University of Waterloo
Faculty of Engineering
Department of Electrical and Computer Engineering

Stockistics: Predicting stocks using NLU

Prepared by

Gaurav Lath, 20617513, glath@edu.uwaterloo.ca

Yi Wu, 20616574, y456wu@edu.uwaterloo.ca

Yuanxin Wang, 20636865, y2469wan@edu.uwaterloo.ca

Abstract

Predicting stock prices is an extremely challenging task as it requires analysis of many variables at the same time. Existing modelling methods are time consuming to build and are limited to human knowledge.

In this project, we propose a deep learning prediction model to solve the challenges of stock prediction. The novelty of our solution comes from supplementing a recurrent neural network model with sentiment data from public news articles along with features extracted from SEC filings. Sentiment data is extracted using Textblob’s sentiment analysis model and features from SEC filings are extracted using document summarization and text classification techniques.

When compared to just a time series model with no added features, the model created in this project demonstrated 75% performance increase in predicting the closing stock price on a given day.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Predicting stock market performance is one of the most important factors for financial service partners to make an investment decision. However, two key challenges have impeded the development of stock prediction in various industries for decades. The first challenge comes from the dependency on human knowledge. Stock analysts spend months integrating different relevant sources that affect stock prices in their mathematical models. This is extremely time consuming and is a bottleneck for trading efficiency. Secondly, humans can often miss important underlying patterns in data, especially in large text documents [1]. Therefore, an approach that makes use of neural networks is likely to be more efficient and accurate for predicting stocks. The report explores one such strategy that works well to predict stocks.

1.2 Scope

The scope of this report is limited to describing only the data sources, preprocessing techniques and machine learning models that were used to implement Stockistics. In addition, the results and analysis published are limited to testing on Salesforce stock ticker CRM, between the range of 2010 and 2019.

2 Architecture

The architecture of the project is drawn in Figure 1. There are three data sources used in the project –

1. Stock Data
2. News Articles
3. SEC Reports

Each of these data sources is pre-processed and then input to different machine learning models. Names of the models implemented are –

1. Time-series Model
2. Sentiment Model
3. NLP Model

All three models output different types of features, which are then combined and passed to a dense layer. The dense layer then outputs the predicted stock close value. The upcoming sections describe the data sources and details about the machine learning models.

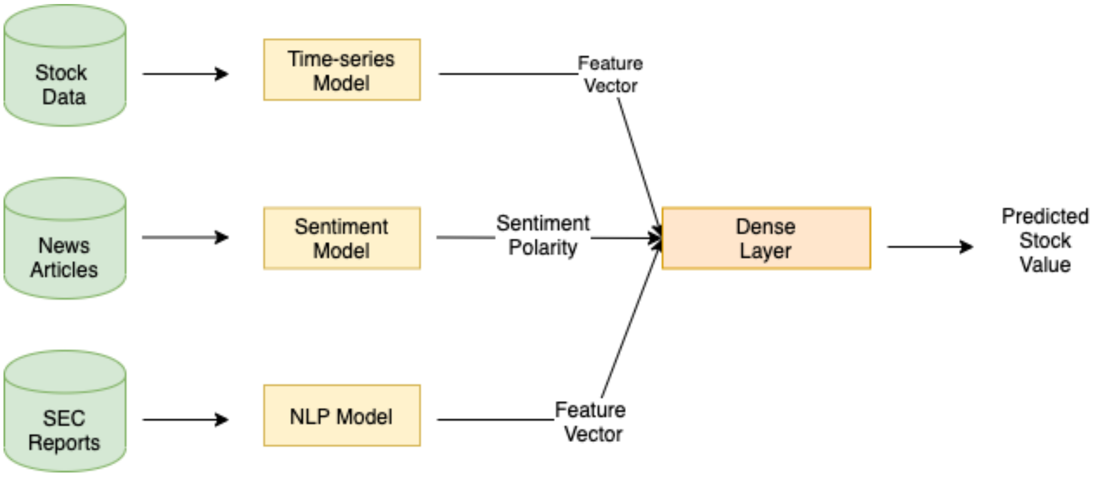


Figure 1. Architecture

3 Data Sources

This section discusses different types of data sources gathered and the intuition behind why they were included. The section then also discusses preprocessing methodologies used to prepare the data before inputting to the machine learning model.

3.1 Stock Data

One of the data sources extracted was stock information. The data was extracted using Yahoo Finance. The fields obtained were Date, High, Low, Volume, Open and Close.

1. High – Highest value a stock held on the given date
2. Low – Lowest value a stock held on the given date
3. Open – The value a stock held on the given date before market opened
4. Close –The value a stock held on the given date when market closed. It is also the target value of the model
5. Volume – The number of shares that were traded on the given date

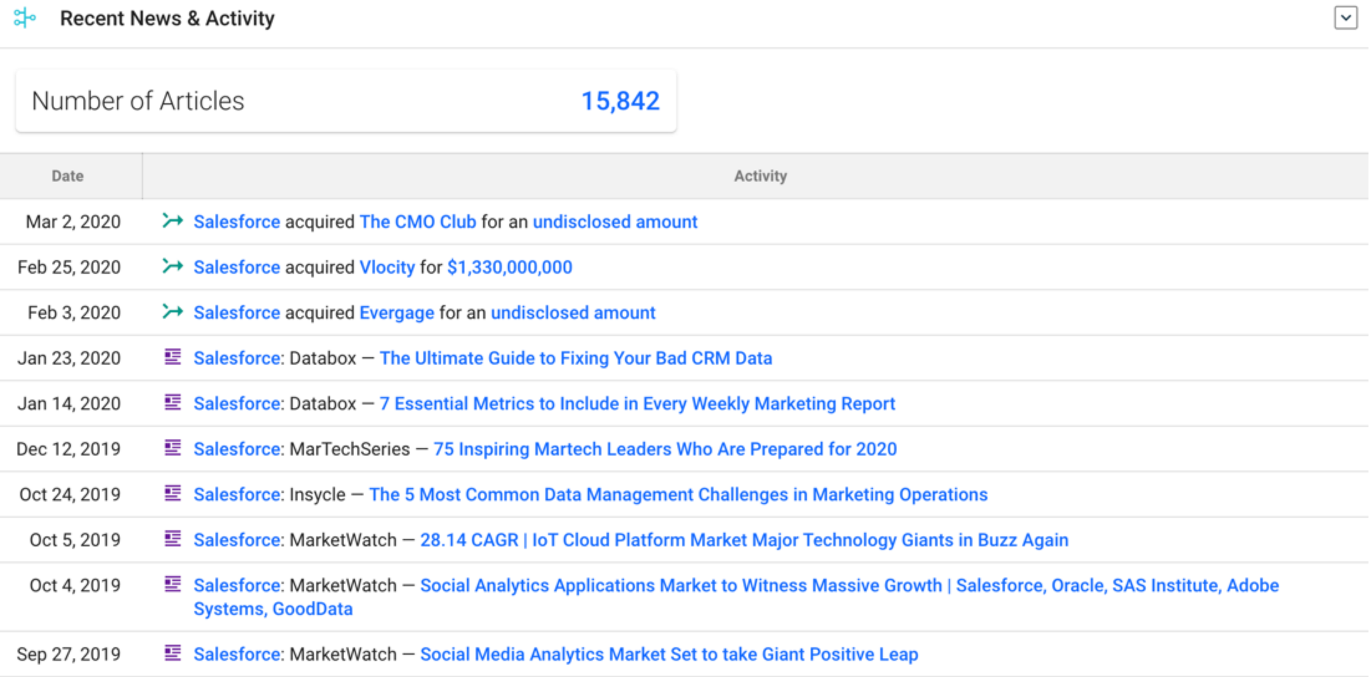
The value of 'High', 'Low', 'Open' and 'Close' variables can indicate the volatility of a stock. A significant deviation of 'High' and 'Low' value from the 'Open' value can indicate volatility and high risk while a smaller deviation can indicate stability and low risk. A high 'Volume' number can indicate that the stock is being actively traded. This can usually be attributed to the occurrence of an external event. Examples include acquisitions, the release of quarterly reports or a controversy. In conjunction with data from news articles and SEC reports, these features are expected to provide a strong co-relation with closing stock prices.

For preprocessing, all the variables excluding the target variable were normalized within the range of 0 and 1. Normalizing makes sure that all the features input to the machine learning model are treated the same and the coefficients of the model are not scaled according to the units of an input feature. In addition, normalization also helps the neural net perform gradient descent optimization faster and helps to avoid getting stuck in local optima.

3.2 News Articles

The stock price depends on the market's supply and demands. Therefore, people's willingness to buy and sell stocks has a large impact on the stock's price. People's opinion is often affected by social media and news. When the news articles suggest some positive aspects of a company, people are more likely to buy stocks of that company, and the stock price increases. In contrast, when the news articles suggest negative aspects of a company, people are likely to sell the stocks and the stock price will drop. Therefore, news articles about a company can potentially be an important feature for stock prediction.

The news articles are taken from CrunchBase, which is a platform for finding information about companies. By setting a time filter and a company's name, the platform displays all the news articles published about the company in the time range set by the user. An example of this can be seen in Figure 2. For the purpose of this project, news about Salesforce was queried between the time range of 2010 to 2019. Through web scraping, the dates and the news articles' titles are collected and put in a CSV file as shown in Figure 3. The file consisted of a total of 10,077 news articles.

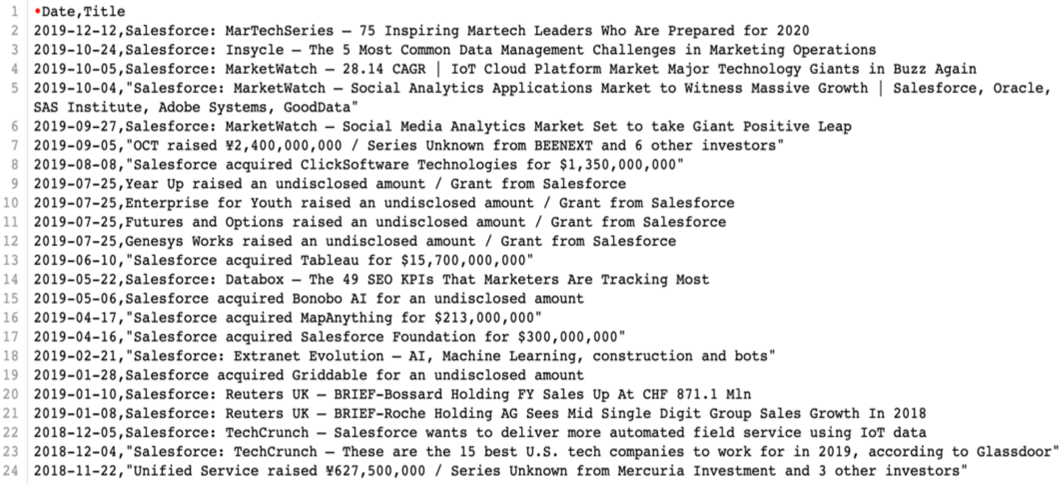


Recent News & Activity

Number of Articles 15,842

Date	Activity
Mar 2, 2020	Salesforce acquired The CMO Club for an undisclosed amount
Feb 25, 2020	Salesforce acquired Vlocity for $1,330,000,000
Feb 3, 2020	Salesforce acquired Evergage for an undisclosed amount
Jan 23, 2020	Salesforce: Databox — The Ultimate Guide to Fixing Your Bad CRM Data
Jan 14, 2020	Salesforce: Databox — 7 Essential Metrics to Include in Every Weekly Marketing Report
Dec 12, 2019	Salesforce: MarTechSeries — 75 Inspiring Martech Leaders Who Are Prepared for 2020
Oct 24, 2019	Salesforce: Insycle — The 5 Most Common Data Management Challenges in Marketing Operations
Oct 5, 2019	Salesforce: MarketWatch — 28.14 CAGR | IoT Cloud Platform Market Major Technology Giants in Buzz Again
Oct 4, 2019	Salesforce: MarketWatch — Social Analytics Applications Market to Witness Massive Growth | Salesforce, Oracle, SAS Institute, Adobe Systems, GoodData
Sep 27, 2019	Salesforce: MarketWatch — Social Media Analytics Market Set to take Giant Positive Leap

Figure 2. Sample articles about Salesforce from CrunchBase [2]



```
Date,Title
2019-12-12,Salesforce: MarTechSeries – 75 Inspiring Martech Leaders Who Are Prepared for 2020
2019-10-24,Salesforce: Insycle – The 5 Most Common Data Management Challenges in Marketing Operations
2019-10-05,Salesforce: MarketWatch – 28.14 CAGR | IoT Cloud Platform Market Major Technology Giants in Buzz Again
2019-10-04,"Salesforce: MarketWatch – Social Analytics Applications Market to Witness Massive Growth | Salesforce, Oracle,
SAS Institute, Adobe Systems, GoodData"
2019-09-27,Salesforce: MarketWatch – Social Media Analytics Market Set to take Giant Positive Leap
2019-09-05,"OCT raised ¥2,400,000,000 / Series Unknown from BEENEXT and 6 other investors"
2019-08-08,"Salesforce acquired ClickSoftware Technologies for $1,350,000,000"
2019-07-25,Year Up raised an undisclosed amount / Grant from Salesforce
2019-07-25,Enterprise for Youth raised an undisclosed amount / Grant from Salesforce
2019-07-25,Futures and Options raised an undisclosed amount / Grant from Salesforce
2019-07-25,Genesys Works raised an undisclosed amount / Grant from Salesforce
2019-06-10,"Salesforce acquired Tableau for $15,700,000,000"
2019-05-22,Salesforce: Databox – The 49 SEO KPIs That Marketers Are Tracking Most
2019-05-06,Salesforce acquired Bonobo AI for an undisclosed amount
2019-04-17,"Salesforce acquired MapAnything for $213,000,000"
2019-04-16,"Salesforce acquired Salesforce Foundation for $300,000,000"
2019-02-21,"Salesforce: Extranet Evolution – AI, Machine Learning, construction and bots"
2019-01-28,Salesforce acquired Griddable for an undisclosed amount
2019-01-10,Salesforce: Reuters UK – BRIEF-Bossard Holding FY Sales Up At CHF 871.1 Mln
2019-01-08,Salesforce: Reuters UK – BRIEF-Roche Holding AG Sees Mid Single Digit Group Sales Growth In 2018
2018-12-05,Salesforce: TechCrunch – Salesforce wants to deliver more automated field service using IoT data
2018-12-04,"Salesforce: TechCrunch – These are the 15 best U.S. tech companies to work for in 2019, according to Glassdoor"
2018-11-22,"Unified Service raised ¥627,500,000 / Series Unknown from Mercuria Investment and 3 other investors"
```

Figure 3. Extracted news dates and articles in a CSV file

3.3 SEC Reports

In equity research department in major financial service companies, SEC (U.S. Securities and Exchange Commission) mandated reports are one of the most important data sources that quantitative analysts use for stock prediction. The significant amount of textual data and quantitative information in the reports can bring inspiring insights to the stock price analysis process.

Among various SEC reports, the 8-K report, which consists of certain material corporate events on a more current basis of a company, plays an important role in stock trend analysis and investment decisions [3]. Since the format of 8-K reports for different companies can be very different, we focus on analyzing all the 8-K reports at Salesforce from 2010 to 2019. We downloaded 142 8-K reports at Salesforce in total using existing Python APIs in Edgar database.

However, most 8-K reports come in with an HTML format and the inconsistency use of HTML tags makes it difficult for us the extract valuable information without noise. Moreover, since an 8-K report contains all company event information for one year, there are over 20 kinds of items involved and it would be computationally expensive to perform analysis on all of them. Given the challenges above, we only look at one specific item in each report: "Item 9.01 Financial Statements and Exhibits". One visualization example of this item is shown in Figure 1 below.

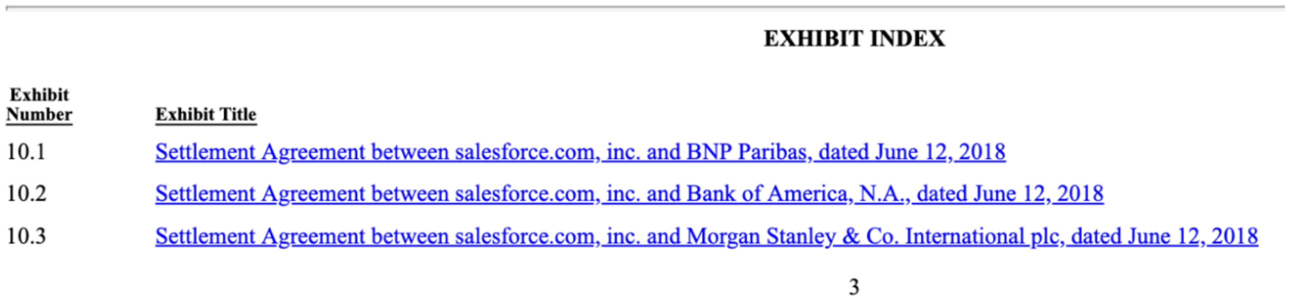


EXHIBIT INDEX

Exhibit Number	Exhibit Title
10.1	Settlement Agreement between salesforce.com, inc. and BNP Paribas, dated June 12, 2018
10.2	Settlement Agreement between salesforce.com, inc. and Bank of America, N.A., dated June 12, 2018
10.3	Settlement Agreement between salesforce.com, inc. and Morgan Stanley & Co. International plc, dated June 12, 2018

3

Figure 4. Sample Item 9 of a Salesforce 8-K Report

The reason we choose this particular item is because the business activities described in this section are clean and compact. On the other hand, in other sections like "Item 5.02 Departure of Directors or Certain Officers; Election of Directors; Appointment of Certain Officers; Compensatory Arrangements of Certain Officers", there might be thousands of words attached to describe human resource activities. Processing such type of data is incredibly challenging. Even if we utilized gated networks like LSTM and GRU, or the attention mechanism [4], the text is still too long for the model to capture long term dependencies. In some other sections, such as "Item 2.06 Material Impairments", the data present is less likely to be correlated to the closing stock price and is therefore not included. In our exploratory data analysis process, we perform statistical analysis of the lengths of text in Item 9.01 and show it in Figure 5. As we can see, the text lengths of Item 9.01 ranges from 0 to around 100, which is the perfect size for an LSTM network to capture effective information.

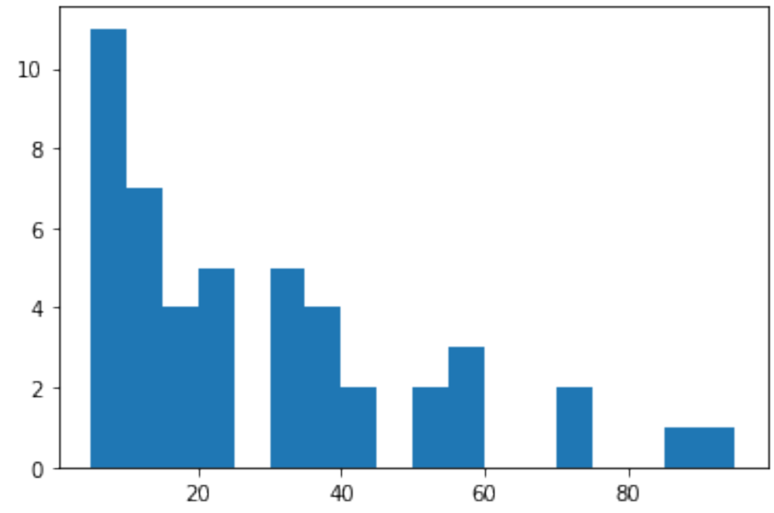


Figure 5. EDA on Text Lengths in Item 9.01

4 Machine Learning Models

This section discusses the implementation of the time-series model, sentiment model and the NLP model.

4.1 Time-series Model

The model used to extract features from time-series data is a Recurrent Neural Net (RNN) type model. The RNN makes use of features obtained from stock prices 60 days before the target date. The architecture of the time series RNN model is described in Figure 8 below.

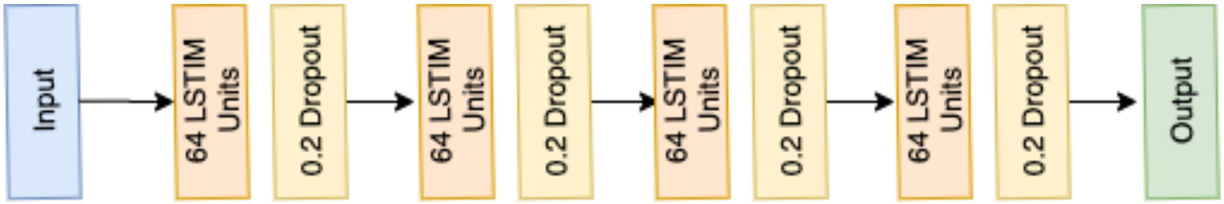


Figure 6. Architecture of the Time Series Model

The model is used to predict the stock prices of each t + 1 time step in the test data. The input of the model consists of stock data from t – 60 time steps. After the input layer, there are 4 layers of 64 LSTM units and a 20% dropout probability node. Finally, there is an output layer with a single unit that outputs the close price of the stock.

An RNN network is advantageous over a traditional neural network when the dataset has a sequence of information where the sequencing aspect of the data is more important than the spatial content of each individual frame [5]. This statement holds for this project as the dataset is a time-series where information from a previous timestamp can heavily impact the outcome of a later timestamp. Keeping this in mind, instead of neurons, LSTM units were implemented in the model. LSTM units can hold memory and are incredibly useful for training on time-series data. The dropout blocks after each LSTM unit helps to reduce overfitting which helps the neural net generalize better for data not seen by the model.

This architecture was trained using the Adam optimizer, for 20 epochs with the error function being the mean squared error function.

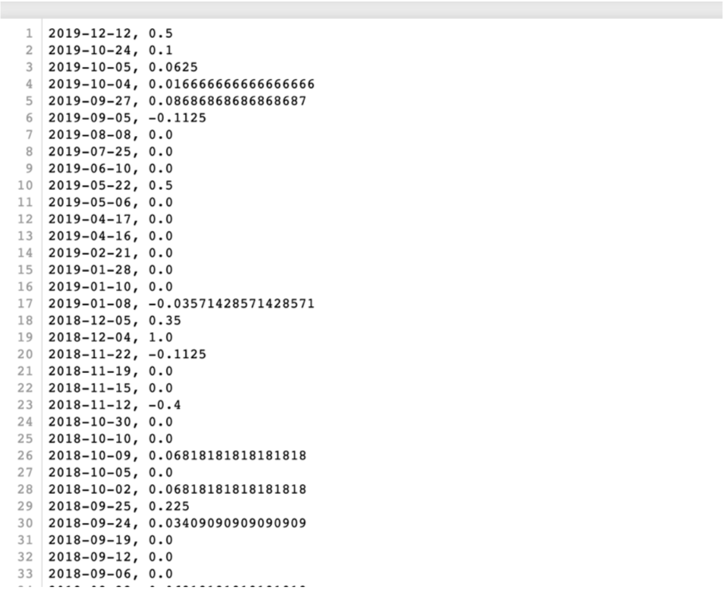
4.2 Sentiment Model

Social media and news have a strong impact on the stock price of a company. Positive news about the company can suggest a growing trend for its stock and negative news usually suggests an unwanted behaviour of the stock. Therefore, analyzing the positiveness of the news articles about a company is helpful for making stock price prediction on that day.

Sentiment analysis functionality is implemented through Textblob. Textblob is a Python library for processing textual data. It provides a simple API interface that allows the user to perform various text manipulations like noun phrase extraction, sentiment analysis, tokenization, word inflection, etc [6]. For the purpose of this project, Textblob has been used to generate sentiment polarity values for news articles about the company. After a sentence is loaded, it is first converted to a Textblob object. Once this is done, Textblob is ready to perform any manipulations the user chooses. After the sentiment analysis API is run, Textblob returns two results, polarity and subjectivity. Polarity is a float number between –1 and +1, where –1 indicates this sentence is negative and +1 indicates that the sentence is positive. Subjectivity is a number between 0 and 1 and suggests how subjective this sentence is. 0 means this sentence is totally objective and 1 indicates the opposite.

In this project, the input data consists of an array of article titles. Textblob does sentiment analysis on each title and gives its sentiment value. For example, when the title includes words like “acquire”, “donate”, “growing” and so on, Textblob outputs a positive value for sentiment polarity and suggests a positive behaviour of the company. In contrast, when the title includes words like “frustration” or “lay off”, Textblob outputs a negative value.

For each article collected from CrunchBase, a sentiment value is calculated. If multiple articles are seen on the same day, then the average value of all articles on the day is used as the feature. Figure 7 shows an example of sentiment polarity values generated for salesforce. The output CSV file is then picked up by the machine learning model described in the next section.



```
2019-12-12, 0.5
2019-10-24, 0.1
2019-10-05, 0.0625
2019-10-04, 0.016666666666666666
2019-09-27, 0.08686868686868687
2019-09-05, -0.1125
2019-08-08, 0.0
2019-07-25, 0.0
2019-06-10, 0.0
2019-05-22, 0.5
2019-05-06, 0.0
2019-04-17, 0.0
2019-04-16, 0.0
2019-02-21, 0.0
2019-01-28, 0.0
2019-01-10, 0.0
2019-01-08, -0.03571428571428571
2018-12-05, 0.35
2018-12-04, 1.0
2018-11-22, -0.1125
2018-11-19, 0.0
2018-11-15, 0.0
2018-11-12, -0.4
2018-10-30, 0.0
2018-10-10, 0.0
2018-10-09, 0.06818181818181818
2018-10-05, 0.0
2018-10-02, 0.06818181818181818
2018-09-25, 0.225
2018-09-24, 0.03409090909090909
2018-09-19, 0.0
2018-09-12, 0.0
2018-09-06, 0.0
```

Figure 7. CSV generated for sentiment polarity of news articles

4.3 NLP Model

After we download the SEC reports and extract Item 9.01 from each of them, we concatenate all the text from each date into a single sentence. This makes it easier to perform tokenization and other text preparation steps. The Keras Tokenizer object is utilized to convert a list of sentences to a list of numerical values. According to what we find in the exploratory data analysis process, we set the max length of the tokenizer to be 100 and set the direction of both the truncation and padding operations to be “post”.

Then we use pretrained "glove.840B.300d.txt", the Glove word embeddings, to convert these numerical values into 300-dimensional word vectors in Keras Embedding layer [7]. After the data is prepared properly, we develop a deep learning architecture that takes in the sentences and outputs a feature vector. The architecture of the model is described in Figure 4 below.

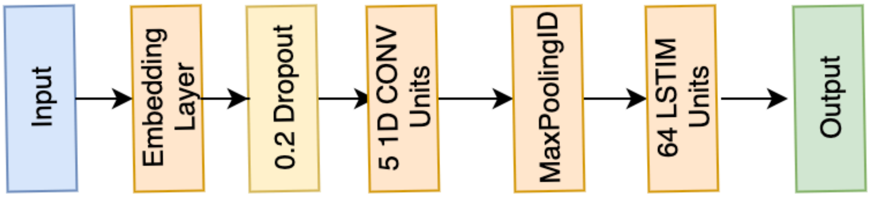


Figure 8. Architecture of the NLP Model

The input Embedding layer takes in vocabulary size and embedding dimension as input and outputs a vector. After the input layer, we utilize a 20% dropout probability node, a 1D Convolutional layer with 5 units, a max pooling layer, and one LSTM layer with 64 units. The combination of 1D CNN and LSTM are beneficial for extracting both the short-term and the long-term temporal trends in stock price changes [8]. Pooling layers are used to reduce the dimension of the intermediate features and reduce computational cost [9]. We also add dropout layers to reduce overfitting.

4.4 Combination

As stated in the architecture section above, the feature vector from original time series model, the feature vector from the NLP model, and the polarity Boolean value from the sentiment classification model, are concatenated together and passed through a fully connected layer (Dense layer) to make the final prediction.

5 Results

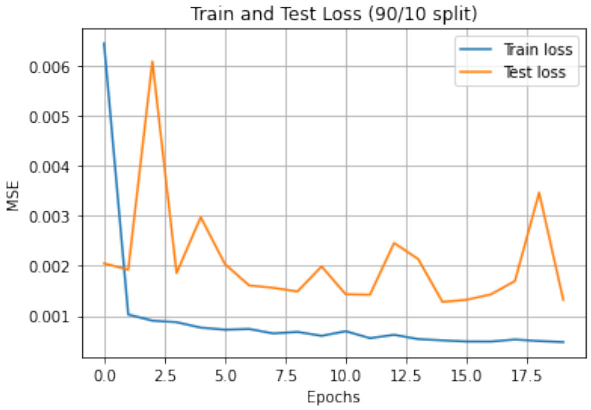
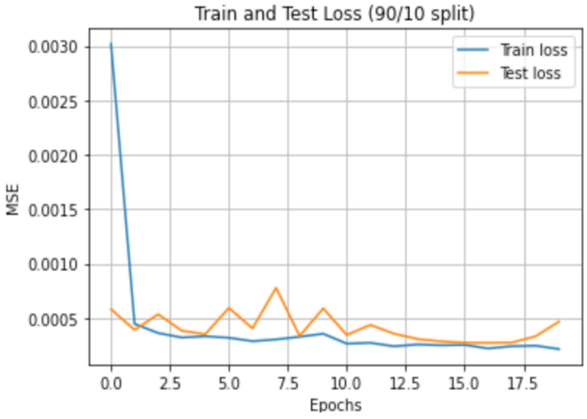
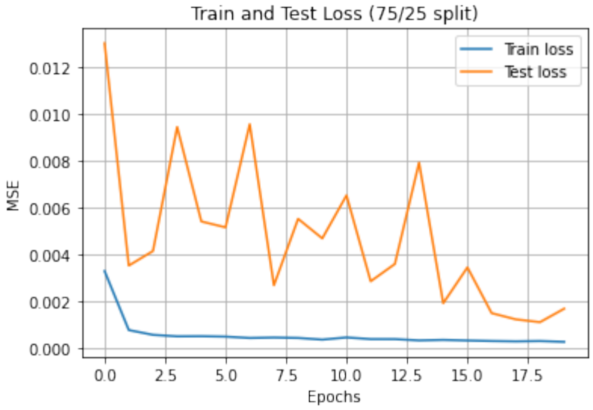
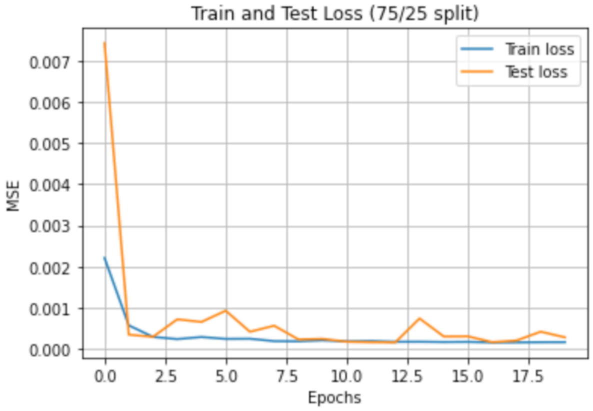
In results analysis, we want to mainly explore whether adding NLP features (including sentiment analysis) from news articles and SEC 8-K reports would improve the performance of a pure time-series forecasting model. Two metrics are used in this project: train/test loss and stock prediction curve comparison. For train-test split, given the nature of time series data, we simply cut the dataset using a split ratio; the former part is the training set and the latter part is the test set.

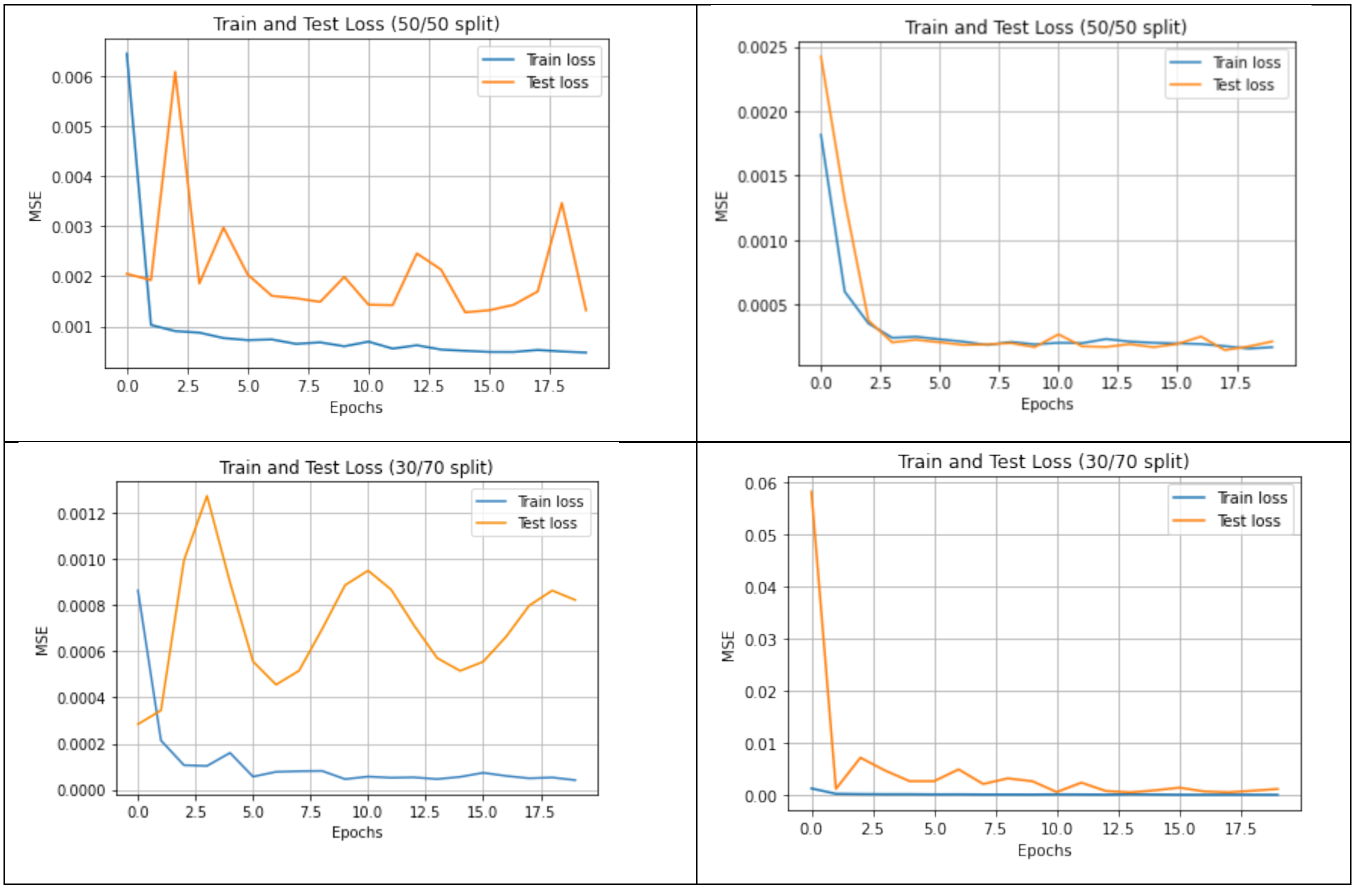
5.1 Train / Test Loss Analysis

When it comes to the train/test loss metrics, we want to compare the two models, namely the Time Series Model (Model 1) and the Time Series Model with NLP Features (Model 2), in different train-test splits. We not only focus on the mean squared error itself, but also the relative difference between train and test curves to monitor if the model overfits.

From Table 1 below, for mean squared errors in every train test split, both models have similar training loss performance after 20 epochs (MSE at around 1.4484e-04). However, when it comes to the test loss, Model 1 suffers from the problem of overfitting with MSE over 0.008 and there are huge gaps between training and test curves while in Model 2, the test error closely follows the training error at around 0.0003 and the model is perfect fitting. We speculate the reason might be that the pure time series model only learns some input-specific features from the training data and cannot generalize well to new samples while NLP features provide the network with more information to learn so that the network does not focus too much on noisy historical trend. In this part, we can conclude that Model 2 is better.

Table 1. Loss Comparison between Two Models for Different Splits

Time Series Model	**Time Series Model with NLP Features**
 Train and Test Loss (90/10 split)	 Train and Test Loss (90/10 split)
 Train and Test Loss (75/25 split)	 Train and Test Loss (75/25 split)

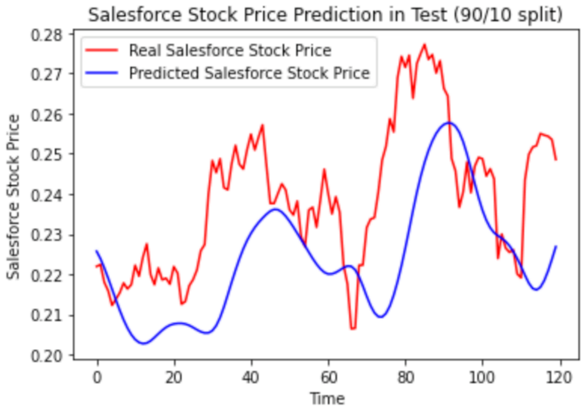
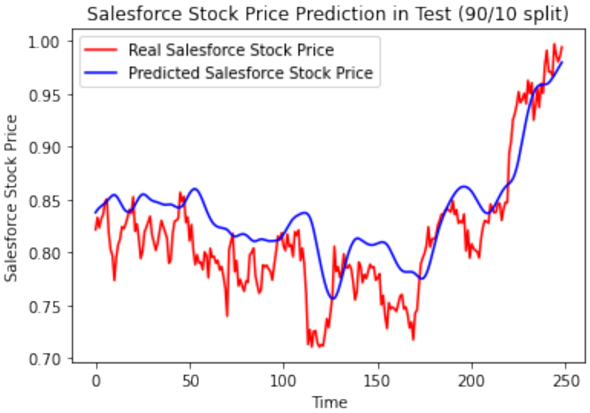
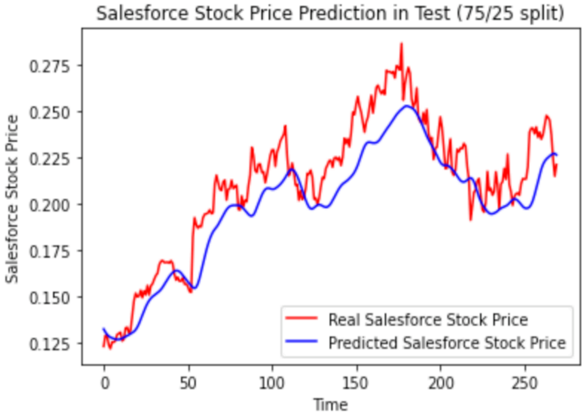
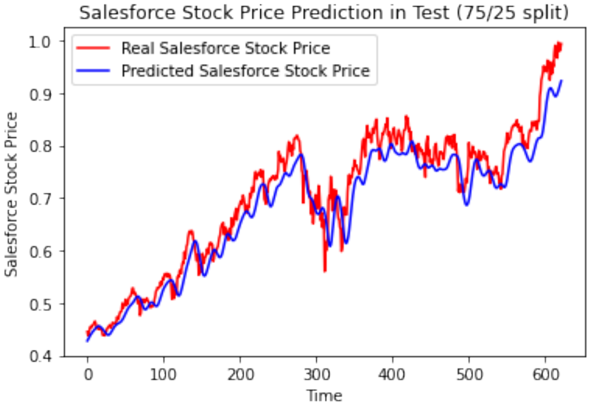
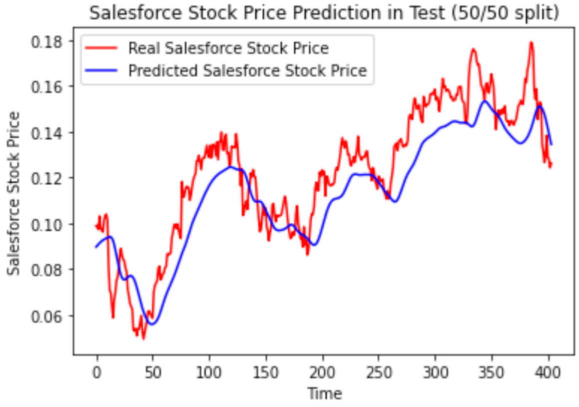
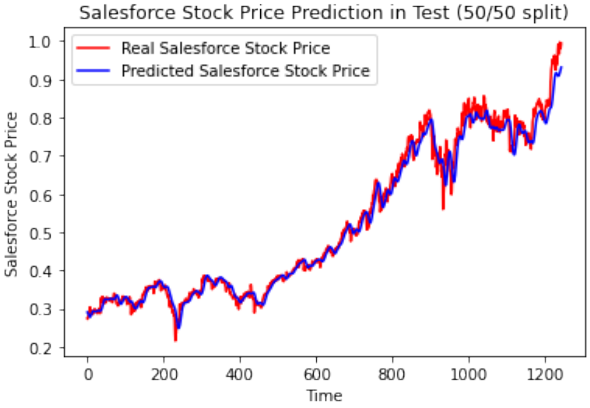


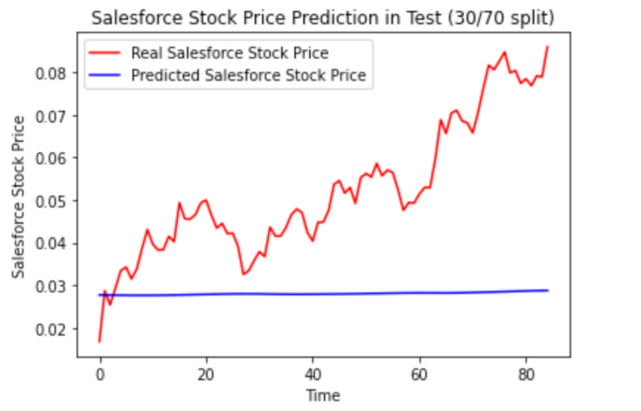
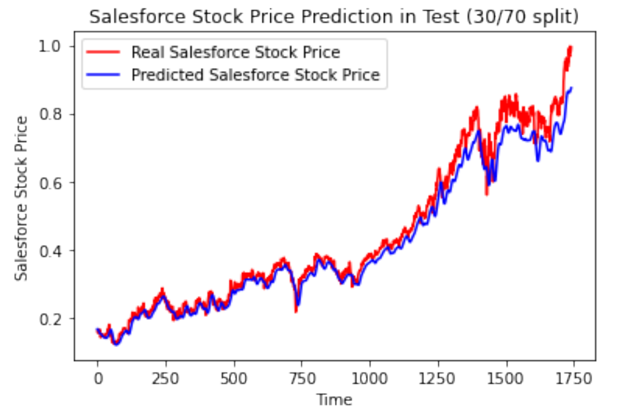
5.2 Test Set Prediction Curve Comparison

When it comes to test set prediction curve comparison, we find that our predictions in Model 2 follow the actual trends more closely than Model 1, which shows that adding NLP features can indeed improve prediction accuracy.

When we gradually increase the size of the test set and decrease the size of the training set, we want to investigate how well this model can generalize. In other words, if the model is trained on much data but only tested on little data, it is likely that the model overfits. However, if the model is trained on little data but it can still perform well on much test data, we can say that the model generalizes well. As we can see in Table 2, when the train test split is 90/10, both models are making bad predictions in the test set. When we gradually adjust the train test split ratio, Model 2 seems to perform better and better and in the last case, it can even generalize to 70% of the dataset with only 30% of the dataset being training data. But Model 1 seems to crash in the 30/70 split case since the model is not robust enough to learn useful features by training with only 30% of the dataset. In this part, we can also conclude that Model 2 is better.

Table 2. Test Set Fitting Comparison between Two Models for Different Splits

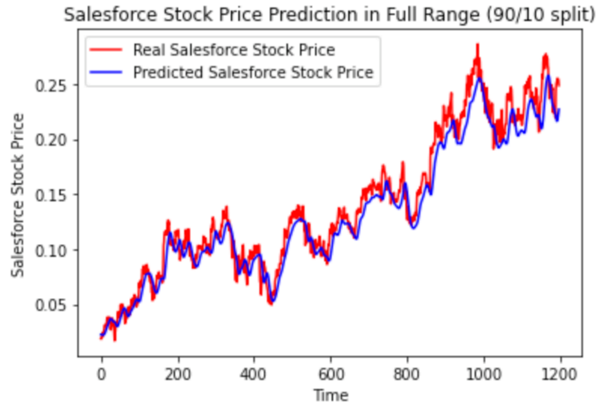
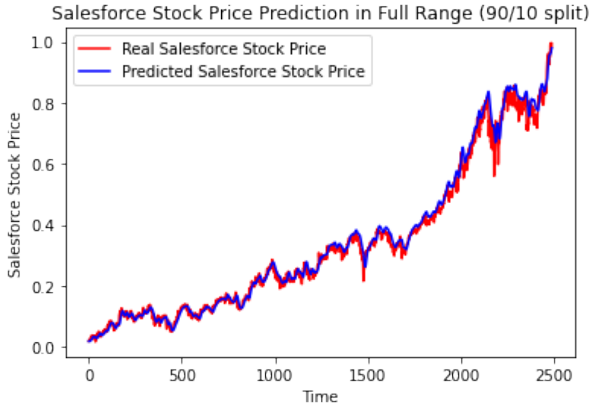
<table>
<tr><th>Time Series Model</th><th>Time Series Model with NLP Features</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>
</td><td>
</td></tr>
</table>

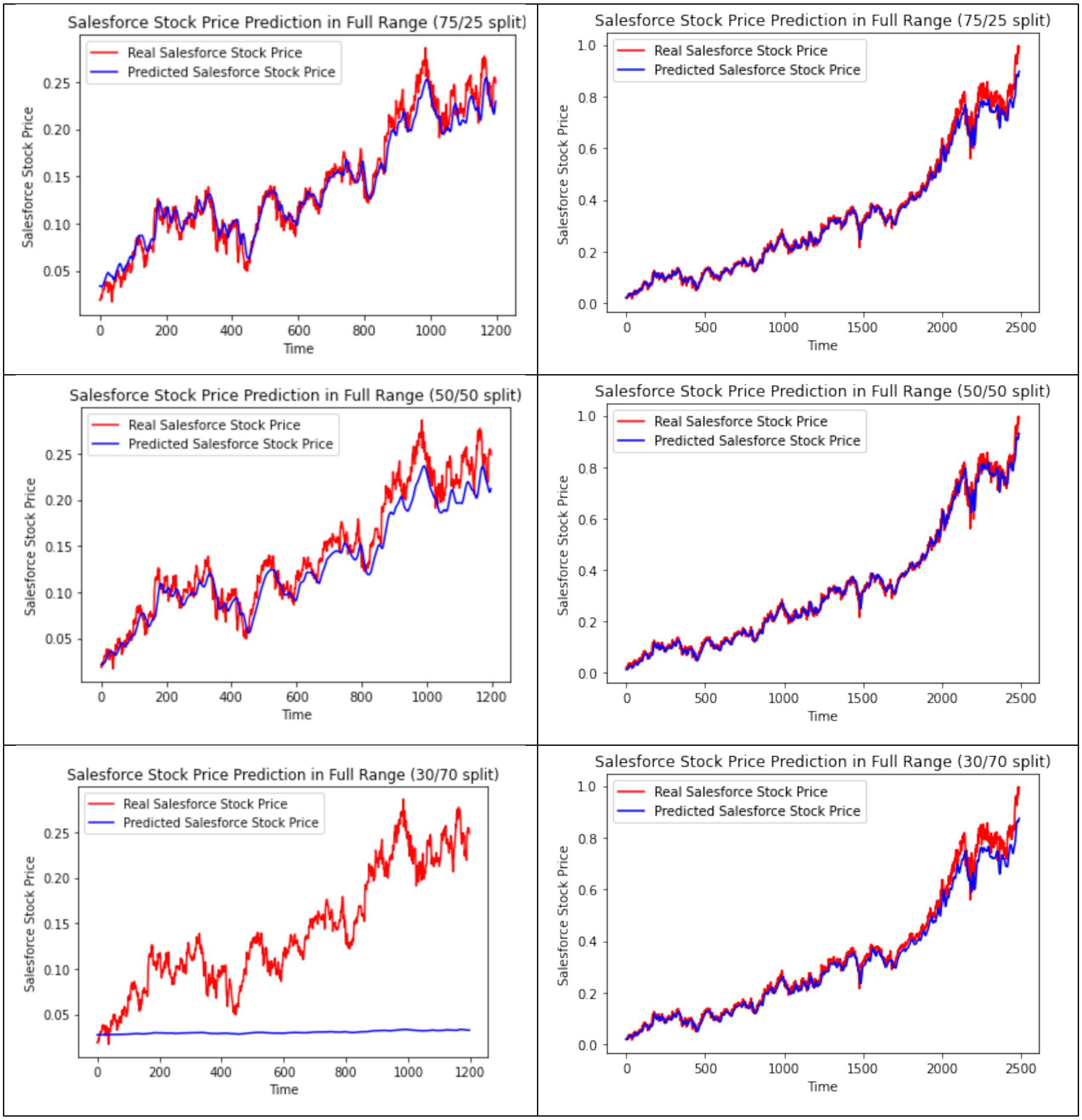
	

5.3 Full Range Prediction Curve Comparison

Since both models fit well with the training set after training for 20 epochs, the plots of training set prediction curve comparison are skipped in this report. However, it is still valuable to see how the models perform in the full time range (training plus test set) in Table 3. Similar to the analysis in Section 5.2, it can be seen that Model 2 shows a better fit in comparison to the actual stock prices for all train/test split ratios.

Table 3. Full Range Fitting Comparison between Two Models for Different Splits

Time Series Model	**Time Series Model with NLP Features**
	



Salesforce Stock Price Prediction in Full Range (75/25 split)

Salesforce Stock Price Prediction in Full Range (75/25 split)

Salesforce Stock Price Prediction in Full Range (50/50 split)

Salesforce Stock Price Prediction in Full Range (50/50 split)

Salesforce Stock Price Prediction in Full Range (30/70 split)

Salesforce Stock Price Prediction in Full Range (30/70 split)

6 Conclusions

In conclusion, after adding NLP features from news articles and SEC 8-K reports to a pure time series model, we can see a significant stock prediction performance increase. Not only do we obtain better full range predictions, but we also find that adding NLP features will help the original model the generalize better. Also, we find that 50-50 is a good train-test split ratio for both model to work well.

For future work, there are several areas the project can be improved. First, we can write more general HTML parser which can work for parsing other company reports, such as Google. Analyzing the results of this model on other company's data will give us a better idea about how well the model generalizes. In addition, we can also add 10-K, 10-Q, 8-Q SEC revenue reports from SEC to further enrich our NLP feature set. Last but not least, it would be interesting to increase and decrease the full-time range to further investigate the robustness of our new model.

References

[1] Y. Ahmed, “Stock Market Predictions with Natural Language Deep Learning”, Dec.2017; https://www.microsoft.com/developerblog/2017/12/04/predicting-stock-performance-deep-learning/.

[2] “Discover innovative companies and the people behind them,” *Crunchbase*. [Online]. Available: https://www.crunchbase.com/. [Accessed: 02-Apr-2020].

[3] “Fast Answers,” *SEC*, 10-Aug-2012. [Online]. Available: https://www.sec.gov/fast-answers/answersform8khtm.html. [Accessed: 02-Apr-2020].

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, “Attention Is All You Need,” *NeurIPS 2017*, pp. 6000–6010, Jun. 2017.

[5] P. Gudikandula, “Recurrent Neural Networks and LSTM explained,” *Medium*, 27-Mar-2019. [Online]. Available: https://medium.com/@purnasaigudikandula/recurrent-neural-networks-and-lstm-explained-7f51c7f6bbb9. [Accessed: 02-Apr-2020].

[6] Tutorial: Quickstart¶. (n.d.). Retrieved from https://textblob.readthedocs.io/en/dev/quickstart.html#sentiment-analysis

[7] J. Pennington, R. Socher, and C. D. Manning , “GloVe: Global Vectors for Word Representation,” *GloVe: Global Vectors for Word Representation*. [Online]. Available: https://nlp.stanford.edu/projects/glove/. [Accessed: 02-Apr-2020].

[8] J. Brownlee, “How to Develop 1D Convolutional Neural Network Models for Human Activity Recognition,” *Machine Learning Mastery*, 05-Aug-2019. [Online]. Available: https://machinelearningmastery.com/cnn-models-for-human-activity-recognition-time-series-classification/. [Accessed: 02-Apr-2020].

[9] “Keras Documentation,” *Pooling Layers - Keras Documentation*. [Online]. Available: https://keras.io/layers/pooling/. [Accessed: 02-Apr-2020].