Conference on Artificial Neural Networks, Volume 4669*, pages 220–229, 2007.

[40] Jurgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks, Volume 61*, pages 85–117, 2015.

[41] Alex Graves and Jürgen Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou,

editors, *Advances in Neural Information Processing Systems 21*, pages 545–552. Curran Associates, Inc., 2009.

[42] Natural language toolkit. https://www.nltk.org/.

[43] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations, Volume abs/1412.6980*, 2014.

[44] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feed forward neural networks. *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.